UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMATICA
Departamento de Ingeniería del Software e Inteligencia Artificial

TESIS DOCTORAL

Impacto de los factores u organizaciones sociales en los procesos de recomendación para grupos

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Lara Quijano Sánchez

Directores

Mª Belén Díaz Agudo
Juan Antonio. Recio García

Madrid, 2015

Impacto de los factores y organizaciones sociales en los procesos de recomendación para grupos

TESIS DOCTORAL

Lara Quijano Sánchez

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Impact of social factors and organizations in group recommendation processes

Memory presented to apply for the degree of Doctor in Computer Science

Lara Quijano Sánchez

Advisors

Mª Belén Díaz Agudo
Juan A. Recio García

Publication Date: **April 2015**

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Impacto de los factores y organizaciones sociales en los procesos de recomendación para grupos

Memoria que se presenta para optar al título de Doctor en Informática

Lara Quijano Sánchez

Directores

Mª Belén Díaz Agudo
Juan A. Recio García

Fecha de Publicación: **Abril 2015**

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

A mis padres,

Agradecimientos

Do or do not. There is no try.
Yoda

Una vez finalizada esta etapa tan bonita de mi vida que emprendí hace ya cuatro años quiero dar las gracias a todas las personas que me han acompañado en este viaje.

En primer lugar quiero agradecer a mis directores de Tesis, Belén y Juanan, el haberme introducido y acompañado por este "mundillo" que me encanta, pues terminé la carrera siendo un ratoncillo de biblioteca sin saber qué rumbo seguir y gracias a vosotros he descubierto mi vocación. Gracias por haberme permitido dar rienda suelta a mi imaginación y escuchar mis ideas, por alentarme a que me atreviera con más, por enseñarme a no ir deprisa y corriendo y obligarme a repasar y ser ordenada. Gracias por lanzarme al mundo, animarme a salir de viaje y relacionarme, todas esas experiencias me han enriquecido mucho tanto profesional como personalmente. Gracias por los consejos, las reuniones, las charlas y todo el tiempo que me habéis dedicado.

A continuación quiero darle las gracias a mis compañeros del grupo de investigación de GAIA, a mis compañeros de despacho del 411 y a todos los profesores que me han dado buenos consejos, motivándome y orientándome tanto en las aulas como por los pasillos de la facultad. Gracias Pedro por dejarme formar parte del grupo. Mil gracias Guille por haberme hecho el tour de iniciación, contestar siempre a mis dudas y estar siempre disponible intentando colaborar. Gracias Raquel, Antonio, Javi, David, Gonzalo, Almudena, Virginia, Manuel, Ismael, Fede, todos!!! gracias por darme siempre buenos consejos, por escucharme cuando me cabreaba con los papers y por estar allí siempre que he necesitado vuestra ayuda.

Por otro lado, el impacto que haya podido tener este trabajo a nivel internacional se debe fundamentalmente a los congresos a los que he podido asistir y a mis estancias de investigación. Por ello mi más sincero agradecimiento al grupo GAIA, del que he formado parte, y a las personas que lo han hecho posible. Así, el doctor Derek Bridge hizo posible mi estancia en la University College Cork: *Dear Derek thank you very much for everything.*

My stay in Cork was the best experience of my life! Thank you for letting me learn from you, for sparing your time, for guiding me, for your advice and last but not least for making sure I was eating properly. Por su parte el doctor Thomas Roth-Berghofer ejerció de anfitrión en mi estancia en la University of West London: *Thank you very much Thomas for inviting me and giving me the chance to work with you and your team.*

Un agradecimiento más formal a los proyectos TIN2009-13692-C03-03 del Ministerio de Ciencia e Innovación, TSI-020110-2009-205 del Ministerio de Industria Turismo y Comercio y en especial a la Universidad Complutense de Madrid por su beca Formación de Personal Investigador pues han financiado este trabajo.

Y ahora llega el momento de nombrar a todos mis amigos y familia que me han apoyado, me han animado, me han ayudado desinteresadamente y se han ofrecido a ser mis conejillos de indias en todo momento:

En primer lugar quiero dar las gracias en especial a mi queridísimo amigo Ángel, porque para mí es el mejor informático que existe en el mundo, porque me ha ayudado en todo momento, porque me ha aguantado, porque me ha enseñado mucho, porque se ha sentado conmigo cuando todo me explotaba y por ser taaaaaan buena persona GRACIAS! Gracias también a María por leerse mis papers sin tener ni idea del tema y echarle horas para ayudarme a mejorar el estilo, eres la mejor traductora del mundo. Gracias Andrea por diseñarme el ya mundial logo de *HappyMovie*. Gracias Iván, Noe y Sara por meteros una y otra vez en mis páginas en prueba para ver si "cascaban". Gracias a toda la panda de la uni: Paco, Ferry, Miguel, Tam, Jesus, Rocio, Alex, Kike, Javi, todos! por acompañarme a tomar mi zumo de las 12, por todas las risas y por contestarme a 8000 tests para la Tesis. Gracias también a todos los que me han contestado algún test alguna vez, a mis amigos de Cork y a mis amigas de la pisci.

También quiero agradecer el apoyo de mi familia, porque siempre han confiado en mi más que yo y porque siempre han hecho todo por ayudarme y cuidarme. Muy en especial quiero agradecerles TODO a mis padres, porque son los mejores del mundo, porque son mis managers, porque me guían, porque sus consejos no tienen precio, porque me apoyan, porque creen en mi, por su cariño, porque no estaría donde estoy sin ellos y porque se lo debo todo.

Y por último gracias a ti Dani por ser mi príncipe azul y vivir conmigo un cuento de hadas.

About this document

This PhD Thesis is presented as a compendium of published papers. The document is divided in 3 parts: a summary of the Thesis in English, a summary of the Thesis in Spanish and the set of published papers that this Thesis has given rise to. The first two parts of the document, English and Spanish summaries, present an introduction of the work developed in this PhD Thesis accompanied by a revision of the state of the art in the field. Next, the goals and main contributions of this research are presented along with a brief summary of the developed work. Finally, the conclusions made throughout this Thesis work are stated. The third part of the document is comprised by the papers that are presented as core of this PhD Thesis.

Esta Tesis Doctoral se presenta como compendio de publicaciones editadas. El documento está dividido en 3 partes: un resumen de la tesis en inglés, un resumen de la tesis en castellano y el conjunto de publicaciones que ha dado lugar esta Tesis. Las primeras dos partes del documento, resúmenes en inglés y en castellano, exponen una introducción al trabajo realizado, acompañada de una revisión del estado del arte del campo. A continuación, se describe el planteamiento de los objetivos del trabajo y las contribuciones principales junto a un breve resumen del trabajo desarrollado. Finalmente, se incluyen las conclusiones a las que se ha llegado tras este trabajo de Tesis. La tercera parte de este documento la componen los artículos que se presentan como núcleo central de esta Tesis Doctoral.

The presented papers within this PhD Thesis are the ones exposed next in chronological order:

Los artículos que se aportan como parte de la Tesis Doctoral son los que se exponen a continuación en orden cronológico:

- Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B. Social Based Recommendations to Groups. in: Procs. of the 14th UK Workshop on Case-Based Reasoning. CMS Press, University of Greenwich, pages: 46-57, 2009. ISBN 978-1-904521-64-8.

- Recio-García, J. A., Díaz-Agudo, B., González-Sanz S. and Quijano-Sánchez, L. Distributed Deliberative Recommender Systems. Journal of Transactions on Computational Collective Intelligence, 1; pages: 121-142, Springer, 2010.

- Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B. Personality and Social Trust in Group Recommendations, in: Procs. of the 22th International Conference on Tools with Artificial Intelligence, ICTAI'10., IEEE Computing Society, Arras, France, pages: 121-126, 2010. ISBN 978-0-7695-4263-8.

- Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B, Jimenez-Díaz, G. Happy Movie: A Group Recommender Application in Facebook. in: Procs. of the 24th International Florida Artificial Intelligence Research Society Conference, FLAIRS. Palm Beach, Florida, USA, AAAI Press, pages 419-420, 2011. ISBN 978-1-57735-501-4.

- Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B. Using personality to create alliances in group recommender systems in: Procs of the 19th International Conference on Case Based Reasoning, ICCBR'11. Springer-Verlag Berlin Heidelberg, Greenwich, London, pages: 226-240, 2011. ISBN 978-3-642-23290-9.

- Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B. User satisfaction in long term group recommendations in: Procs. of the 19th International Conference on Case Based Reasoning, ICCBR'11. Springer-Verlag Berlin Heidelberg, Greenwich, London, pages: 211-225, 2011. ISBN 978-3-642-23290-9.

- Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B. Group recommendation methods for social network environments in: 3rd Workshop on Recommender Systems and the Social Web, en el 5th ACM International Conference on Recommender Systems, RecSys'11, Chicago, IL, USA, 2011. ISBN 978-1-4503-0683-6.

- Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B. HappyMovie: A Facebook Application for Recommending Movies to Groups, in: Procs. of the 23th International Conference on Tools with Artificial Intelligence, ICTAI'11., IEEE Computing Society, Boca Raton, Florida, USA, pages: 239-244, 2011. ISBN 978-1-4577-2068-0.

- Quijano-Sánchez, L., Bridge, D., Díaz-Agudo, B., Recio-García, J. A., A Case-Based Solution to the Cold-Start Problem in Group Recommenders, in: Procs. of the 20th International Conference on Case Based Reasoning, ICCBR'12, Lyon, France, pages: 342-356, 2012. ISBN 978-3-642-32985-2.

- Quijano-Sánchez, L., Bridge, D., Díaz-Agudo, B., Recio-García, J. A., Case-Based Aggregation of Preferences for Group Recommenders, in: Procs. of the 20th International Conference on Case Based Reasoning, ICCBR'12, Lyon, France, pages: 327-341, 2012. ISBN 978-3-642-32985-2.

- Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo,B, Jimenez-Díaz, G. Social Factors in Group Recommender Systems. Journal Transactions on Intelligent Systems and Technology. Special Issue: Social Recommender Systems volume 4, issue 1, pages: 8, ACM Transactions on Intelligent Systems and Technology, 2013. ISSN 2157-6904.

- Quijano-Sánchez, L., Bridge, D., Díaz-Agudo, B. y Recio-García, J. A. A case-based solution to the cold-start problem in group recommender, in Procs. of the Best Papers in Sister Conferences Track at the 23rd International Joint Conference on Artificial Intelligence, IJCAI'13. AAAI Press, Beijing, China, pages: 3042-3046, 2013. ISBN 978-1-57735-633-2.

- Quijano-Sánchez, L., Bridge, D., Díaz-Agudo, B. y Recio-García, J. A. A case-based solution to the cold-start problem in group recommender, in Procs. of the Best Papers in Sister Conferences Track at the 23rd International Joint Conference on Artificial Intelligence, IJCAI'13. AAAI Press, Beijing, China, pages: 3042-3046, 2013. ISBN 978-1-57735-633-2.

- Recio-García, J. A., Quijano-Sánchez, L., Díaz-Agudo, B. Including Social Factors in an Argumentative Model for Group Decision Support Systems. Decision Support Systems, DSS, 56, pages: 48-55, Elsevier, 2013. ISSN: 0167-9236.

- Quijano-Sánchez, L., Bridge, D., Feedback on group recommendations, in Workshop on Case-based Reasoning in Social Web Applications, in: Procs. of the 21st International Conference on Case Based Reasoning, ICCBR'13, Saratoga Springs, NY, USA, pages: 17-26, 2013. ISBN 978-3-642-39055-5.

- Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B., A Reusable Methodology for the Instantiation of Social Recommender Systems, in: Procs. of the 25th International Conference on Tools with Artificial Intelligence, ICTAI'13, IEEE Computing Society, Washington DC, USA, pages: 777-782, 2013. ISBN: 978-1-57735-633-2.

- Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B., An Architecture for Developing Group Recommender Systems Enhanced by Social Elements, Applied Intelligence, volume 40, issue 4, pages: 732-748, Springer, 2014. ISSN: 0924-669X.

- Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B., Development of a Group Recommender Application in a Social Network, Special Issue on Knowledge-Bases for Cognitive Infocommmunications, Knowledge-Based Systems, pages: 72-85. Elsevier, 2014. ISSN: 0950-7051.

- Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B., A Reusable Methodology for the Instantiation of Social Recommender Systems. International Journal on Artificial Intelligence Tools (IJAIT), volume 23, issue 6, World Scientific, 2014. ISSN: 0218-2130.

Abstract

In present-day society we are living in an environment where information overload grows bigger everyday. This fact –having so many different choices– makes finding concrete items of our liking a very difficult task, specially in the e-commerce field. As a result of this, recommender systems are becoming increasingly important, guiding users in the task of choosing items in catalogues and/or services available in the Web that are of their liking. Besides, with the advent of the Social Web and the growing popularity of Social Networks rises the opportunity to take advantage of the personal information provided by users when interacting with them and use it as new sources of knowledge for the development of new recommendation strategies that focus and benefit from the social information stored in them. It is in this "social" context where groups of people gather together and plan to consume items collectively as part of a joint activity, such as cinema outings, going to a restaurant, planning holidays and many more. This way of consuming items in groups is the origin of group recommender systems, a line of work in which we are focused and whose goal is to ease group decision-making processes when having to choose a joint activity.

The rising research line in group recommender systems currently suffers from a number of short-comings that hamper their effectiveness based on the incorrect modeling of users' social behaviour. In this PhD Thesis we formulate the hypothesis that adding social empowered techniques to recommender systems could significantly improve the overall quality of group recommendations. Our main goal has been the generation of a set of recommendations that satisfy a group of users, with potentially competing interests. To do so, we have reviewed different ways of combining peoples' personal preferences and proposed an approach that takes into account the social behaviour within a group. Our approach, named *Social Recommendation Model* (*SRM*), defines a set of recommendation techniques that include the analysis and use of several social factors: the *personality* of group members in conflict situations, the *trust* between them, the concept of *homophily* inside the structure of the group, the *persuasive* capabilities of its members, and, the reuse of the *experience* obtained through previous recommendations. Therefore, the main contribution of this PhD Thesis work is the use of social information to make enhanced recommendations to groups. We propose a generic platform that allows the reuse of our *Social Recommendation Model* in different domains through an architecture named ARISE (*Architecture for Recommendations Including Social Elements*) and a methodology to perform this reuse through a software developing process based on templates that conceptualize the *SRM*. Besides, in order to verify its viability we have developed two applications that represent an instantiation of the platform and that serve us as use cases to evaluate the *SRM*. These instanciations also serve us to evaluate the performance of the different proposed recommendation techniques.

KeyWords: Group Recommenders, Personality, Trust, Social Networks, Experience, Social Applications, Generic Architecture, Templates.

Resumen

En la sociedad actual en la que vivimos existe cada vez más una gran sobrecarga de información sobre productos de consumo. Esto hace que resulte muy difícil encontrar productos concretos de nuestro agrado entre la gran baraja de posibilidades, especialmente en el ámbito del comercio on-line. A raíz de este problema los sistemas recomendadores están cobrando cada vez más importancia, guiando al usuario en la selección de productos en catálogos y/o servicios disponibles en la web. Además, con la aparición de la web social y la creciente popularidad de las redes sociales surge la oportunidad de aprovechar la información personal proporcionada por los usuarios al interacturar en las mismas como fuentes de conocimiento para el desarrollo de nuevas estrategias de recomendación que se centren y beneficien de dicha información social. Es en este entorno "social" donde las personas se agrupan y planean consumir productos como parte de una actividad conjunta, ya sea ir al cine, a un restaurante, de vacaciones o un sin fin de posibilidades. De esta forma de consumir productos en grupo nacen los sistemas de recomendación grupal, trabajo en el que nos hemos centrado y cuyo objetivo es facilitar los procesos de toma de decisiones a grupos que desean realizar una actividad conjunta.

La línea de investigación en recomendadores grupales, que se encuentra actualmente en auge, sufre una serie de deficiencias, radicadas en un incorrecto modelado de los comportamientos sociales de los usuarios, que reducen su eficiencia. En esta Tesis Doctoral formulamos la hipótesis de que la inclusión de conocimiento social en los sistemas de recomendación supondría una mejora significativa en la calidad global de las recomendaciones grupales. Nuestro objetivo principal ha sido la generación de recomendaciones que satisfagan a un grupo de usuarios con intereses potencialmente opuestos. Para ello, hemos revisado diferentes formas de combinar las preferencias personales de las personas y hemos propuesto un enfoque que tiene en cuenta el comportamiento social de los usuarios dentro de un grupo. Nuestra aproximación, que llamamos *Modelo de Recomendación Social* (*MRS*), define un conjunto de técnicas de recomendación que incluyen el análisis y el uso de varios factores sociales: la *personalidad* de los componentes del grupo en situaciones conflictivas, la *confianza* entre ellos, el concepto de *ho-*

mofilia dentro de su estructura, la capacidad de *persuasión* de sus miembros y, la reutilización de la *experiencia* obtenida en recomendaciones pasadas. Por lo tanto, la aportación fundamental de este trabajo de Tesis doctoral es el uso de información social para mejorar las recomendaciones a grupos. Proponemos una plataforma genérica que permite la reutilización de nuestro *Modelo de Recomendación Social* en diferentes dominios mediante una arquitectura llamada ARISE (*Architecture for Recommendations Including Social Elements*) y una metodología para realizar esa reutilización por medio de un proceso de desarrollo software basado en plantillas que conceptualizan el comportamiento del *MRS*. Además, para demostrar su viabilidad hemos desarrollado dos aplicaciones que instancian la plataforma y que sirven de casos de estudio para evaluar el *MRS*. Estas instanciaciones también nos sirven para evaluar las diferentes técnicas de recomendación propuestas.

Palabras Clave: Recomendaciones grupales, Personalidad, Confianza, Redes Sociales, Experiencia, Aplicaciones Sociales, Arquitectura Genérica, Plantillas.

Contents

List of Figures

List of Tables

Part I

Contributions

Chapter 1

Introduction

Group recommender systems are currently one of the most active research lines in the recommendation area. Traditional recommender systems –i.e. individual recommenders– have proved their relevance and repercussion by being one of the main pillars in e-commerce (Adomavicius and Tuzhilin, 2005; Ricci et al., 2011). For example, Webs as popular as Amazon[1] use recommendation techniques to guide users in the task of choosing items of their liking. Individual recommenders focus on the easier case: one user consuming products. However, they ignore the fact that some items, mainly in leisure items, are consumed in groups of people, such as couples, families and parties of friends. The choice of a date movie, a family holiday destination, or a restaurant for a celebration meal all require the balancing of the preferences of the group members. This kind of recommendations are provided by group recommender systems (Jameson and Smyth, 2007). The process followed by these systems normally consists in the aggregation of ratings[2] (real or estimated) for each group member (Jameson and Smyth, 2007; Baltrunas et al., 2010; Berkovsky and Freyne, 2010; Pessemier et al., 2013). The aggregation functions typically used are inspired by the social welfare functions developed by the *Social Choice Theory* research (Masthoff and Gatt, 2006). However, this widely accepted approach for group recommendations ignores the social factors that influence real group decision-making. The research line followed by this PhD Thesis stems from this lack of coverage and proposes a novel group recommendation model, called *Social Recommendation Model* (*SRM*), that includes social factors.

With the advent of the Social Web (Open Diary, 1998[3]) and the growing popularity of Social Networks[4], where users explicitly provide personal

[1] `http://www.amazon.com/`

[2] Ratings are defined as users' preferences for different items.

[3] `http://www.OpenDiary.com`

[4] Nowadays (2015), there are more than 200 Social Networks and almost 2 billion of users.

information and interact with others and the system, it is becoming clear that the key for the success of recommendations is to develop new strategies which focus on recommendations leveraged by these new sources of social knowledge. In our model we propose the use of Social Networks as execution platform and/or social information source, proving that this approach improves group recommenders' performance. In our proposal, as we will see throughout this Thesis work, we outline two different social aspects inside group dynamics: the individual role that each group member plays, that depends on the personality or persuasive capabilities of each individual. And, on the other hand, the social relationships and behaviour between group members, that depends on the trust between group members or the sense of justice, and that rise when studying the group as an entity itself.

Commonly, group recommender systems include prediction of ratings and aggregation techniques (Jameson and Smyth, 2007). Following this schema, the system works as follows: First, for each group member, an individual recommender system predicts a set of ratings for the candidate items. Secondly, the group recommender aggregates the ratings: for each candidate item, it might for example take the average of group members' ratings, the minimum or the maximum (Masthoff, 2004). Finally, it recommends to the group the items with the highest aggregated ratings. However, this approach does not take into account the fact that groups of people can have very different characteristics, including structural characteristics like size, similar or antagonistic personal preferences, or different social relationships between its members. Here stems the general hypothesis that we formulate in this Thesis: ***"The real satisfaction of a group regarding a group recommendation cannot be accurately estimated using the simple aggregation of its members' individual preferences. Considering people as social entities that relate with each other allows the better estimation of their individual satisfaction regarding the result of the recommendation and, therefore, improves the global group satisfaction"***. More concretely we have studied two main social factors: *personality* and *trust* and three secondary social factors: *homophily*, *persuasiveness* and *justice*. These social factors define each person as a potentially influenced component of a social community (or group) determined by the environment, in most cases Social Networks, s/he belongs to. In order to contrast this hypothesis we have designed and evaluated a set of methods that include social factors to different existing group recommendation techniques. This set of methods form our *Social Recommendation Model* (*SRM*), which is this Thesis main contribution. In it, we simulate people's behaviour based on the corroborated idea that the relationship between individuals and their networks of people directly influence their lives (Christakis and Fowler, 2011). In addition to the development of these methods from a functional point of view, we propose a generic architecture named ARISE (*Architecture* *for Recommendations Including Social Elements*) that can be used to instantiate the proposed *SRM* in concrete recommenders of different domains and applications. Besides, we have provided a software development methodology to ease the instantiation of the ARISE architecture into concrete social recommender applications. This methodology, following a previous research line (Díaz-Agudo et al., 2007), is based on templates (Quijano-Sánchez et al., 2013b) that formalize the functional behaviour of social recommender systems and facilitate their configuration and deployment. The common and key factor in all the different types of recommenders that can be built in different domains using this generic architecture, is the inclusion of social factors. To illustrate and validate the improvement in the recommendation outcome due to the usage of our *SRM* and the viability of our generic platform ARISE, we have instantiated it into two real-life recommender systems: the group recommender *HappyMovie*[5], which is a particular instantiation of our generic architecture ARISE for the movies recommendation domain in the social network Facebook. And, the individual recommender *HappyShopping*, a Facebook application which is other particular instantiation of ARISE that follows our development methodology based on templates this time, in the clothing domain. Besides, *HappyMovie* also serves us as a use case and experimental environment where we are able to evaluate our ARISE architecture and our *SRM* with real data.

1.1 Hypothesis and goals

When organizing this PhD Thesis's research we formulated a series of hypothesis (that break down the general hypothesis previously described) that we wanted to prove and, respectively, a series of goals to fulfil in order to validate these hypothesis. These goals have led to different contributions in their area. These contributions are reflected in the published papers that we here present as core of this work (Part III of this document).

Next, we break down the posed hypothesis and its corresponding goals, the contributions that resulted from them and the papers that collect those contributions:

Hypothesis 1 (H1)
There is a need to improve group recommender systems by better modelling decision-making processes, possibly through the inclusion of social factors.

Goal 1 (G1)
Study the elicitation and usage of social factors in group recommendation processes and their ability to ease group decisions.

[5]http://gaia.fdi.ucm.es/research/happymovie

- **Contribution 1**: Study of existing recommender systems and different group and individual recommendation techniques.

  Supported by papers in:

  - Chapter 2. State-of-the-art.

- **Contribution 2**: Study of social factors in recommender systems and evaluation of Social Networks and the information that can be extracted from them.

  Supported by papers in:

  - Chapter 2. State-of-the-art.

- **Contribution 3**: Identification and study of people's group behaviour, in relation to conflict solving, according to their personality.

  Supported by papers in:

  - Chapter 8, (Quijano-Sánchez et al., 2009).
  - Chapter 10, (Quijano-Sánchez et al., 2010).

- **Contribution 4**: Identification of social factors that influence in people's trust and how to elicit them from Social Networks.

  Supported by papers in:

  - Chapter 10, (Quijano-Sánchez et al., 2010).
  - Chapter 18, (Quijano-Sánchez et al., 2013c).
  - Chapter 23 (Quijano-Sánchez et al., 2014b).

- **Contribution 5**: Identification of additional social factors that influence in group decision-making processes.

  Supported by papers in:

  - Chapter 12, (Quijano-Sánchez et al., 2011d).
  - Chapter 19, (Recio-García et al., 2013).
  - Chapter 13, (Quijano-Sánchez et al., 2011c).

Hypothesis 2 (H2)
It is possible to develop group recommender systems that model groups' social behaviour by including social factors.

Goal 2 (G2)
Development of our *SRM* through the inclusion of the social factors identified in the previous goal.

- **Contribution 6**: Proposal of recommendations based on delegation, DBR (Delegation-Based Recommendations).

  Supported by papers in:

  - Chapter 18, (Quijano-Sánchez et al., 2013c).
  - Chapter 22, (Quijano-Sánchez et al., 2014a).

- **Contribution 7**: Proposal of recommendations based on influence, IBR (Influence-Based Recommendations).

  Supported by papers in:

  - Chapter 18, (Quijano-Sánchez et al., 2013c).

- **Contribution 8**: Proposal of recommendations based on Coalitions.

  Supported by papers in:

  - Chapter 12, (Quijano-Sánchez et al., 2011d).

- **Contribution 9**: Proposal of recommendations based on Distributed Models and Argumentation.

  Supported by papers in:

  - Chapter 9, (Recio-García et al., 2010).
  - Chapter 19, (Recio-García et al., 2013).

- **Contribution 10**: Proposal of recommendations based on memory of past recommendations.

  Supported by papers in:

  - Chapter 13, (Quijano-Sánchez et al., 2011c).
  - Chapter 18, (Quijano-Sánchez et al., 2013c).
  - Chapter 23, (Quijano-Sánchez et al., 2014b).

- **Contribution 11**: Proposal of recommendations that solve the *cold-start* problem.

  Supported by papers in:

  - Chapter 16, (Quijano-Sánchez et al., 2012b, 2013a).

- **Contribution 12**: Proposal of social recommendations based on CBR (Cased-Based Reasoning).

  Supported by papers in:

  - Chapter 17, (Quijano-Sánchez et al., 2012a).

- **Contribution 13**: Evaluation of our *SRM* for the different existing aggregation strategies.

  Supported by papers in:

  - Chapter 14, (Quijano-Sánchez et al., 2011a).
  - Chapter 22, (Quijano-Sánchez et al., 2014a).

- **Contribution 14**: Evaluation of the proposed methods.

  Supported by papers in:

  - Chapter 22, (Quijano-Sánchez et al., 2014a).

Hypothesis 3 (H3)
It is possible to generalize our *SRM* in a way that it is applicable to different domains and in a way that other recommender systems developers can reuse it.

Goal 3 (G3)
Provide a generic architecture and a development methodology that allows the instantiation of our *SRM* in different domains.

- **Contribution 15**: Proposal a reusable generic architecture: ARISE.

  Supported by papers in:

  - Chapter 22, (Quijano-Sánchez et al., 2014a).
  - Chapter 21, (Quijano-Sánchez et al., 2013b, 2014c).

- **Contribution 16**: A semi-automatic instanciation of the ARISE architecture through the usage of *Social Recommenders Design Templates*.

  Supported by papers in:

  - Chapter 21, (Quijano-Sánchez et al., 2013b, 2014c).

Hypothesis 4 (H4)
It is possible to validate and evaluate our generic architecture ARISE through different concrete applications in different domains.

Goal 4 (G4)
Development of an application in a social network that validates our *SRM*, in the movies domain for group of users and in the shopping domain for individual users in social environments.

- **Contribution 17**: Development of an application in the social network Facebook that implements ARISE and the proposed social recommendation techniques of our *SRM*: *HappyMovie*.

  Supported by papers in:

  – Chapter 11, (Quijano-Sánchez et al., 2011e).

  – Chapter 15, (Quijano-Sánchez et al., 2011b).

  – Chapter 23, (Quijano-Sánchez et al., 2014b).

- **Contribution 18**: Development of an application in the social network Facebook that proves that the ARISE architecture is viable for other domains and that the proposed *Social Recommenders Design Templates* ease the develop of new social applications: *HappyShopping*.

  Supported by papers in:

  – Chapter 21, (Quijano-Sánchez et al., 2013b, 2014c).

1.2 Thesis structure

This PhD Thesis has been organized around the goals (G1-G4) introduced in the previous section. Hence, each chapter (3 to 7) corresponds to one of the posed goals and each section corresponds to a summary of the resultant contribution (publication) presented in Part III of the Thesis. Sections summarize the content of the papers cited in them and papers represent the research that has been carried out in order to fulfil each goal.

In Chapter 2, we first describe a theoretical framework of recommendations in general and group recommendations in particular. This chapter presents a general view of group recommenders and revises several related previous works (including among others works that outline the benefits of studying social ties and using social networks and the information stored in them). In Chapter 3 we perform a study of how to improve existing group recommendation techniques where we review the benefits of including social factors in group recommendations processes along with some proposals on how to elicit social factors (G1). In Chapter 4, we introduce our *SRM* and present different proposals of designed group recommender systems that use it (G2), these are the different methods that form the model. Besides, we present several experiments that evaluate and validate our *SRM*. The next point that we survey in Chapter 5, is reusing and generalizing our method through the design of a generic architecture and a development methodology based on templates of social recommenders (G3). Next, as proof of concept, in Chapter 6 we present a case study –*HappyMovie*– in the movies domain where the new studied recommendation techniques are applied and an evaluation of their efficiency is presented. Following this a second case study is presented in a different domain, this allows us to prove the viability and reproducibility of our generic architecture and its corresponding associated methodology (G4). Finally, in Chapter 7 conclusions reached through this research work are presented as well as future lines of work.

Following the previous explanation this Thesis work is structured in the following chapters:

Chapter 2. State of the art. This chapter presents a general vision of individual and group recommender systems, followed by an explanation of the importance of Social Networks in the last few years and how, from them, we can extract social information that improves the results of group recommender systems. Last, some examples of ongoing works in the social recommendation area are presented.

Chapter 3. Study of the elicitation and usage of social factors in group recommendation processes and of their ability to ease group decisions. In this chapter we verify hypothesis H1 by fulfilling goal G1. To do so, we identify the different social factors that serve as building blocks of our *SRM*, we motivate the reasons of including them and present some ideas of how to elicit them.

Chapter 4. Development of the social group recommendation methods that form the *SRM*. In this chapter we verify hypothesis H2 by fulfilling goal G2. To do so, we present our *SRM* and several different approaches that use it.

Chapter 5. Generic architecture and development methodology for the instantiation of the model. In this chapter we verify hypothesis H3 by fulfilling goal G3. To do so, we present our generic architecture ARISE and our *Social Recommenders Design Templates*, that represent the steps that must be followed in order to reuse our *SRM* with different data and/or domains.

Chapter 6. Use Cases in a Social Network. In this chapter we verify hypothesis H4 by fulfilling goal G4. To do so, we detail our social application *HappyMovie*. This application is a movie group recommender system integrated in the social network Facebook whose goal is to present a set of movies that the system predicts will be of the group's liking according to the individual preferences of the group's members and their social relationships. Finally, a secondary case study, *HappyShopping*, is presented this time in the clothing domain. With this second social application we study the benefits and viability of using ARISE and the *Social Recommenders Design Templates* in a different domain to the one they were designed in.

Chapter 7. Conclusions and Future Work. In this chapter we present the conclusions that we have reached after the fulfilment of this PhD Thesis and propose some future lines to continue our research.

Chapter 2

State of the art

2.1 Recommender systems

Recommender systems are born with the aim to facilitate decision-making in domains/areas where choice possibilities are many and varied. They have the effect of guiding users in a personalized way to interesting objects to buy or consume in a large space of possible options (e.g. see (Adomavicius and Tuzhilin, 2005; Ricci et al., 2011) for an overview). Nowadays, we can find recommenders for all sorts of products: holidays, books, movies, restaurants, cars and much more (Jung, 2012; McCarthy, 2002; Batet et al., 2012; Vaz et al., 2012; Jameson, 2004). Undoubtedly, the greatest application field for this kind of systems is leisure activities. Some of these activities are typically carried out in groups of people instead of individually, therefore, it makes sense to recommend not only to individuals but to whole groups of people that wish to perform a joint activity (Jameson and Smyth, 2007). Hence, the fact that these two types of recommenders exist (individual and group).

In this chapter we present different types of recommender systems, the differences between them according to their goals and design and the shortcomings they suffer, fact that has motivated the fulfilment of this Thesis. Right after, we introduce the concept of social factors, whose inclusion in group recommender systems will be this Thesis' main contribution. To do so we review the impact of Social Networks and Social Media in the last few years, their uses, the information that can be extracted from them and the use that other recommender systems have made of them.

2.1.1 Individual recommender systems

In the literature there are two main classes of individual recommenders, the difference between them lies in the source of their knowledge: *content-based* recommenders, perform recommendations according to the description of the items to recommend (Pazzani and Billsus, 2007), and *collaborative* recommenders exploit users' retrieved information along with their ratings about items (Bridge et al., 2005). Between these two opposite ends there are several *hybrid* approaches (Burke, 2002) that combine the techniques used in *collaborative* and *content-based* recommendations.

These recommendation approaches (*collaborative*,*content-based* and *hybrid*) have different strengths and weakness. A common weakness most recommenders present is the so called *cold-start problem* (Herlocker, 2000; Schafer et al., 2007b). This problem occurs when the system does not have enough information about a new user in order to deduce anything about their taste. Another weakness, in this case only related to *content-based* recommenders, that some systems often present is recommendations that are too *uniform*. For example, Amazon's recommender system initially suffered from a *portfolio effect* (Linden et al., 2003; Burke, 2002), i.e. offered recommendations so similar they were of little use to the user. An overview of these and other weakness different recommenders present can be found in Tintarev's Thesis (Tintarev, 2009).

Individual recommenders can also be classified according to the following characteristics:

- Based on who takes the initiative. We can divide recommender systems in two different types according to who takes the initiative in the recommendation. We can therefore have *reactive* recommenders, where users seize the initiative by raising a query to the system. An example of this type of system is TAAABLE (Cordier et al., 2012), that recommends cooking recipes that may satisfy the user after s/he has made an initial query. On the other hand, we find *proactive* recommenders, in them the recommender is the one that takes the initiative, making an initial suggestion to the user based on users' past history, item ratings, or any other previously selected strategy (McGinty and Smyth, 2003a).

- Recommendation dynamics. Here we also distinguish two different types of recommenders: *single-shot* and *conversational* (Smyth, 2007). *Single-shot* are those recommenders that only return a single set of suggestions in a given session, once presented, the user will be able to choose or discard them. If the given recommendation is not of the user's liking s/he will have to start all over again and ask for alternative items. *Conversational* recommenders are those that adopt an iterative approach. Users can elaborate their requirements, as part of an extended recommendation dialog, until they reach a fitting item. Different forms of *conversational* recommender systems can be distinguished by the way they elicit user requirements (Shimazu, 2001, 2002). For example, *navigation-by-asking* recommenders ask users a series of questions regarding their requirements. Systems may also or alternatively show the users particular products and elicit requirements in the form of feedback on the proposed products, this type of strategy is called *navigation-by-proposing*.

- Ability to personalize. Some recommenders make an special effort to include in recommendation process users' characteristics, preferences and/or needs (de Gemmis et al., 2011). Users' profiles can contain information related to navigation history, preferences, users' needs or any other information that the system considers advisable to store. The ability to personalize is related to how the system manages (if at all) this information, that is if the recommendation strategy uses or takes into account this information when producing an outcome.

- Quality of the recommended items. There are recommenders that follow traditional similarity approaches and recommenders that commit for innovation in the similarity measures introducing a measure of quality in the recommended items (McGinty and Smyth, 2003b). This measure is related to *diversity* in the retrieved items to recommend. Diversity in a retrieved set is defined as the existing dissimilarity between each pair of items in the set. Recommenders that focus on improving the recommendation quality consider that an item's quality is improved in as much dissimilar it is with the already retrieved items, as long as it is still similar to the user's raised query.

Next, we take a closer look at *collaborative* and *content-based* recommender systems.

2.1.1.1 Collaborative recommender systems

Collaborative recommenders (Ekstrand et al., 2011; Koren and Bell, 2011; Herlocker et al., 2002; Candillier et al., 2007) are those that do not require item descriptions, as they use the ratings already assigned by the users to several products. In general, collaborative filtering (CF) is the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources, etc. Collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). The main drawback that this recommendation technique presents is the necessity of obtaining user ratings for the items that are to be recommended. To solve this problem, *Long tail* techniques have frequently been used. These techniques refer to the statistical property of the larger share of user population, that rests within a distribution tail similar in shape to the Gauss distribution. This technique has been used by Amazon[1] and

[1] `http://www.amazon.com/`

Netflix[2] among other comercial applications. The number of needed ratings is reduced if user profiles are provided, in which case the recommender uses this information to perform a recommendation to the active user according to the ratings of other users with a similar profile. This allows the system to recommended to the active user items that are not in her/his profile but in the profile of similar users. Collaborative filtering systems are usually categorized into two subgroups: *memory-based* and *model-based* methods.

Memory-based methods, memorize the rating matrix and issue recommendations based on the relationship between the queried user and item and the rest of the rating matrix. On the other hand, *Model-based* methods fit a parameterized model to the given rating matrix and then issue recommendations based on the fitted model. The most popular *memory-based* methods are *neighborhood-based* methods, which predict ratings by referring to users whose ratings are similar to the queried user (*user-based recomendations*, (Breese et al., 1998)), or to items that are similar to the queried item (*item-based recomendations* (Sarwar et al., 2001)). These techniques are motivated by the assumption that if two users have similar ratings on some items they will have similar ratings on the remaining items. Or alternatively if two items have similar ratings by a portion of the users, the two items will have similar ratings by the remaining users. *Neighborhood* methods vary considerably in how they compute the weighted average of ratings. Specific examples of similarity measures that influence the averaging weights are *Pearson correlation* (Herlocker, 2000), *Vector cosine*(Pham et al., 2011), and *Mean-Squared-Difference* (MSD) (Herlocker et al., 2002).

Model-based methods, on the other hand, fit a parametric model to the training data that can later be used to predict unseen ratings and issue recommendations. *Model-based* methods include *cluster-based* CF (Ungar and Foster, 1998; Connor and Herlocker, 2001), *Bayesian classifiers* (Miyahara and Pazzani, 2000), and regression based methods (Vucetic and Obradovic, 2005). The *Slope-One* method (Li et al., 2011; Lemire and Maclachlan, 2007) fits a linear model to the rating matrix, achieving fast computation and reasonable accuracy. A recent class of successful CF models are based on low-rank matrix factorization. The regularized SVD method (Billsus and Pazzani, 1998; Karatzoglou and Weimer, 2010) factorizes the rating matrix into a product of two low rank matrices (user-profile and item-profile) that are used to estimate the missing entries. An alternative method is Non-negative Matrix Factorization (NMF) (Lee and Seung, 2000) that differs in that it constrains the low rank matrices forming the factorization to have non-negative entries. Recent variations are Probabilistic Matrix Factorization (PMF) (Salakhutdinov and Mnih, 2007), Bayesian PMF (Salakhutdinov and Mnih, 2008) or Non-linear PMF (Lawrence and Urtasun, 2009) among others.

[2] `http://www.netflix.com/`

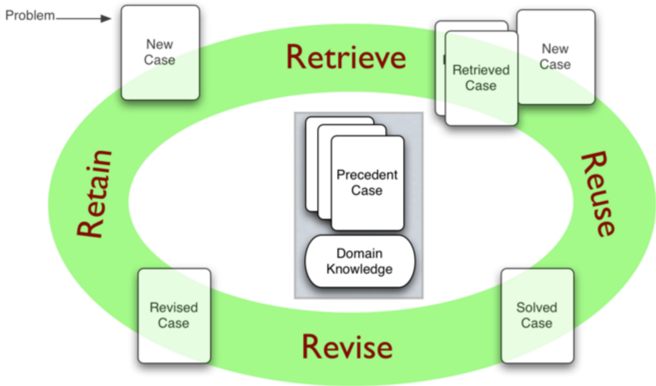


Figure 2.1: CBR cycle, Aamodt & Plaza 1994.

In contrast to *collaborative* recommenders we find *content-based* recommenders, which we next proceed to describe.

2.1.1.2 Content-based recommender systems

Content-based recommenders (Lops et al., 2011), use descriptions of the items to be recommended and provide a recommendation outcome formed by the set of items whose descriptions are more similar to the users' requirements. *Cased-Based* recommenders are a special type of *Content-based* recommenders in which each retrieved case represents a previous recommendation of a similar item to the one the users seeks. These type of recommenders based themselves in cased-based reasoning techniques (CBR), see Figure 2.1. CBR systems (Leake, 1996) retrieve a case base formed by previously solved problems and use it to solve similar problems.

Next, we explain the differences between individual recommenders and the recommenders studied in this Thesis, group recommenders.

2.1.2 Group recommender systems

Recommender systems have traditionally recommended items to individual users, but there has recently been a proliferation of recommenders that address their recommendations to groups of users (Jameson and Smyth, 2007). These systems raise a number of challenging issues, like retrieving group preferences, helping groups' decision-making process and explaining to the group the system's reasons for suggesting an item. In reference to information retrieval methods, most group recommenders developed so far apply methods for acquiring information about users' preferences that are barely distinguishable from the methods applied in recommender systems for individuals. These methods can be basically divided in:

- Acquisition of Preferences Without Explicit Specification. Many recommender systems do not require their users to specify their preferences explicitly. These systems get by with implicitly acquired information about users. An example is Let's Browse (Lieberman et al., 1999), which recommends web pages to a group of two or more persons who are browsing the web together.

- Explicit Preference Specification. Other types of group recommenders do require an explicit specification of preferences. An example is Travel Decision Forum (Jameson et al., 2004) that helps a group of users to agree on the desired attributes of a vacation they are planning together. A less explicit form of preference specification is found in PolyLens (O'Connor et al., 2001), a system that recommends movies to groups of users. It is an extension of the individual movie recommender system MovieLens (Schafer et al., 2007a), which is based on collaborative filtering, where users do not explicitly describe their movie preferences, but they do rate individual movies on a scale from 1 to 5 stars. This system produces group recommendations through the aggregation of group members' preferences, using a minimizing misery strategy that stands for the hypothesis that a group's happiness is equal to the happiness of its least happy member.

Once the system acquires the required knowledge about users' preferences, several different types of group recommender systems appear depending on the way they choose to manage these preferences and according to the specific goals and characteristics of the system. Examples of differences when managing users' preferences are: systems that focus on negative preferences, like in Adaptive Radio (Chao et al., 2005), or: systems that share information about the specified preferences as we can see in Travel Decision Forum (Jameson et al., 2004) or CATS (McCarthy et al., 2006). The first type only makes sense when the selected recommendation approach is mainly designed to avoid items that are specially damaging for a particular group member. The second type represents systems where in a group recommendation is useful that each member has a notion of the other members' preferences. For example, to learn from other group members, to save time when having to specify one's preferences, to understand other member's reasons and reach a consensus more easily or to be able to foretell other group members' behaviours. Easily we can see that with this approach a manipulation problem raises, for example, if a user does not want a particular item to be recommended, s/he can rate it as "hate it" and ensure with this action that the item is never a recommendation outcome. This situation appeared with MusicFX as the authors described in (McCarthy and Anagnost, 1998).

Depending on the size and homogeneity of the group, finding a recommendation that satisfies each group member individually can be a difficult task (Jameson, 2004). In most cases, the recommender must choose an option that satisfies as many group members as possible according to their individuals preferences. Therefore, some type of aggregation method is required, by which information about individual preferences is combined in such a way that the system can assess the suitability of particular items for a group as a whole. The need to choose an aggregation method is the most obvious and intensively studied difference between group recommendation and recommendation for individuals. The topic of preference aggregation is a multifaceted and complex one that has been addressed in various scientific fields. Although the various approaches differ in the ways in which they gather and represent users preferences, almost all approaches make use of one of these three schemas: (a) merging of sets of recommendations. Used in cases where the outcome to be presented is a set of candidate solutions, among which the group is supposed to select one. The common procedure in this case is a simple aggregation method consisting of generating a small number of recommended solutions for each member and then merging them (e.g. using the union or the intersection) in a single list. This method was one of those considered for the PolyLens system (O'Connor et al., 2001); (b) aggregation of individual ratings for particular items. For each candidate item and group member, the system predicts how that particular member would rate the selected item, then it aggregates all the item's predictions into a single rating and returns the set of candidate items with the highest predicted ratings. This is the most popular approach when designing group recommenders and therefore it is the one we have taken in our approach. In the next subsection we will review some of the different aggregation functions this type of systems adopt. An example of this behaviour can be found in Pocket RestaurantFinder (McCarthy, 2002); and (c) construction of group preference models. In this approach, the system somehow uses its information about each group members' preferences to arrive at a model of the preferences of the group as a whole. Let's Browse (Lieberman et al., 1999) is an example of this type of systems.

2.1.2.1 Group recommender systems that use aggregation of preferences

Once a general approach is chosen (merging recommendations, preference aggregation or construction of group preference models), if it is preference aggregation, we must select a particular computational procedure (or mechanism) for the aggregation. There are different goals that could be desirable and therefore taken into account when choosing an aggregation function (e.g. total satisfaction, fairness, or comprehensibility). Besides, conflicts between them can easily arise. Hence, the chosen aggregation function will depend on the necessities of the system in each particular moment. Next, we present some examples, note that in them $\hat{r}_{u,i}$ is the rating prediction of item i for each user $\{u : 1 \dots n\}$ in the active group G_a and $\hat{r}_{G_a,i}$ is the final rating that item i gets for G_a:

- *Average satisfaction.* This aggregation function computes the average of the predicted satisfaction for each member and is used as a basis for the selection of candidates. The function is as follows:

$$\hat{r}_{G_a,i} = 1/n * \sum_{u=1}^{n} \hat{r}_{u,i} \tag{2.1}$$

- *Least misery.* Even if the average satisfaction is high, a solution that leaves one or more members very dissatisfied is likely to be considered undesirable. In PolyLens (O'Connor et al., 2001) the minimization of misery is the only criterion applied. It is also possible to take this factor into account as a constraint that must be fulfilled by a solution: The lowest predicted rating must not fall below a given threshold. The function is as follows:

$$\hat{r}_{G_a,i} = \min_{u \in G_a} \hat{r}_{u,i} \tag{2.2}$$

- *Ensuring Some Degree of Fairness.* This aggregation function tries to find a solution that satisfies everyone equally well. This situation is in general preferred to one that satisfies some at the expense of others. The function is as follows:

$$\hat{r}_{G_a,i} = 1/n * \sum_{u=1}^{n} \hat{r}_{u,i} - \omega * standardDeviation(\hat{r}_{u,i}) \tag{2.3}$$

  Where ω is a weight that reflects the relative importance of fairness.

These strategies have been criticized (Chen et al., 2008; Masthoff and Gatt, 2006) because they combine users preferences always in the same way, without taking into account real life interactions between group members. Our approach focuses on mitigating this problem. Other solutions to this problem are the ones proposed in (Chen et al., 2008), where the authors use genetic algorithms to establish the ideal weight that each individual rating should have in the final group rating prediction. This solution has some obvious draw backs, for example the necessity of having previous group ratings for other items. Or the ones presented in (Masthoff and Gatt, 2006), where the trend of some recent works in including the social aspects of group members' relationships when making group recommendations is included. This work reflects the idea of combining individual satisfaction and *emotional contagion* in order to recommend a sequence of video clips for a group. The authors think that a member changes the selection of her/his best clip according to the clip selected during the previous selection step. This change can be reflected in the recommendation algorithm as an individual satisfaction function that computes the individual affective state. This state influences the affective state of the other members, producing an emotional contagion that should be taken into account during the recommendation process.

Other factor that varies preference aggregation is the inclusion of different profiles that reflect the personality of the users involved in the group recommendation process (Recio-García et al., 2009). In this work, forerunner of this Thesis, a group recommendation method that takes into account the different personalities of group members is proposed. In the final group recommendation, users' individual preferences have different weights according to how each user would react in a conflict situation.

Once the system provides a recommendation and given the many ways in which group recommendations can be produced - derived and often conflicting as different goals can be pursued - it is natural that group members should want to understand to some extent how a recommendation was arrived at - and in particular, how attractive a recommended item is likely to be to each individual group member -. Hence, some recommender systems provide along with the recommendation itself an explanation of how it is reached (Herlocker et al., 2000; Tintarev, 2007). An example of this type of system, that uses explanations to justify the proposed outcome, is Let's Browse (Lieberman et al., 1999). Explanations in recommender systems present multiple variations, they can go from a simple confidence value to a complex visualization of the pros and cons of a solution. However, there is no guarantee that the proposed recommendations would be followed, no matter how suitable or convincing the system's explanation or recommendation is. With group recommendations, extensive debates and negotiations may be required, which may be especially problematic if the members are not able to communicate easily. To this end some systems chose to provide not a single solution but a platform for group members to argue and reach an agreement as explained in (Jameson et al., 2004).

Summing up, we conclude that there is a need to adapt recommendation processes to group composition (Jameson and Smyth, 2007; Masthoff, 2004). This is backed up by some recent works that have focused their studies on analyzing the effectiveness of group recommendations according to different aspects, such as group size or inner group similarity (Baltrunas et al., 2010), and on studying different weighting models to combine the preferences of group members according to their activity or role within the group (Berkovsky and Freyne, 2010). Our proposal continues this general belief of adapting and improving group recommendations through the inclusion of factors that add more information about the group itself, group composition and group members. More concretely, we have based our research in the study of Social Networks, the social factors and trust models that can be inferred from them and how these factors can help improve group recommenders. The value of including social information to recommender systems has now started to be acknowledged by the recommender research community and has produced a new type of recommenders named social recommenders.

2.1.2.2 Evaluation metrics in group recommender systems

Group recommenders require evaluation functions that measure the accuracy recommendations they provide. This is a key matter in order to validate the produced outcomes. To achieve this, a common goal is to compare the results obtained by the group recommender system $rec(G_a)$ to the real group preferences/choice $fav(G_a)$ (Quijano-Sánchez et al., 2014a). In order to choose a suitable evaluation metric there are several factors to be taken into account:

The first one is the length limitation in the $fav(G_a)$ list. Real groups of users are only interested in a few items they really want to consume and consequently we need to limit the list of proposed items (Shi et al., 2012). Therefore, it is not advisable to use general measures like recall or precision (Billsus and Pazzani, 1999; Maybury et al., 2004). The second factor to be considered is wether $rec(G_a)$ in the chosen domain (movies, clothes, etc) is or not an unordered set. For example, in our particular case we will focus on recommenders that propose a set of k items without any kind of ranking -these are afterwards voted by the members of the group to make a final decision-. This feature discards several evaluation metrics that compare the ordering of the output and validation lists like the Mean Absolute Error (MAE) (Herlocker et al., 2004; Adomavicius and Tuzhilin, 2005) or the Normalized Discounted Cumulative Gain (nDCG) (Baltrunas et al., 2010). In that case, where ordered lists are not considered, there are some metrics used in the Information Extraction field (Tomlinson, 2006) that are suitable.

For example, *precision@n* evaluates how many of the items in $rec(G_a)$ are in the $fav(G_a)$. This kind of evaluation can be seen from a different point of view: we are usually interested in having at least one of the items from $rec(G_a)$ in the $fav(G_a)$ list. This measure is called *success@n* and returns 1 if there is at least one hit in the first n positions (Quijano-Sánchez et al., 2013c). Therefore, *success@n* (or simply $s@n$) can be used to evaluate group recommender systems by computing the rate of recommendations where there is at least one hit in $fav(G_a)$. For example, 90% accuracy using $s@n$ represents that the recommender suggests at least one correct item for 90% of the groups being evaluated. In fact, $s@n$ is equivalent to having $precision@n > 1/n$. A $2s@n$ metric could also be defined (equivalent to $precision@n > 2/n$), which represents how many times $fav(G_a)$ contains at least two items from $rec(G_a)$. Obviously, it is a much more restrictive measure.

Next, we review the increasing importance of Social Networks, the useful information for the recommenders community that can be extracted from them and some of the ongoing work in social recommendation.

2.2 Social factors in recommender systems

Online Social Networks, such as Facebook, provide a wealth of information that we can leverage for recommending a variety of artifacts, such as news articles, movies, books, etc. While recommender systems have been extensively researched since the mid-1990s (McCarthy and Anagnost, 1998; Lieberman et al., 1999), the study of social-based recommender systems is a new area (Jiliang Tang and Liu, 2013). One of the key factors that social-based recommendations use is the study of the multiple dimensions within a user's social network. Within these dimensions, social trust, users' interests and user similarity stand out. In (Gartrell et al., 2010), authors leverage these dimensions seeking to develop novel ensemble recommender systems. Another example of social recommenders are the ones that study social relations; various approaches are proposed to build social recommender systems such as trust ensemble (Ma et al., 2009), trust propagation (Jamali and Ester, 2009), or directly trust user based recommenders (Zhang et al., 2009).

There is recent work reporting significant recommendation performance improvement for social recommender systems (Golbeck, 2006b; Nepal et al., 2012; Pera et al., 2010; Yang et al., 2012; Bao et al., 2012; Hu et al., 2012). On the other hand, there are also unsuccessful attempts at applying social recommendation (IBM, 2012; Quora, 2012). As we will next see, although there are several works that concurrently to our research are performing studies focused on social recommenders research, most of them have focused their goals in using these social tools to assist them in the concrete case of individual recommendations and miss the fact that Social Networks are able to capture many aspects of groups' social behaviour, information that can help enhance group recommenders. There are works that do focus on social group recommendation, like for example (Gartrell et al., 2010). In this work, a theoretical model is presented where they use social information like social ties or expertise levels to perform group recommendations. However, they are limited by the necessity of obtaining this social information through long questionnaires (most of them face-to-face and guided). This is not effective when proposing a usable tool for groups of users. It is for this reason that in this Thesis we have wanted to take one step forward in recommender systems research and use the information that can be found in nowadays Social Networks to infer different social factors (mainly personality and trust in our case) and use them along existing group recommendation techniques to design an application in a Social Network that automatically performs recommendations to groups of users through the creation of events. This way we are able to offer an easily and dynamic accesible service (requiring minimum effort from the user when having to provide information).

Next, we will firstly present a brief introduction to existing Social Networks and how they are a perfect environment to generate trust models, whose use in recommender systems has been one of the forefathers of social recommender systems.

2.2.1 Social networks

Social Networks are online communication platforms where content is created by users through the use of Web 2.0 technologies (making easier tasks like editing, publishing and exchanging information). Similarly, Social Media refers to interaction among people in which they create, share, and/or exchange information and ideas in virtual communities and networks. Generically we could speak of a Social Web that harbours Social Networks and contains social information.

Social Networks' offered services focus on reflecting and building social relationships between people that for example share common interests and/or activities. Essentially they consist in user representation, commonly through a profile, her/his social connections and a variety of additional services.

From our research line point of view, Social Networks provide a measure of trust between the different users that use them and allow us to picture a whole network structure build between them. A connection between users in a Social Network symbolises an affinity between them inside the network's theme. Social Networks like Facebook[3], LinkedIn[4], Instagram[5], Twitter[6] or Lastfm[7] (among many others), have as goal information exchange between users. Their themes vary from work related, to pictures or music exchange. Social Networks' expansion peek has occurred in the last few years, matching Internet's expansion where they have gained great importance[8]. Users inside Social Networks search for places to find people who they can relate (similar to them) and share ideas within the network's theme. Social Networks are mainly divided by:

- **Target users and theme**:

[3] `http://www.facebook.com`

[4] `http://www.linkedin.com`

[5] `http://instagram.com`

[6] `https://twitter.com`

[7] `http://www.last.fm/`

[8] Currently, there are more than 200 Social Networks (see `http://en.wikipedia.org/wiki/List_of_social_networking_websites`) and almost 2 billion of users (see `http://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/`).

  - *Horizontal social networks*: Are those aimed at all types of users without a defined theme. Facebook and Twitter are the most representative examples.
  - *Vertical social networks*: Are those conceived in an aggregating theme axis. Their goal is to bring together a concrete collective around a defined theme. Inside this type there are *profesional* social networks like LinkedIn, or *leisure* social networks like Lastfm.

- **The main subject of the relation**:
  - *Human social networks*: Are those whose main goal is to promote human relationships, bringing together users according to a social profile built through users likes, hobbies, trips and/or activities. These social networks are built in a more intimate basis, fact that allows the inference of trust models and/or social contagion. Examples of this type of networks are Facebook, Tuenti or Google+.
  - *Content social networks*: Are those networks where relationships are formed by joining profiles through published content, items the user has or files that are in her/his computer. Twitter and Flickr are the most representative examples.

Multidisciplinary research has proved that Social Networks operate at many levels. In its simplest form, a Social Network is a map where all relevant ties between nodes are studied. In this case we speak of "sociocentric" or "complete" networks. Another option is to identify the network surrounding a user (inside the different social contexts where s/he interacts); in this case we speak of "personal network". Social Network analysis is being used for a wide set of different matters where the information stored in them is being used to build systems with different goals:

- Social Networks' evolution can sometimes be simulated through agent based models that provide information about communication patterns, rumour propagations and the social structure itself (Yahja and Carley, 2005).

- Diffusion of innovations is a theory that seeks to explain how, why, and at what rate new ideas and technology spread through cultures (Rogers, 1983). It explores Social Networks and their role and influence in new ideas' broadcasting. The theory says that diffusion is the process by which an innovation is communicated through certain channels over time among the members of a social system. The origins of the diffusion of innovations theory are varied and span multiple disciplines including Social Networks.

- (Fowler and Christakis, 2008) study assures that there is an interrelationship between happiness and Social Networks. When an individual is happy, people close to her/him have a 25 per cent higher chance to also being happy. Besides, people that belong to the very center of a Social Network tend to end up being happier than those who belong to the outskirts.

- In McDonald (2003) the author shows different systems that use Social Networks as a tool to recommend users to collaborate with or as a tool to use Social Networks visualizations to take advantages of the different collaborations that can appear inside a work environment.

Next we explain one of the many uses that information stored in Social Networks can have, the creation of trust models.

2.2.2 Trust models

Current research has pointed out that people tend to rely more on recommendations from people they trust (friends) than on recommendations based on anonymous ratings (Sinha and Swearingen, 2001). This social element is even more important when performing group recommendations, where we try to help users choose an item to consume as a group. In real life situations, this kind of decision-making usually follows an argumentation process, where each user defends her/his preferences and rebuts other's opinions. Here, when users must settle to reach a common decision, trust among users is the major factor. Hence, it makes sense than when modeling decision-making processes in group recommendation systems this trust factor is taken into account.

There is a huge body of work about the generation of trust models. These studies have been boosted due to the raising of the current collaborative web (Web 2.0), that has encouraged the idea of Web Of Trust (WOT) (Golbeck, 2006b; O'Donovan and Smyth, 2005; Victor et al., 2008). The WOT represents trust among users, modelled on an online network. There are specific approaches that use a customed trust network to recommend items. One example is FilmTrust (Golbeck, 2006b), which exploits this type of trust networks in the movie preferences domain. (Golbeck, 2006a) presents a study of how to infer trust relations within Social Networks. The challenge of computing trust is to determine how much one person in the network should trust another. Certainly, trust inferences will not be as accurate as a direct rating. But in this work, Golbeck presents an algorithm for inferring trust in networks with continuous rating systems, named TidalTrust, that improves other trust computing algorithms' accuracy by 10%. However, these specific trust networks are quite difficult to generate because they require explicit feedback from users, and that, can generate rejection.

Another promising approach, and the one we take in this Thesis, is to infer knowledge of trust from existing Social Networks like Facebook or Twitter. These networks contain implicit information that can be exploited in order to improve recommendation processes. This option has the main advantage of being completely transparent to users. Users are not required to provide explicit information about their trust to other users because this knowledge is extracted implicitly from their daily interaction in the social network. However, it has the obvious drawback that every user involved in the group recommendation process must belong to the social network. Nevertheless, the rising popularity of this kind of Web applications minimizes this risk. Even more, it is becoming usual to organize events (like going to the cinema) through Social Networks, so, group recommendation techniques could be integrated into these Web sites, idea that this Thesis implements.

2.2.3 Social recommenders

Next we present some examples of recommenders that use social information in their recommendation processes. In Social Networks, profiles capture users' "intent". Using this information several recommender systems have been created to help users take decisions (Jiliang Tang and Liu, 2013). This type of social recommendations can go from recommending which people to trust based on Twitter information (Tavakolifard et al., 2013) to using LinkedIn social graph to recommend a company which people to hire (Posse, 2012). Another example is (Konstas et al., 2009), where they use the social information stored in the music social network *last.fm*, and capture explicitly expressed bonds of friendship as well as social tags to improve the recommendations' accuracy.

Users in today's online social networks often post a profile, consisting of attributes like geographic location, interests, etc. Such profile information is used on recommender systems as a basis for grouping users, for sharing content, and for suggesting users who may benefit from interaction. However, in practice, not all users provide these attributes. In (Mislove et al., 2010) they use the information stored in online social networks to infer the attributes missing in some users' profiles.

Other approaches use the concept of the Social Semantic Web that subsumes developments in which social interactions on the Web lead to the creation of explicit and semantically rich knowledge representations (Gruber, 2008). The Social Semantic Web can be seen as a Web of collective knowledge systems, which are able to provide useful information based on human contributions and which get better as more people participate. Conventionally, finding answers to questions and learning from the knowledge mine existing on the Social Web has primarily been a manual process (Golbeck, 2006b; O'Donovan and Smyth, 2005; Victor et al., 2008). It requires a lot of intelligence in sifting through the endless information in the Social Web. However, recent studies have developed recommenders that obtain this type of information automatically (Nachawati et al., 2012).

All these works take into account some of the different factors involved in our proposal: usage of social factors (Konstas et al., 2009; Tavakolifard et al., 2013), more concretely the use of social factors in group recommenders (Gartrell et al., 2010), trust evaluation in recommender systems (Ma et al., 2009; Jamali and Ester, 2009; Zhang et al., 2009), automatic elicitation of knowledge stored in Social Networks (Nachawati et al., 2012), etc. However, we have not found any work that integrates, evaluates, and automatically extracts from Social Networks social factors in group recommendation processes. Throughout this Thesis we will study how to use the information and resources that Social Networks offer to compute our social factors, how to use them to model group composition and how to use them to enhance group recommender systems. We will prove that by doing so we are able to build systems that better simulate group argumentations that take place in real life and therefore generate better recommendations, more similar to the decisions that groups would take in reality.

Summing up, social recommendation is still in the early stages of development, and there are many challenging issues needing further investigation. Following this reasoning we consider the necessity to discuss and propose new research directions that can improve social recommendation capabilities and make social recommendation applicable to an even broader range of applications.

After the study of the state-of-the-art performed in this chapter we posed this Thesis first hypothesis: ***"H1: There is a need to improve group recommender systems by better modelling decision-making processes, possibly through the inclusion of social factors."***. In next chapter we will validate this hypothesis and propose some initial ideas for our *Social Recommendation Model* (*SRM*).

Chapter 3

Study of the elicitation and usage of social factors in group recommendation processes and of their ability to ease group decisions

3.1 General vision

It is becoming trend to employ recommendation technologies to aid users in the task of finding interesting items in the Web (Mika, 2011). There is a wide range of products such as books, music, games, trips, etc, that are difficult to discover in the Web due to the overwhelming amount of information available. Recommender systems (Jameson and Smyth, 2007) enable users to find items and provide a richer and more interactive user experience than classical interfaces based on catalogues of products.

Initially, existing recommenders were focused on individual users (Ekstrand et al., 2011; Lops et al., 2011). However, nowadays the rise of the collaborative Web (a.k.a. Web 2.0) has encouraged the development of activity-planning through Social Networks, like watching a movie, going to a restaurant, listening to a radio station or travelling with friends. A clear example are events organized through Social Networks like Facebook. Here, recommender systems can play a significant role, since agreement on a common item by several users is not a simple task (even when it is addressed in face-to-face argumentations). To address this issue, the number of recommender systems that deal with the challenge of making recommendations for groups of people has increased (O'Connor et al., 2001; McCarthy, 2002). These systems, like LET'S BROWSE or FlyTrap (Lieberman et al., 1999; Crossen et al., 2002), commonly just aggregate real or predicted ratings for group members. However, we believe that group recommendation is not a mere aggregation of individual preferences. Statement that has being embraced in the last few years by several researchers in recommender systems (Jameson and Smyth, 2007; Masthoff, 2004), confirming our belief in the need to adapt recommendation processes to group composition.

Humans are social individuals and, therefore, social behaviour has a great impact on their group decision-making processes (Rutherford and Ahlgren, 1991; Goode, 2000; Wanga et al., 2006). Our proposal takes into account this fact and assumes that the general satisfaction of the group does not always mean aggregating its members' preferences. It is clear that groups have an influence on individuals when coming to a decision. This is commonly referred to as *emotional contagion*: the effect of individuals' affective state on others in the group (Masthoff and Gatt, 2006; Barsade, 2002; Hatfield et al., 1994). This *contagion* is usually proportional to the *tie strength* or *trust* between individuals as closer friends have a higher influence (Victor et al., 2008; Golbeck, 2006a; O'Donovan and Smyth, 2005). Besides, the principle of *homophily*, states that people tend to trust and be friends with people who they share interests with (Burt, 1982; Miller Mcpherson and Cook, 2001; McPherson and Smith-Lovin, 1987). However, the influence an individual inside a group also depends on the individual's degree of *conformity* (Masthoff and Gatt, 2006). It has been proved that humans adjust their opinions to conform with those of a group when the majority of the group expresses a different opinion (Masthoff and Gatt, 2006). The degree of conformity is counteracted by the individual's behaviour when facing a conflict situation. Here, *personality* influences the acceptance of others' proposals (Recio-García et al., 2009). Besides, it has been proven that individuals' decisions change upon receiving more sophisticated arguments and remain the same otherwise. This individual reaction upon *persuasion* is an essential part of understanding groups' aggregation of preferences and can explain the advantage of studying group behaviour as a whole entity over group members' individual behaviour in decision-making processes (El-Shinnawy and Vinze, 1998; Penczynski, 2014). Finally, the concept of *fairness/justice* (Liang et al., 2007; Masthoff and Gatt, 2006) in the long run should be taken into account. That is, to balance the items recommended and maximize the satisfaction of users whose previous recommendations were contrary to their preferences for the groups' sake.

Previous research on group recommendation considered the preferences of every member in the group with the same degree of importance and tried to satisfy the preferences of every individual (Jameson and Smyth, 2007; Baltrunas et al., 2010; Berkovsky and Freyne, 2010; Pessemier et al., 2013). However, we believe that all these social elements (emotional contagion, trust, personality, ...) should be included in the recommendation model to fully represent the group behaviour when choosing a shared item. Although it seems natural to model this social knowledge, a major limitation appears: social factors are very difficult to estimate. Up to now, it was impossible to obtain these factors without annoying users with intrusive methods such as several questionnaires. But nowadays the Collaborative Web provides a tool that can be used to lighten this problem: Social Networks. Social Networks let users interact and develop their social relationships in a computer-based environment. Indeed, several works have pointed out that social elements can be inferred from them (Golbeck, 2006b; Bischoff, 2010). For example, a tie between users can be estimated by measuring the number of messages exchanged or the number of friends in common.

After this initial study we are able to verify our hypothesis H1: ***"There is a need to improve group recommender systems by better modelling decision-making processes, possibly through the inclusion of social factors."***. Throughout the rest of this chapter we will explain how we infer these social factors directly from Social Networks. Our goal is to use this social knowledge to better model group decision-making processes and therefore provide better recommendations that are closer to the real outcome when performing a common social activity. To do so, previous to the recommendation process, we analyse two main social factors (although as we will later see we sometimes take additional social factors into account): the *personality* of each individual, that is, their expected behaviour when facing conflict situations in the decision process of the product to consume, if they are open minded or not, if they participate actively in the decision-making processes, etc. And the *trust* between group members, who trusts who, who influences who, etc.

3.2 Identification and study of people's group behaviour, in relation to conflict solving, according to their personality

It is a fact that when we face a situation in which peoples concerns appear to be incompatible, *conflict situations* arise. Here conflict is understood as a difference that prevents agreement. More concretely, in group interactions it is defined as a competitive or opposing action of incompatibles: antagonistic state or action (as of divergent ideas, interests, or persons) (Mer, 2002). Different people have different expectations and behavior in conflict situations (Masthoff and Gatt, 2006) and this should be taken into account. When we started our research to improve group recommendation processes, we decided to study the different behaviors that people have in conflict situations according to their personality.

In this section we present a group recommendation method that differen-

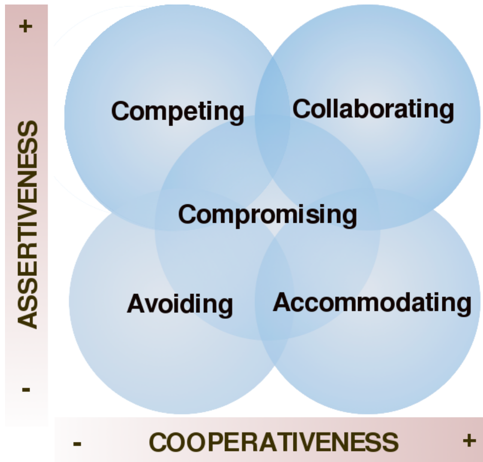


Figure 3.1: TKI personality types

tiates users in the group according to their personality. This novel technique characterizes people using the Thomas-Kilmann Conflict Mode Instrument (TKI) (Thomas and Kilmann, 1974). TKI is a test designed to measure people's behaviour in conflict situations. We have chosen this concrete test because it is a leading instrument in conflict resolution assessment that is often used by Human Resources and Organizational Development consultants to facilitate learning about how conflict-handling styles affect personal and group dynamics[1]. It provides a tangible and measurable value, easy to interpret as opposed to other similar tests like for example *Ego Gram* Berne (1964), *Pen Model* Barrett et al. (1998) or NEO-PI-R Paul T. Costa and McCrae (1995) (see (Quijano-Sánchez et al., 2014b) for a discussion of the selected personality test). In our work, we use the TKI test to build users personality profiles (p_u). This is done through 30 questions with two possible answers[2]. This profile describes a person's behavior in conflict situations along two basic dimensions: *assertiveness* and *cooperativeness*. These two dimensions of behavior are used to define five personality modes of dealing with conflicts: competing, collaborating, avoiding, accommodating and compromising (see Figure 3.1). The details of how the personality value (p_u) is computed for each user are described in the contributions presented in Chapters 10, (Quijano-Sánchez et al., 2010) and 18, (Quijano-Sánchez et al., 2013c).

Contributions 8, (Quijano-Sánchez et al., 2009) and 10, (Quijano-Sánchez et al., 2010) present a preliminary method of our *Social Recommendation Model* (*SRM*)[3], named *Personality Based Recommendations* (*PBR*), that

[1] http://www.kilmanndiagnostics.com/catalog/thomas-kilmann-conflict-mode-instrument
[2] A sample of the test in Spanish can be found at http://www.lara.warhalla.com/
[3] Our *SRM* (detailed in Chapter 4) comprehends all social recommendation methods that use social factors in the recommendation process.

consists of recommending items to a group by obtaining the different roles that people play when interacting in decision-making processes. This recommendation method is summarized in Equation 3.1:

$$pbr(\hat{r}_{u,i}, G_a) = \frac{1}{|G_a| - 1} \cdot \sum_{u \in G_a \wedge v \neq u} (\ \hat{r}_{u,i} + (p_u - p_v)\ ) \tag{3.1}$$

In this equation, $|G_a|$ represents the number of components in the active group G_a receiving the recommendation (this value is used to normalize the result). p_u and p_v are the personality values of users u and v in the group (in our case, they are computed using the TKI test). And $\hat{r}_{u,i}$ is the estimated rating for user u and item i, which will be boosted if after computing the difference between p_u and p_v we observe that user u's personality is stronger (more assertive) than user v's personality. Note that $\hat{r}_{u,i}$ can be computed using any desired individual recommendation algorithm, e.g. *collaborative*, *content-based*, etc.

In our proposal, the inclusion of the *personality* factor in a group recommendation process runs as follows: assertive behaviors penalize the differences with the best choice of other members (the other choices do not satisfy the user's personal concerns), while cooperative behaviors reward the differences with the best choice of other members (it is not the user's preference choice but it will be good enough for other members and for the group). Experiments have proved that the personality composition slightly influenced the performance of the group recommendation. This improvement was only reached for specific group configurations, as stated after comparing the results with different group recommendation approaches (Quijano-Sánchez et al., 2010, 2013c).

Once we have studied the individual characterization of people inside a group using personality values, we decided to study other factors regarding the structure of the group itself and how users interact with each other. Through the inclusion of the *personality* factor we have been able to measure users individual social behaviour. However, in order to better model group decision-making processes it is important, as seen in the introduction of this chapter, to study the group as an entity on its own and analyze its structure and behaviour as a whole not just as the union of its members. Therefore, we needed to explore other social factors, which we next detail.

3.3 Identification of social factors that influence in people's trust and how to elicit them from Social Networks

Social network users post on their profiles a huge amount of personal information that can be analyzed to compute *tie strength* with other users: likes and interests, personal information, pictures, games, etc. These factors are characteristic of *Human social networks* (see 2.2.1 for a definition) and they cannot be extrapolated to other kinds of networks (Wu et al., 2010), like for example Twitter, that are *Content social networks*. The use of trust and other social knowledge obtained from Social Networks in the development of recommender systems is not new (Golbeck, 2006b; Avesani et al., 2005). In this Thesis' work we have reviewed several existing works (Gilbert and Karahalios, 2009; Golbeck, 2006a) that identify the factors to be analysed. In most of these works, trust elicitation is done by directly asking users to rate their trust with other, task that users find tedious and can resent. That is why we have designed a non-intrusive method to compute trust between two users. The process consists of calculating the inter-personal trust by analysing users' profiles and interactions in the social network. In order to move from theory to practice it is important to note that these factors are not easy to quantify and are limited by the social networks' APIs extraction power.

Previous works have reported that *trust* and *tie strength* are conceptually different but that there is a correlation between them (Levin et al., 2004). Granovetter (1973) defines *tie strength* as a (probably linear) combination of four factors: the amount of time, the emotional intensity, the intimacy (mutual confiding), and the reciprocal services which characterize the tie. The reviewed literature identifies these four factors as some of the major dimensions of predictive variables. With these dimensions as a guide, Gilbert and Karahalios (2009) identified 74 Facebook variables as potential predictors of tie strength. They presented a diagram showing percentages that indicate the predictive power of their top seven tie strength dimensions[4] and also the top three predictive variables for each dimension. From the predictive variables that Gilbert and Karahalios (2009) provided, we selected the ones that were more representative of each major dimension and which could also be extracted from users' Social Network profiles (as we have said before, we are limited to the elicitation power that the social network grants us). Once we choose the different variables involved in the computation of the *tie strength* (which include common pictures, likes and interests, number of common friends or wall comments among others), we combine them using a weighted average and obtain what we refer to as our trust factor $t_{u,v} \in (0, 1]$, an estimated prediction of the *tie strength* between users u and v, where 0 signals weak ties (not trusted people) and 1 signals strong ties (highly trusted people)[5].

[4]Note that Gilbert and Karahalios (2009)'s top four dimensions are the ones that Granovetter (1973) used as definition of tie strength and therefore the ones that we have adopted, as the literature has not resolved the issue of giving a concrete tie strength definition, let alone specified how many discrete tie strength levels exist.

[5]We here point out that in this Thesis work we have considered our trust factor $t_{u,v}$ as the *tie strength* between two users seen from a general and domain independent point

Contributions 10, (Quijano-Sánchez et al., 2010), 18, (Quijano-Sánchez et al., 2013c) and 23 (Quijano-Sánchez et al., 2014b) present the study and elicitation process followed by our method in order to compute the trust factor. We are aware that the estimation of a value that measures *tie strength* between two people is directly influenced by the information retrieved through the chosen social network (that in our case study, Chapter 6, has been Facebook), that there are several ways to compute an estimation of it and that it is not as precise as a direct evaluation. However, although we cannot conclude the trust factor computation with a design prescription, after the studies presented in (Quijano-Sánchez et al., 2013c) or (Quijano-Sánchez et al., 2014b) among others, we are comfortable enough in presenting it as a useful estimation, so as to group recommendation, of trust between users.

3.4 Identification of additional social factors that influence in group decision-making processes

So far, we have presented the elicitation process of our two main social factors. As a result of this we obtain the *personality* and *trust* factors. These factors are the pillars of the methods in our *Social Recommendation Model* (*SRM*), that, as we explained in Chapter 1, is the set of algorithms that include social factors to different existing group recommendation techniques. Note that, in addition to the *personality* and *trust* social factors, throughout this PhD Thesis we have studied three additional social factors. Firstly we studied *homophily*, which is an extensively studied factor in the social sciences (Burt, 1982; Miller Mcpherson and Cook, 2001; McPherson and Smith-Lovin, 1987) that refers to the idea that users in a social system tend to bond more with ones who are similar to them than to ones who are dissimilar and whose study on information diffusion can be valuable in several contexts like understanding *social roles* of users (Choudhury et al., 2010a,b). We study this social factor in the contribution presented in Chapter 12, (Quijano-Sánchez et al., 2011d). The second additional social factor is *persuasiveness*, which is defined as a deliberate attempt on the part of one party to influence the attitudes or behavior of another party so as to achieve some predetermined end (Kaptein and van Halteren, 2013) and that when applied to *argumentation* (or group recommendation) processes is studied as a series of persuasive messages between parties to reach a consensus (Todorov et al., 2002). We study this social factor in the contribution presented in Chapter 19, (Recio-García et al., 2013). And the third factor is *justice* that when understood as a social factor can be defined as the equal distribution of welfare, opportunities, and privileges within a society (Oxf, 2010) and that

of view. A different matter would be to try to predict $t_{u,v}$ as the trust in someone's recommendation in a particular domain. We leave the analysis of this possibility as possible future work.

when applied to group recommendation can be referred as an homogeneous distribution of *satisfaction* within a group (Liang et al., 2007; Masthoff and Gatt, 2006). We study this social factor in the contribution presented in Chapter 13, (Quijano-Sánchez et al., 2011c).

3.5 Conclusions

The main goal of this Thesis is the improvement of the techniques that were typically used when recommending different products to groups. As mentioned above, most of the previous works in group recommendation consider the preferences of every member of the group with the same relevance and try to satisfy the preferences of every individual (McCarthy and Anagnost, 1998; McCarthy, 2002; Lieberman et al., 1999). However, groups of people have very different characteristics, like size, different relationships between their members or different distribution of people with similar or antagonistic personal preferences. Our approach presumes that the general satisfaction of a group regarding a recommendation is not maximized through the simple aggregation of its members' preferences. The novelty of our approach lies in the social factors' elicitation process and its later inclusion to group recommendation processes.

In this chapter we have presented 5 social factors (*personality*, *trust*, *homophily*, *persuasiveness* and *justice*), which will be the ones used in our *SRM*. Besides, it is through the social factors presented in this chapter that we will measure the *Impact of social factors and organizations in group recommendation processes* and we will contrast this Thesis formulated hypothesis: ***"The inclusion of social factors improves the performance of group recommendation techniques"***. In next chapter we detail the methods in our *SRM* and a series of performed experiments, with both real and synthetic users, where after testing several group recommendation methods (like aggregation of preferences without social factors, inclusion of the *personality* factor, inclusion of the *trust* factor or inclusion of other social factors) we will show a remarkable improvement in the recommendations results when social factors are included in the process (the contributions presented in Chapters 18, (Quijano-Sánchez et al., 2013c), 22, (Quijano-Sánchez et al., 2014a) and 23, (Quijano-Sánchez et al., 2014b)).

Chapter 4

Development of the social group recommendation methods that form the *SRM*

4.1 Introduction

In the previous chapter, we explained the potential benefits of including social factors in group recommendation processes. Contributions presented in Chapters 18, (Quijano-Sánchez et al., 2013c), 22, (Quijano-Sánchez et al., 2014a) and 23, (Quijano-Sánchez et al., 2014b) show that the inclusion of social factors improves the results of traditional group recommenders. The goal of this chapter is to validate our second hypothesis: ***"H2: It is possible to develop group recommender systems that model groups' social behaviour by including social factors"***. To do so, we will explain the different methods that we have designed in order to apply social factors to traditional group recommendation techniques. This approach, that we have named *Social Recommendation Model* (SRM), uses, along traditional individual and group recommender techniques, variables that measure the *personality* of each group member and the *tie strenght* between each other, that is the p_u and $t_{u,v}$ factors explained in the previous chapter. Besides, as introduced at the end of the previous chapter, we have studied the inclusion of additional social factors (that complement our two main factors) to the SRM, these are *homophily*, *persuasiveness* and *justice*.

In this chapter we explain different ways of designing and improving group recommendation processes through the inclusion of these five social factors. To do so, we will start from the basic pillar of group recommenders: the obtention of users' individual preferences. These individual preferences can be differently obtained: by asking users to rate different items (Costello et al., 2006) or, more commonly, by using an individual recommender (like the ones described in Chapter 2.1.1) that provides users' real rating estimations to the system. In our method we use both *collaborative* and *content-based* recommenders depending on the system's needs. These recommenders have as outcome a rating $\hat{r}_{u,i}$, that estimates what each user u in an active group G_a would give to each item i in the catalogue I of target items to recommend. Once the individual preferences of each group member are predicted, the next step is to choose an aggregation function. The selection of a proper aggregation strategy is a key element in the success of the generated recommendation for the group. For this reason (as we will see in Section 4.9), we have reviewed several existing aggregation techniques and performed a study of which ones are better for each group recommendation strategy (whether social or not) and each group configuration (big groups, small groups, etc) (Quijano-Sánchez et al., 2011a, 2014a).

There are several techniques for individual preferences aggregation (Masthoff, 2004), being *least misery* (Masthoff, 2004) (where the minimum is taken), *most pleasure* (Masthoff, 2004) (where the maximum is taken) and *average satisfaction* (Abramowitz and Stegun, 1964) (where the average is taken) the most common ones. Our *SRM* is based on simple preference aggregation approaches. These approaches (Masthoff and Gatt, 2006; O'Connor et al., 2001) provide a prediction for an active group G_a using an aggregation function ($\sqcup$ in next equation) that aggregates the individual ratings ($\hat{r}_{u,i}$) that have been estimated for every user u in the group and item i:

$$\hat{r}_{G_a,i} = \bigsqcup_{\forall u \in G_a} \hat{r}_{u,i} \tag{4.1}$$

This equation provides an estimated rating $\hat{r}_{G_a,i}$ for a given item i and an active group G_a. After this computation, the group recommenders modelled by our *SRM*, as well as most group recommenders, propose the k items in I with the highest estimated group scoring.

This process of simple aggregation (Equation 4.1) is common to most group recommenders in the literature. However, it is in this point where we include social factors and therefore digress from traditional group recommendation techniques in order to better reflect the different aspects that each different group presents. This is our way of considering decision-making processes, i.e. recommendations, not a linear process but a variable one that depends on several factors. To reflect these group characteristics, our *SRM* modifies the individual predicted ratings provided by the individual recommender $\hat{r}_{u,i}$ with our social factors. Hence, our *SRM* can be defined as the set of methods that follow Equation 4.2 instead of Equation 4.1.

$$\hat{r}_{G_a,i} \quad = \bigsqcup_{\forall u,v \in G_a \wedge v \neq u} SocialFunction(\, \hat{r}_{u,i} \, , \, p_u \, , \, t_{u,v} \, , \, sf \,) \tag{4.2}$$

Where: $\hat{r}_{G_a,i}$ is the estimated rating for a given item i and active group G_a, p_u fits in a range of $(0, 1]$, where 0 represents a cooperative personality and 1 a assertive personality and $t_{u,v}$ fits in a range of $(0, 1]$, where 0 represents a low trust level and 1 a high trust level. The last element, sf, is a set of different social factors $\{fs_1, .., fs_n\}$ that can be included or not, depending or whether we want to further apply more social factors or not to the function. In our case, $n = 3$ and it refers to the secondary social factors studied till the moment: *homophily*, *persuasiveness* and *justice*.

We have devised several *Social Functions* (see Equation 4.2) that combine the predicted ratings ($\hat{r}_{u,i}$) with the *personality* (p_u), *trust* ($t_{u,v}$) and others social factors. In each remaining section of this chapter (Sections 4.2-4.8) we will explain a different way of combining these social factors following different motivations and techniques, i.e., the different methods the conform our SRM.

4.2 Proposal of recommendations based on delegation, DBR (Delegation-Based Recommendations)

In this section we explain our Delegation-Based Recommendation method (*DBR*). This method (that we just summarize in here) is extensively explained in the contributions presented in Chapters 18, (Quijano-Sánchez et al., 2013c) and 22, (Quijano-Sánchez et al., 2014a) among others. The *DBR* method is inspired by an approach previously described in (Golbeck, 2006a), where individual predictions are based on other users' estimated ratings. The idea behind this approach is that users create their opinions based on the opinions of their social environment. The average of these opinions is weighted depending on the level of trust with every friend. Additionally, in our proposal, the personality of every friend is also taken into account, thus modifying the base opinion. So basically, in each user's turn in $u \in G_a, |G_a| = n$, the user's opinion is not taken into account but in the other (n-1) turns that is when the user influences others. Instead of using the information contained in a user's opinion just once, this method takes it into account every time another user of the same group states an opinion.

DBR for a user u in an active group G_a and an item i in the catalogue

I of items to be recommended is computed as follows[1]:

$$\begin{aligned} dbr(\hat{r}_{u,i}, G_a) &= \frac{1}{T} \sum_{v \neq u \in G_a} t_{u,v}[\, \hat{r}_{v,i} + \theta_{v,i} \cdot (p_v - p_u)\,] \\ \text{where} & \\ T &= \sum_{v \neq u \in G_a} t_{u,v} \end{aligned} \tag{4.3}$$

As we can see in this equation we take into account the predicted preference $\hat{r}_{v,i}$ of every friend v for item i. This predicted rating is increased or decreased depending on the differences of personality between both friends (u and v), $p_v - p_u$. This way if user v has a strong (or selfish) personality s/he will have a higher impact on the prediction for user u. However, it is important to note that a user v with a strong personality and a high preference for item i, $\hat{r}_{v,i}$, would try to increase the opinion of user u about that item. In the opposite case, a low preference for the item, user v would try to decrease u's opinion. This behaviour is modeled using the $\theta_{v,i}$ parameter as follows, lets say that $\hat{r}_{v,i}$ is in a range of [a,b]:

$$\theta_{v,i} = \begin{cases} 5 & \text{if } \hat{r}_{v,i} \geq \frac{b-a}{2} \\ -5 & \text{if } \hat{r}_{v,i} < \frac{b-a}{2} \end{cases} \tag{4.4}$$

We have chosen constants 5 and -5 because after several studies in group personality composition (Quijano-Sánchez et al., 2011b, 2014a) we have observed that the mean difference in group personality composition is 0.2 and therefore the impact of $\theta_{r_{v,i}} \cdot (p_v - p_u)$ in Equation 4.3 will typically be $\pm 5 * 0.2 \approx \pm 1$, which in comparison with other tested ranges ($\theta_{r_{v,i}} = 1, .., 10$) has proven to be the most adequate.

Finally, the prediction of user v that has been modified according to the personalities is also weighted by the trust between both users $t_{u,v}$. Note that this equation is not normalized by the group size and uses the accumulated trust[2] (represented as T). We have chosen this option following the findings of Golbeck (2006b) where a method for group recommendations using trust is proposed.

Although *DBR*'s main idea of using the preferences of the rest of the group members instead of the preferences of the user being evaluated in each

[1] The final equation that we here display is the one that appears in the contribution presented in Chapter 22, Quijano-Sánchez et al. (2014a). However, we must note that the computing of the *dbr* has evolved and improved throughout this Thesis work (see Table 4.1), mainly, thanks to the feedback given by different journal and conference reviewers. Hence, in previous works (Quijano-Sánchez et al., 2010, 2013c) we can see a simpler equation to the one here presented.

[2] Trust values always are greater than 0 so we do not have problems with this normalization.

Version and paper it appears	**Equation** *DBR*
V1. (Quijano-Sánchez et al., 2013c)	$dbr(\hat{r}_{u,i}, G_a) = \frac{1}{T}\sum_{v\neq u\in G_a} t_{u,v}[\,\hat{r}_{v,i} + p_v\,]$ where $T = \sum_{v\neq u\in G_a} t_{u,v}$
V2. (Quijano-Sánchez et al., 2010)	$dbr(\hat{r}_{u,i}, G_a) = \frac{1}{T}\sum_{v\neq u\in G_a} t_{u,v}[\,\hat{r}_{v,i} + \alpha\cdot(p_v - p_u)\,]$ where $T = \sum_{v\neq u\in G_a} t_{u,v}$ *and* $\alpha = 5$
V3. Quijano-Sánchez et al. (2014a)	$dbr(\hat{r}_{u,i}, G_a) = \frac{1}{T}\sum_{v\neq u\in G_a} t_{u,v}[\,\hat{r}_{v,i} + \theta_{v,i}\cdot(p_v - p_u)\,]$ where $T = \sum_{v\neq u\in G_a} t_{u,v}$ *and* $\theta_{v,i} = \pm 5$

Table 4.1: Evolution of the delegation-based rating computation (*dbr*) throughout this Thesis work. Note that the order of the different versions does not match with the publication dates. This is due to the common delays in the paper's publication process.

moment could seem counterintuitive, we have performed several experiments with alternative methods, e.g. the *IBR* method that we will next explain, and they have all provided worse recommendations than *DBR*. We can see these conclusions reflected in Figure 4.2 which we will later explain along the other two recommendation methods that appear in it (*IBR* and *Coalitions*). Once we established that the *DBR* method was, among our other social methods, the one that obtained better results, in Quijano-Sánchez et al. (2014a) we performed an experiment to prove the benefits of including social factors. In it, we built 4 group recommenders: a recommender that uses simple aggregation techniques, a.k.a. with no social factors, and three others that represent the gradual inclusion of our two main social factors (*personality* and *trust*). These experiments allowed us to prove, as shown in Figure 4.1, that the recommender that took into account both of our main social factors, i.e. our *DBR* method, was the one that obtained better results. In these experiments we compared the choices of each observed group (15 in total) with the results provided by the 4 group recommenders. To do so, we used an evaluation function called *success@3* (explained in Chapter 2, Section 2.1.2.2), that evaluates if there is at least one hit[3] in the first *3* positions. The 4 group recommenders that we compared were:

- *Non social.* A standard group recommender using the *Average Satisfaction* aggregating function following Equation 4.1.

- *Personality.* That only uses the personality values and implements the *Personality Based Recommendation* method described in Chapter 3, Section 3.2 (Equation 3.1).

- *Trust.* That implements the *Delegation-Based Recommendation* method

[3]We say that there is one hit if the recommender guesses correctly at least one item that belongs to the list of the top 3 items chosen by the group.

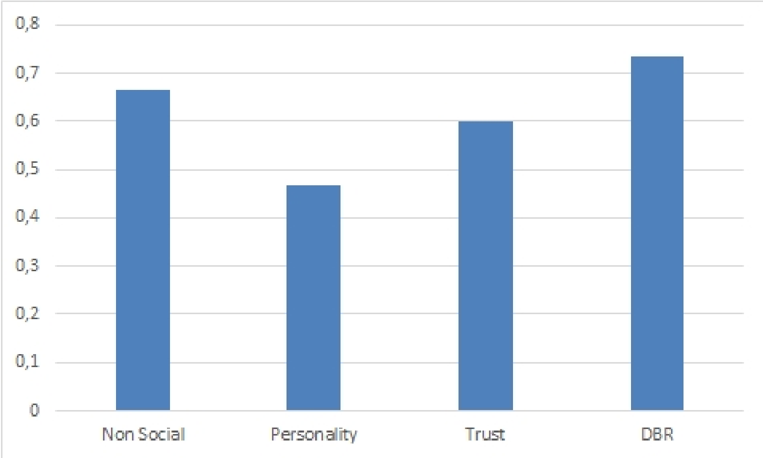


Figure 4.1: Hits' percentage for the 4 implemented group recommenders using the average satisfaction aggregation function as aggregation function.

(*DBR*) presented in this section (Equation 4.3), but that only takes into account the trust values $t_{u,v}$ (p_u values are nullified).

- *DBR*. That is the full Trust and Personality *DBR* algorithm (Equation 4.3).

Summing up, the *DBR* method, whose rating computation is presented in Equation 4.3, tries to simulate the following behavior: when deciding which item to choose within a group of users we ask the people who we trust. Moreover, we also take into account their personality to give a certain importance to their opinions (for example, because we know that an assertive person may get angry if we do not choose her/his preferred item). This method is up till now the one that has obtained better recommendation results (when comparing it to the *IBR* and *coalitions* methods that we will next explain). However, we believe that further experiments could be performed in order to determine which method out of the whole set in our *SRM* provides better results and, more importantly, what group conditions and/or configurations could make one method better than other. This intuition leads us to the idea of adaptive recommenders, presented in the future work section of Chapter 7.

4.3 Proposal of recommendations based on influence, IBR (Influence-Based Recommendations)

In this section we explain the *IBR* method, which is other of the proposed social recommendation strategies that form the *SRM*. This method (that we just summarize here) is detailed in Chapter 18, (Quijano-Sánchez et al., 2013c). The *IBR* method, equally to *DBR*, is built as a strategy to include social factors in traditional preference simple aggregation techniques. However, contrary to *DBR* that focuses on the concept of delegation, this new method focuses on the concept of influence, hence the name Influence-Based Recommendation. The intuition behind this method is to simulate the influence that each member of the group has in a given person. It supposes that a user may modify her/his preferences for an item depending on other members' preferences for the same item. For example, if our preference for an item is estimated with a rating of 3 and we have a friend whose estimated rating for the same item is 5, we could think of modifying our rating to 4. Depending on the trust in this friend, we decide the variation level of our rating (i.e. 3.5 if the trust is low, and 4.5 if trust is high). Furthermore, the variation of our rating also depends on our personality. If we have a strong personality (high personality value) we will not be willing to change our rating, but if we have a mild personality (low value) we could be easily influenced by other users. These concepts are reflected in the following equation:

$$ibr(\hat{r}_{u,i}, G_a) \quad = \quad \hat{r}_{u,i} + (1 - p_u) \frac{\sum\limits_{v \in G_a \wedge v \neq u} t_{u,v} \cdot (\; \hat{r}_{v,i} - \hat{r}_{u,i} \;)}{|G_a| - 1} \tag{4.5}$$

Where user's u in active group G_a estimated rating for item i following our *IBR* method, $ibr(\hat{r}_{u,i}, G_a)$, is computed by modifying the predicted individual rating $\hat{r}_{u,i}$ according to its difference with the other group members' predicted ratings $(\hat{r}_{v,i} - \hat{r}_{u,i})$. This difference is weighted with the trust value between group members $(t_{u,v})$ and user's u personality value (p_u).

Results of the evaluation of the *IBR* method can be found in (Quijano-Sánchez et al., 2013c) and (Quijano-Sánchez et al., 2011d). Besides, they are here summarized in Figure 4.2.

4.4 Proposal of recommendations based on coalitions

In this section we describe our method based in *Coalitions* which is another method in our model. This method follows a new approach to solve conflict situations (defined in Section 3.2) by modeling users interaction in group recommender systems. Here, instead of computing a global recommendation for an active group of users G_a based on individual preferences, users' personality and interpersonal trust, we propose a model where each user negotiates to try to convince other members about a common item to consume. We exploit the principle of *homophily*, term used to reflect the tendency that individuals have to associate and link with people who they share interests with. This feature has been shown to exist in many social networks (Miller Mcpherson and Cook, 2001; Lazarsfeld and Merton, 1954). In our model, users with strong personalities try to create *alliances* with other users to support their personal preferences. This way, these type of users, that we will refer to as *influencers*, work to obtain the required votes to get their proposal chosen by the group. These *influencers*, or leaders, try to influence other users and use their leadership to create an alliance inside the group.

Influencers, are typically characterized as thought "leaders", or just plain interesting personalities who have the ability to influence potential users. In practice, these individuals, also called *connectors* (Gladwell, 2000), may be identified as highly connected individuals or individuals that bridge two relatively large sub-communities. This social behaviour has been extensively researched in the social sciences over the past few decades (Burt, 1982; Miller Mcpherson and Cook, 2001; McPherson and Smith-Lovin, 1987).

Our approach based in *coalitions*, detailed in the contribution presented in Chapter 12, (Quijano-Sánchez et al., 2011d), uses personality and trust (as defined in Chapter 3) as the mean to define *alliances* inside a group of people. An alliance is defined as a subgroup that agrees about the same recommendation result. A leader creates alliances with other users s/he trusts in order to support a concrete item i. Leaders in an active group G_a ($l \in L_a \subset G_a$) are identified as users whose personality value exceeds a given "strong personality" threshold. "Possible allies" ($PA(l)$) are those who have a trust value with the leader over a given "high trust" threshold. Once leaders and their "possible allies" are identified the *alliance* formation process begins. To do so, we search which items i, in the catalogue I of items to be recommended, would each leader choose. These items, which we refer to as "leader favourites" $Fav(l)$, are obtained by selecting the k best items $\hat{r}_{u,i}$ that our individual recommender finds. Once each leader has her/his favourite list $Fav(l)$ s/he proposes to each user $v \in PA(l)$ each item $i \in Fav(l)$. If the estimated rating for this item i, $\hat{r}_{v,i}$, is greater than a threshold δ, we include user v to the leader's allies, $LA(l)$. δ is computed through a function that uses user's preferences, her/his personality (the threshold is greater the higher the user's personality value is, as s/he will be more difficult to persuade) and the trust with the leader (the threshold is smaller the higher the user trusts the leader). The recommender proposes the list $Fav(l)$ that belongs to the leader that manages a bigger allies list $LA(l)$.

Regarding the results of this method, in the contribution presented in Chapter 12, (Quijano-Sánchez et al., 2011d) we performed experiments using the evaluation function *success@3* (described in Chapter 2, Section 2.1.2.2 and also used in Section 4.2) where we compared the three social recom-

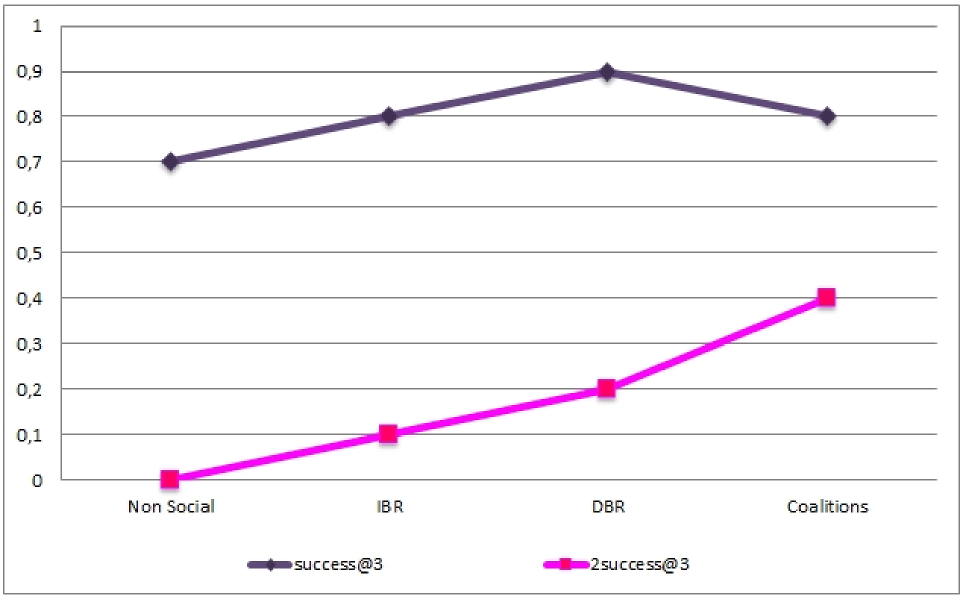


Figure 4.2: Hits' percentage of the 4 implemented group recommenders.

menders (that belong to our *SRM*) described up till now: *DBR* (Equation 4.3), *IBR* (Equation 4.5) and recommender that implements our *coalitions* based method with a *Non social* recommender (Equation 4.1). Results, summarized in Figure 4.2, show that equally to the two other social recommendation approaches the *coalitions* based method improves (in terms of global percentage of hits) the results of the *Non social* recommender. When we compare, using the *success@3* function, the *coalitions* based method with the *DBR* method we can see, as concluded in Section 4.2, that *DBR* method performs better. However, if we use a more restrictive evaluation function, *2success@3*[4] we see that *coalitions* based method significatively improves the other recommendation techniques.

Up till now we have presented three different approaches: *DBR*, *IBR* and *Coalitions* in charge of generating (in a static way) group recommendations by using some of the the social factors proposed in this Thesis. However, in our research we want to take one step forward in the study of the *Impact of social factors and organizations in group recommendation processes* and include an additional social factor, *persuasiveness*. To do so, next we propose a method that uses the social network topology in which a group receiving a recommendation is organized to perform social recommendations that follow a dynamic argumentation process.

[4]That instead of checking if the recommender guesses correctly at least one item that belongs to the list of the top 3 items chosen by the group, checks if there are at least two items of that list.

4.5 Proposal of recommendations based on distributed models and argumentation

In previous sections we have described the main social factors studied in our *SRM*: *personality* (Section 3.2) and *trust* among group members (Section 3.3). We now continue our research with the study of *persuasiveness* in a distributed architecture that imitates social network connections in group decision-making processes. Our main goal in the contributions presented in Chapters 9, (Recio-García et al., 2010) and 19, (Recio-García et al., 2013) is to improve social group recommenders by taking into account the network topology and the social factors: *personality*, *trust* and *persuasiveness*. To include these three features in our model we define a multi-agent architecture following a social network topology, where each agent will be a persuasive element representing a member of the group. Each agent representing a user will look for her/his best interests by *arguing* with the other agents it is connected to, again, following our social recommender proposal, the final recommendations will be influenced by the personality of each user and the mutual trust.

Summing up, our goal is to include the possibility of generating dynamic argumentation processes instead of static recommendation processes. To do so, we propose a new method for our *SRM* that consists of using a distributed architecture of agents with deliberation capabilities that argue and defend the preferences of the represented user to reach a joint solution. In the network of agents each agent has knowledge regarding the persuasive capabilities (Kaptein and van Halteren, 2013; Todorov et al., 2002) of the agents it is connected to and the personality of the user it is representing. Our model is based on the idea of taking into account the social connections of the collaborative agents along with the level of trust with the agent they collaborate with in each moment (McDonald, 2003; Golbeck, 2006b; Ziegler and Golbeck, 2007). This contribution's main goal is to improve the results in group recommender systems by moving to a distributed model with social network topologies, introducing social factors, like *personality*, *trust* and *persuasiveness*, plus an argumentation process that enables users to argue and defend their opinions by delegating this task to the agents that represent them. In the experiments carried out in contributions presented in Chapters 9, (Recio-García et al., 2010) and 19, (Recio-García et al., 2013) we have been able to conclude that distributed models and argumentation techniques including personality and social trust improve the satisfaction of users involved in a group decision-making process.

So far we have revised different approaches to include our social factors in group recommendation methods (both in static way: *DBR*, *IBR* and *Coalitions*, and in a dynamic way: *DistributedModels*). However, as we will next see, in this Thesis we have studied the importance of including one last social factor, *justice*, that allows us to homogenize users' satisfaction with past recommendations.

4.6 Proposal of recommendations based on memory of past recommendations

Until now we have focused on the specific situation where the recommender makes a recommendation just once. But frequently, a group will expect to use the system several times, thereby getting a bigger sample of recommendations. Our novel recommendation techniques proposed up till now (the different methods in our SRM) always tend to favor the same users, because they have stronger personalities or because they are closer to each other. Therefore, we could end up with a situation where we have some dissatisfied users because we take their opinions less into account for the group's sake. In order to avoid situations of uneven satisfaction levels inside a group, we propose a new method for our model where we take into account an additional social factor: *justice*. A motivation for this choice is an hypothetic a situation where a certain recommendation is very promising for the group in general, but one of the users ends up very dissatisfied with the recommendation. In that case, it would be desirable that future recommendations favor this component of the group so that s/he could reach a proper level of satisfaction.

To address this issue, we propose the use of a memory of past recommendations. This way, if one member accepts a proposal that s/he is not interested in, next time her/his preferences will be prioritized in the recommendation process. This means that her/his opinion will have a higher weight next time. These weights will also be influenced by the different personalities of each group member. For example, a user who dislikes the recommended item (gives it a low rating) may nevertheless be satisfied with the recommendation, especially if s/he appreciates that it has been necessary to balance conflicting interests. Her/his satisfaction might be all the greater if s/he has a more accommodating (less selfish) personality type, or if the recommendation better matches the tastes of group members with whom s/he has stronger connections, being her/his opinions influenced through *contagion*[5] and *conformity*[6] reasons (Masthoff and Gatt, 2006). This behaviour is modeled by immediately compensating users who have been negatively affected and have strong personalities and bearing in mind that users with mild personalities might not mind giving in several times.

[5] The influence of an individual's affective state on that of others in the group (Barsade, 2002; Hatfield et al., 1994).

[6] Action that causes individuals to change their opinions due to group pressure (wanting others to like them) or the better trust in other people's judgment (more than in their own) (Deutsch and Gerard, 1955).

We design a satisfaction value s_u, that represents the level of satisfaction of a user u. In it, a user who is extremely happy with the recommendations will have this satisfaction value close to 1. However, the more upset with the recommendations s/he is, the more that this value will decrease, reaching down to 0 in the worst case. The next step is to include this factor that represents *justice* in a group in our *SRM*. For example, if we use the *DBR* approach (Section 4.2) the equation to follow would be:

$$\begin{aligned} dbr_m(\hat{r}_{u,i}, G_a) &= \frac{1}{T} \sum_{v \neq u \in G_a} t_{u,v} [\, \hat{r}_{v,i} + \theta_{r_{v,i}} \cdot (p_v - p_u) \,] + m_v \\ \text{where} & \\ T &= \sum_{v \neq u \in G} t_{u,v} \\ m_v &= \alpha(1 - s_v) p_v \end{aligned} \tag{4.6}$$

Where, m_v represents the memory of past recommendations. Parameter $\alpha \,\epsilon\, (0, 1)$ is used to modify the impact of memory in the dbr_m. It has a positive or negative value according to $\hat{r}_{v,i}$, equally to the $\theta_{r_{v,i}}$ value used in the *dbr* computation (Section 4.2). Note that initially all users are assigned a $s_v = 1$. Therefore, the first time that a group receives a recommendation the memory factor is nullified in the equation as it is not necessary because there are not previous recommendations. Finally, it is important to note that s_u is also weighted depending on the personality of the user v (p_v) to reflect the importance of satisfying that particular user.

After the obtention of the product/s to be recommended (Equation 4.2), the final step in recommendation processes that use memory of past recommendations is to update the s_u factor for each group member ($u \,\epsilon\, G_a$) according to their individual satisfaction with the current group recommendation ($\hat{r}_{G_a,i}$). This is done through the following equation:

$$s_u(t) = (1 - \delta) \cdot is_u(t) + \delta \cdot s_u(t - 1) \tag{4.7}$$

Where $s_u(t)$ is user's u accumulated satisfaction with group G_a. $is_u(t)$ is the instant satisfaction, computed after the last received recommendation. And, $\delta \in [0..1]$ is used to adjust the impact of the previous satisfaction value $s_u(t-1)$.

Note that this process has been designed and fully evaluated in the contribution presented in Chapter 23, (Quijano-Sánchez et al., 2014b) and also studied in the contributions presented in Chapters 13, (Quijano-Sánchez et al., 2011c) and 18, (Quijano-Sánchez et al., 2013c).

4.7 Proposal of recommendations that solve the cold-start problem

A well-known problem that collaborative recommenders suffer is the one known as *cold-start* problem (Herlocker, 2000; Schafer et al., 2007a). This problem (as introduced in Chapter 2, Section 2.1.1.2) occurs when individual recommenders have difficulties in making good predictions for new users that have very few ratings. This lack of ratings makes very difficult the task of finding the most similar user, which, as we remember (Section 2.1.1.1), is the methodology that collaborative recommenders follow. Group recommenders inherit this problem because they rely on the use of individual recommenders. Some solutions that have been proposed for the *cold-start* problem in individual recommenders are: including population averages, ask for more ratings or hybrid recommenders that use also content-based recommenders (Schafer et al., 2007a).

The contribution presented in Chapter 16, (Quijano-Sánchez et al., 2012b, 2013a) is a solution based in a Case-Based Reasoning (CBR) approach (Leake, 1996) along with information related to the group receiving the recommendation. Our goal is to improve group recommendation results when *cold-start* situations occur. We use a case base CB in which each case records information related to a group and its previous recommendations. When an active group G_a requests a new recommendation and one or more of the group members are in *cold-start*, we find a case $c \in CB$ that represents similar group G_r. Case c describes a previous recommendation event where the users were not in *cold-start* and played similar roles in their group to the roles that the users play in the active group. Basically what we do is: give to the user $u \in G_a$ that is in *cold-start*, all the ratings that s/he does not have but her/his most similar user ($v \in G_r$) does. To do so, for each group that has a user in *cold-start* we find the most similar group in our case base CB. We identify the most similar group as the one that on average has the highest similar users. Next, for each user in *cold-start* in the active group G_a we find the most similar user in the retrieved group G_r. We identify how similar one user is to another as the average of how similar they are studying their common ratings, their age, gender, personalities and trust between group members.

The results of the experiments that we have carried out (and that are fully detailed in the contributions presented in Chapter 16, (Quijano-Sánchez et al., 2012b, 2013a)) show that our CBR *cold-start* method works better than any other *cold-start* strategy that does not take group information into account or not strategy at all. Therefore, we are able to conclude that our method improves the quality of group recommenders.

4.8 Proposal of social recommendations based on CBR

In previous sections (4.2-4.7) we have presented different methods (*DBR*, *IBR*, *Coalitions*, *Memory*, *DistributedModels* and *Cold-start*), that belong to our *SRM*, whose main goal is to improve the results obtained by group recommender systems through the inclusion of social factors. The dynamic of the proposed methods is to combine different social factors using the equations that we have designed for this purpose (Equations 4.3,4.5 or 4.6 for example). As we have seen in the previous sections and in the corresponding cited papers (Quijano-Sánchez et al., 2013c, 2011d, 2014a), this *social approaches* indeed improve the results of other group recommendation methods that do not use social factors. However, there might be situations where the use of generic equations are not equally effective for all the different types of configurations that groups might present (big or small groups, groups whose overall personality is strong, groups with strong connections between members, etc). Following this idea, in the contribution presented in Chapter 17, (Quijano-Sánchez et al., 2012a) we have extended our group recommendation research with a method that uses a case base of previous group recommendation events, which will allow us to replay behaviour-patterns between similar groups.

As we explained before (Section 4.1), group recommender systems commonly aggregate predicted ratings for group members (Jameson and Smyth, 2007). That is, for each group member, an individual recommender system predicts a set of ratings for the candidate items; then, the group recommender aggregates the ratings. The new group recommender method that we here present takes the same approach, i.e. it aggregates the preferences of the group members, but it uses Case-Based Reasoning (CBR) for the aggregation. This way we avoid using the same social recommendation method for each group characterisation.

To do so, our system stores a case base of past group recommendation events, CB. Each case c records the members of the group it represents; the candidate items to consume; the previous item that the group chose to consume together, which we will call the selected item; and the ratings that each group member gave to the selected item after consuming it. To make a recommendation to a new active group G_a our CBR system deploys a unique combination of two collaborative recommender systems: *user-based* and *item-based* (Schafer et al., 2007a) (described in Chapter 2 Section 2.1.1.1).

Firstly, we use a *user-based* collaborative recommender (Zhao and Shang, 2010) to predict a rating for each candidate item by each group member. Next, we retrieve cases, i.e. past group recommendation events, that involve groups that are similar to the active group G_a. For the case retrieval we use a *user-user* (Resnick et al., 1994) similarity measure, and, as a *by-product* (Sarwar et al., 2001), it aligns each member u of the active group G_a with a member v of the group G_r in the case c. The similarity measure compares group members on their age, gender, personality and ratings and the degrees of trust between members of each group.

Once the system retrieves the k most similar cases $c_1, .., c_k \in CB$ to the active group G_a, it reuses the contribution that each group member made in choosing the selected item and transfers them to the corresponding member of the active group G_a. That is, it predicts the ratings that each user u in the active group G_a would give to the products to be recommended using the ratings that user's u most similar user, v gave in her/his corresponding case c. This is done by scoring the new candidate items using *item-item* similarity techniques (Wang et al., 2006). In this way, the retrieved cases act as implicit models of group decision-making, which are transferred to the decision-making in the active group. Finally, it recommends the candidate items that have obtained the highest scores.

The advantage of a case-based approach to preference aggregation approaches (DBR, IBR, $Coalicions$, etc) is that it does not require us to commit to a model of social behaviour, expressed in a set of equations, that may or may not be valid across all groups. Rather, the CBR system's aggregation of the predicted ratings will be a lazy and local generalization of the behaviours captured by the neighbouring cases in the case base.

4.9 Evaluation of our SRM for the different existing aggregation strategies

In this chapter's introduction we explained how our *Social Recommendation Model* (*SRM*) uses an individual recommender to obtain group members' rating estimation ($\hat{r}_{u,i}$), modifies these estimated ratings with social factors ($SocialFunction(\ \hat{r}_{u,i}\ ,\ p_u\ ,\ t_{u,v}\ ,\ sf\ )$) and finally uses a selected aggregation function in order to combine these ratings (Equation 4.2). As there are several existing aggregation functions (Masthoff, 2004) we have performed experiments with several of them and discovered that the *Average Satisfaction* function is one of the best ones for our SRM.

In the contributions presented in Chapters 14, (Quijano-Sánchez et al., 2011a) and 22, (Quijano-Sánchez et al., 2014a) we have performed several experiments both with real and synthetic users using different studied aggregation functions. Our goal is to observe which aggregation function works better with different implemented group recommenders. To do so, we have performed recommendations that used our SRM, recommendations that only used the personality factor, recommendations that only used the trust factor and recommendations that used no social factor at all. Results (see Figure 4.3) show that no method beats our SRM (in this case implemented

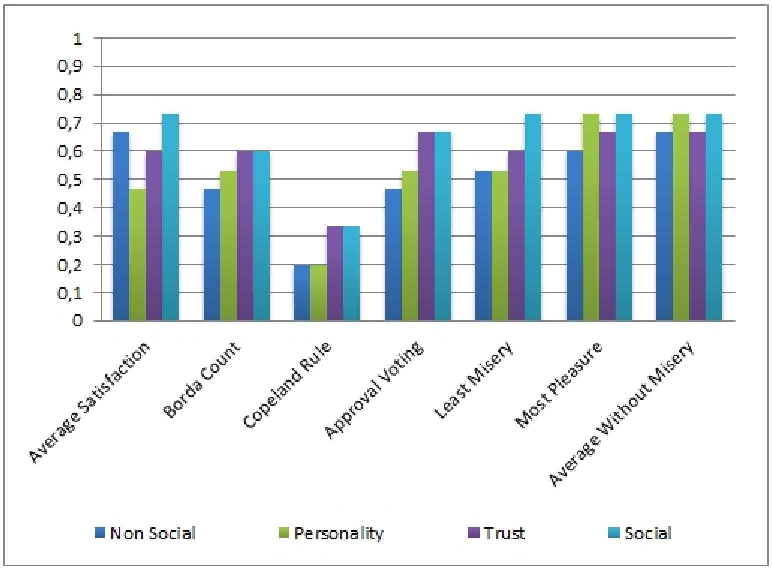


Figure 4.3: Comparison results of our *SRM* (social), a method that does not use social factors (non social), a method that only uses the personality factor (personality) and a method that only uses the trust factor (trust) when using different aggregation functions.

with the *DBR* method) that, as we remember uses mainly two social factors (*personality* and *trust*) and that one of the best performing aggregation functions for this method is *Average Satisfaction*). In Quijano-Sánchez et al. (2011a, 2014a) we can find the complete results, conclusions and experimental details. There, we have also studied which aggregation function better adapts to each group configuration based on size. To that end, we have performed several recommendations to groups of different sizes (see Figure 4.4) this time only for our *SRM*. Results show that while some aggregations functions like *Average Satisfaction* perform better for smaller groups, others like *Least Misery* or *Most Pleasure* behave differently and perform better for bigger groups. These and other results can be found in (Quijano-Sánchez et al., 2011a, 2014a).

4.10 Evaluation of the proposed methods

In order to evaluate the methods proposed throughout this chapter and observe if by introducing social factors in group recommendation approaches, we improve the results of existing state-of-the-art group recommenders, we have tested our methods in the movies domain. We have chosen this domain because it has several available datasets and the task of obtaining information related to movies is quite simple. Besides, the fact that all users have

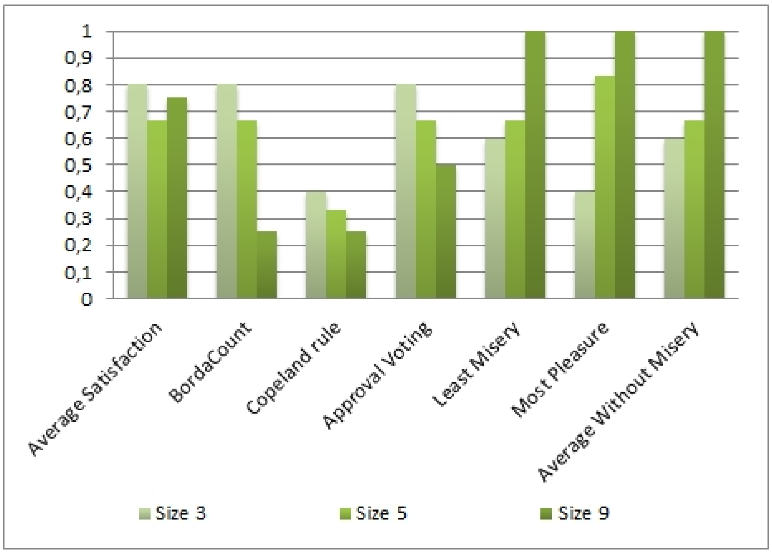


Figure 4.4: Comparison results of our *SRM* for groups of size 3, 5 and 9.

a general knowledge about movies, has also influenced the decision of using this domain as our validation domain. This general knowledge allows users to know the domain without having to be experts, making the usage of the system and the evaluation of the results much easier.

Next, we briefly explain the patterns of the experiments that we have performed, that as detailed in the contributions presented in Chapters 10, (Quijano-Sánchez et al., 2010), 14, (Quijano-Sánchez et al., 2011a) and 22, (Quijano-Sánchez et al., 2014a) among other papers, have consisted in the extraction of different sources of knowledge and the simulation of our *Social Group Recommendation Methods* (Sections 4.2 to 4.8).

Initially, we have performed experiments with synthetic data. The reason for this initial choice (using generated data instead of data from real users as we have done later) is that it allows us to explore all possible permutations of the studied factors in group configurations, even the ones that are less likely to occur in real life scenarios. To do so we need to have control of the data distribution, fact that does not occur when using real data.

After studying and analysing the results obtained in the synthetic data experiments, we have continued our experiments establishing two goals: (1) the verification of the previously obtained results, proving that the simulated synthetic data is realistic by obtaining similar results and/or the same conclusions when repeating the experiments with real data (this fact has been proven in several papers, e.g. (Quijano-Sánchez et al., 2010) and Quijano-Sánchez et al. (2014a)); (2) a deeper study about how to improve group recommendations through the extraction of social factors from Social Networks. To do so, experiments with real user data were needed[7]. In order to obtain the required real user data we have created events in the social network Facebook[8] and asked different groups of users to join them[9]. Through these events we are able to automatically extract the social data required for the computation of the trust factor ($t_{u,v}$) and to obtain the complimentary needed user data by asking users to answer to 2 questionnaires[10].

The first questionnaire is used to obtain the personality factor (p_u). To do so, as seen in Section 3.2, we use the TKI personality test (Thomas and Kilmann, 1974). The second questionnaire obtains users individual preferences in the movies domain ($r_{u,i}$). In it, users rated (between 0 and 5) more than 40 heterogeneous movies selected from MovieLens database (Bobadilla et al., 2009). Around 58 users have participated in our experiments.

Next, we need an evaluation function that measures the group recommender's accuracy by comparing its results with the choices that users would have made if the proposed scenario was a real life situation. To evaluate the results, we have asked the groups of users formed in the Facebook events to meet together and debate which 3 movies out of a provided movie listing of 15 they would like to watch as a group, simulating that they are going together to the movies. We have managed to obtain 15 groups of 9, 5 and 3 members (4, 6 and 5 groups respectively). The 3 movies that each group chose are stored as the *real group favourites* set, *rgf*. This way, in order to evaluate our recommenders we can compare the set of the best 3 movies that the group recommender offers –*gf*– with the group's real preferences –*rgf*–.

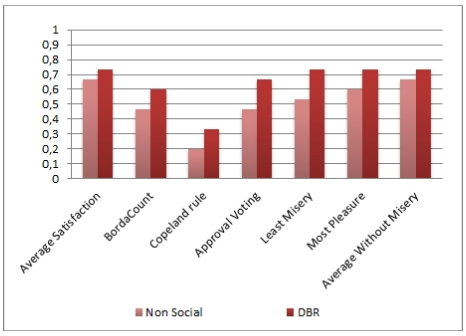
Finally, our experiments have been divided in the following parts: (1) we have automatically generated groups of users that follow different personality distributions and social network topologies in the case of synthetic data experiments and/or we have used the extracted data from the created Facebook events for the real data experiments; (2) we have built a collaborative individual recommender (Chapter 2, Section 2.1.1.1); (3) we have built different group recommender systems that use this individual recommender but that implement different approaches: firstly we have built a standard

[7]In order to collaborate with the recommenders research community we have anonymized the data that we have obtained throughout our experiments and attached it for downloading in `http://sourceforge.net/projects/jcolibri-cbr/files/misc/`.

[8]http://www.facebook.com

[9]For the algorithmic evaluation of the methods in our *SRM* we used two questionnaires outside Social Networks to obtain users' preferences and personality, we performed a face-to-face evaluation and only used Social Networks as means to compute the trust factor. However, as we will see in Chapter 6, throughout the research process followed during this PhD Thesis, we improved the evaluation process and developed an application prototype in the social network Facebook, *HappyMovie*. This application serves us as means to retrieve all the required information, making the retrieval process more dynamic, as means to present the obtained results (recommendations), providing visibility to our work, and, as means to obtain feedback, making the evaluation process easier.

[10]These questionnaires are available (in Spanish) in `http://www.lara.warhalla.com/`.



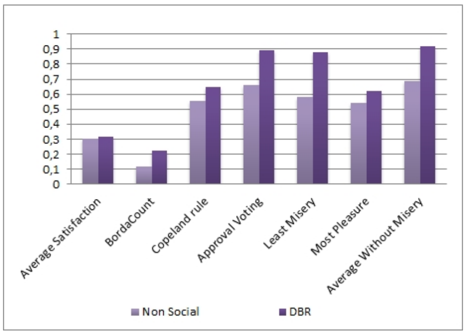


Figure 4.5: Comparison results of one of our *Social Approaches* (DBR) with a *Standard Approach* ($NonSocial$) for the different studied aggregation functions. Real users (left) and Synthetic users (right).

recommender, also denoted by *base* or *Non Social* (explained in Section 4.9, Equation 4.1), this recommender just limits itself to aggregate user preferences. Next, we have built two more recommenders, one that only uses the social factor related to the personality of each user (as the one explained in Section 3.2, Equation 3.1), and other that only uses the social factor related to trust between users (explained in Sections 3.3 and 4.2). Last but not least, we have implemented all the methods that have been proposed in this chapter, i.e. our *social recommenders* that integrate our SRM and that take into account our social factors of *personality*, *trust*, *homophily*, *persuasiveness* and *justice*: DBR (Section 4.2, Equation 4.3), IBR (Section 4.3, Equation 4.5), $Coalition$ based (Section 4.4), the one based in $DistributedModels$ (Section 4.5), $Memory$ based ones (Section 4.6, Equation 4.6), the ones based in the *cold-start* problems (Section 4.7) and the ones based in CBR (Section 4.8); (4) we have compared the recommendation results (*gf*), obtained with the different recommenders, with the data that reflects the real group decision-making (*rgf*).

4.11 Conclusions

In this chapter we have presented our SRM, whose general approach consists of modelling social group recommender systems by capturing groups social behavior inside group recommendation techniques. To capture the different social aspects inside group dynamics the model uses different social factors. Up till now, we have analysed and included the 5 social factors presented in the previous chapter: *personality*, *trust*, *homophily*, *persuasiveness* and *justice*. However, our model, represented by Equation 4.2 allows the inclusion of more social factors, fact that we leave for future work. This equation, Equation 4.2, defines a $SocialFunction$ that combines the studied social factors with group recommendation techniques. In this chapter we have presented

7 methods that implement this *SocialFunction*: *DBR* (Section 4.2), *IBR* (Section 4.3), *Coalitions* (Section 4.4), *DistributedModels* (Section 4.5), *Memory* (Section 4.6), *Cold-start* (Section 4.7) and *CBR* (Section 4.8). Besides, we have performed several experiments to validate these methods and our *SRM*. The results of this evaluation process have allowed us to evaluate the *Impact of social factors and organizations in group recommendation processes* and to confirm both, this Thesis main hypothesis: ***"The real satisfaction of a group regarding a group recommendation cannot be accurately estimated using the simple aggregation of its members' individual preferences. Considering people as social entities that relate with each other allows the better estimation of their individual satisfaction regarding the result of the recommendation and, therefore, improves the global group satisfaction"***, and this chapter's hypothesis: ***"H2: It is possible to develop group recommender systems that model groups' social behaviour by including social factors"***. This fact, the improvement of the results when using social factors, is reflected in Figures 4.1, 4.2 and 4.3 (previously presented in this chapter), in papers (Quijano-Sánchez et al., 2010, 2011a,d,c, 2012b,a; Recio-García et al., 2013; Quijano-Sánchez et al., 2013c, 2014a) (here summarized), and in Figure 4.5[11].

[11] This Figure belongs to the experiments carried out in 22, (Quijano-Sánchez et al., 2014a), where we compared for both real users (left) and synthetic users (right) the behaviour of our social method *DBR* (explained in Section 4.2) in comparison with a standard recommender that does not use social factors (*Non Social* in the Figure) for all the different studied aggregation functions. The used evaluation function is *success@3* (explained in Section 4.2 and Chapter 2, Section 2.1.2.2).

Chapter 5

Generic architecture and development methodology for the instantiation of the model

5.1 Introduction

In previous chapters we have explained how Social Networks store personal information about their users and how through them, users interact with others and the system (they comment users' walls, they friend each other, they rate items, etc). These new sources of knowledge can be used to improve recommendation techniques and develop new strategies which focus on social recommendation. In Chapters 3 and 4 we have presented our ideas of how to improve group recommendations through the inclusion of social factors like the personality of each group member, the trust that they have between each other or justice in the long run. Besides, we have proved that by including social information in group recommendation processes we improved the accuracy of recommendations (as seen in Section 4.10).

In this chapter we pose a new hypothesis: ***"H3: It is possible to generalize our** SRM **in a way that it is applicable to different domains and in a way that other recommender systems developers can reuse it"***. To validate it we establish two new goals:

(1) Abstract our *Social Recommendation Model* (*SRM*), to do so we will generalize our social recommendation methods pointing out the steps that must be followed when performing social recommendations in general, including both individual and group recommendations (note that in this goal we do not focus only on group recommendations, as we have previously done, but in generalizing our method for any type of social recommendation). The result of this goal is a compilation of our techniques and algorithms in an organized generic architecture named ARISE.

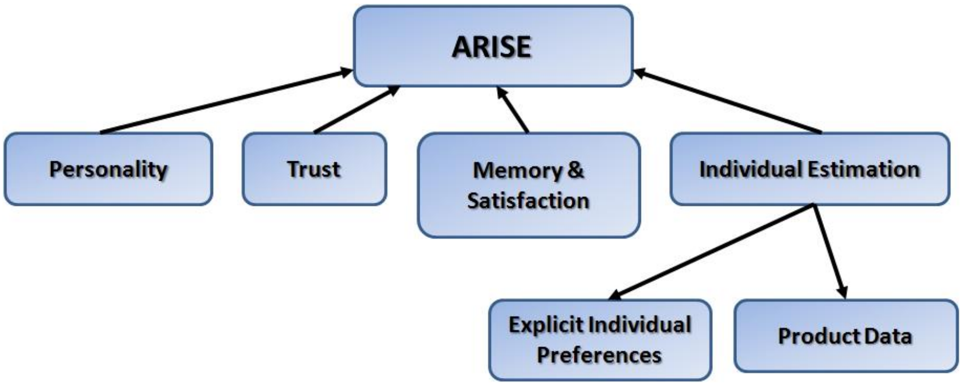


Figure 5.1: Global view of ARISE.

(2) Create, share and validate a set of templates that help developers create social recommender systems in a semi-automatic way.

5.2 Generic architecture: ARISE

The abstraction of our *SRM* in a reusable generic architecture that serves as an instantiation model for concrete social recommenders in different domains is what we call ARISE[1]. The common and key factor in all the different types of recommenders that can be built in different domains using this generic architecture is the inclusion of social factors. These social factors, that in our *SRM* are mainly the personality and trust factors (see Chapter 3), define each person as a potentially influenced component of a social community or group determined by the environment, in most cases Social Networks, s/he belongs to. In our *SRM*, we have simulated people's social behaviour based on the idea that the relationship between individuals and their networks of people directly influence their decisions (Christakis and Fowler, 2011).

In the contributions presented in Chapters 21, (Quijano-Sánchez et al., 2013b, 2014c) and 22, (Quijano-Sánchez et al., 2014a) we introduce ARISE's architecture (Figure 5.1). We can see that it is divided in seven different modules: personality, trust, memory & satisfaction, individual estimation, explicit individual preferences, product data, and the ARISE module itself. Next we briefly summarize each module's functionality (which is fully detailed in the papers (Quijano-Sánchez et al., 2013b, 2014c,a)):

Personality Module: This module fulfils the task of obtaining a value that represents the personality of each user. This personality value, p_u, fits within a range of (0,1], 0 being the reflection of a very cooperative person and 1 the reflection of a very selfish one.

Trust Module: This module fulfils the task of obtaining the trust val-

[1] *Architecture for Recommenders Including Social Elements*

ues, $t_{u,v}$, between every user u and v that belong to a common social environment or group. Note that $t_{u,v} \in (0,1]$, 0 being the reflection of a person not to be trusted and 1 the reflection of a highly trusted one.

Memory & Satisfaction Module: This module stores all the recommendations that have been made for every user and every group. This avoids repeating past recommendations and also ensures a certain degree of *fairness/justice* in the long run. We believe that this is a necessary step when providing a whole set of fair recommendations. This way, if one user accepts a proposal that s/he was not interested in, next time s/he will have some kind of priority in the recommendation process.

Note that the personality, trust and memory and satisfaction modules are the ones in charge of computing the social factors of *personality*, *trust* and *justice* seen in the previous chapters and used by our *SRM*. And that the other two social factors: *homophily* and *persuasiveness*, studied in this Thesis and included in our *SRM* (Sections 4.4 and 4.5), are computed through the previous ones (Recio-García et al., 2010; Quijano-Sánchez et al., 2011d; Recio-García et al., 2013).

Individual Estimation Module: This module is in charge of computing individual predictions, $\hat{r}_{u,i}$, for each user u and each item i in the catalogue I of items to be recommended. The individual predictions, or recommendations, consist on a basic building block of the architecture as our recommendation approach predicts the rating that each user would assign to every item in the catalogue and later, if used for group recommender applications, these estimated ratings are aggregated to obtain a global prediction for the group.

Explicit Individual Preferences Module: This module obtains information about users' preferences ($r_{u,i}$). This is required in order to be able to predict users' ratings for new items. Commonly, it just consists of ratings given to some products in the catalogue.

Product Data Module: This module obtains the catalogue of products to be recommended, I.

ARISE **Module**: This module is only needed when using the architecture for social group recommender systems. It combines all the information provided by the rest of the modules and offers a group recommendation. To do so, it uses the social recommendation strategies presented in the previous chapter.

5.3 Development methodology to ease the instantiation of social recommender systems: *Social Recommenders Design Templates*

After designing the generic architecture ARISE, our next goal was to design a tool that could ease the work of other recommender developers. To do so, we have proposed a tool based on the template-based design of jCOLIBRI (Recio-García et al., 2014). In the contribution presented in Chapter 21, (Quijano-Sánchez et al., 2013b, 2014c), we have created a set of social templates that represent an intermediate step between ARISE and any social other application that can be built following its structure. We propose a case-based reasoning approach (see Chapter 2, 2.1.1.2 and 2.1), where the cases are previously designed systems. When a designer wants to create a new social recommender application it retrieves a similar system previously designed (our templates), it reuses it, and revises if everything they need is in the templates, adding new methods if necessary.

Our templates are formed by *tasks*, that represent the different steps that developers must take when designing a new social recommender system. These *tasks* help developers ease their designing work. Also, for each *task* we provide different *methods* that solve the *task* with a particular implementation. These *methods* help developers ease and quicken their implementation work[2]. We can see our *generic social templates* in Figure 5.2, we refer to them as generic because through them we can obtain different implementations that represent a final instanciation of the model that we propose. These templates are composed by *generic tasks* and *simple tasks*. *Generic tasks* encapsulate sequences of *simple tasks*. Depending on the breakdown of each *generic task* into sequences of *simple tasks*, we obtain several *final templates* that represent a concrete instantiation of our model. In Figure 5.2 we can see that there are two different templates, one called *pre-cycle template* and the other one called *cycle template*, this division is done following jCOLIBRI's architecture (Díaz-Agudo et al., 2007), where the *pre-cycle* loads the resources and the *cycle* executes the CBR cycle.

If we observe Figure 5.2 and Figure 5.1 we can see that some of the different *templates's tasks* correspond to some of ARISE's modules. In the templates, each *task* can be implemented by several different *methods*. Most of them are already implemented and therefore are reusable, hence facilitating developers the process of building a new recommender system by using our *social templates*.

Next, we briefly present each of the *tasks* in which our templates are divided and some of the possible *methods* that can be used to implement

[2]Note that these concepts regarding *tasks* and *methods* are based on the *Knowledge level modelling* methods (Aamodt and Plaza, 1994).

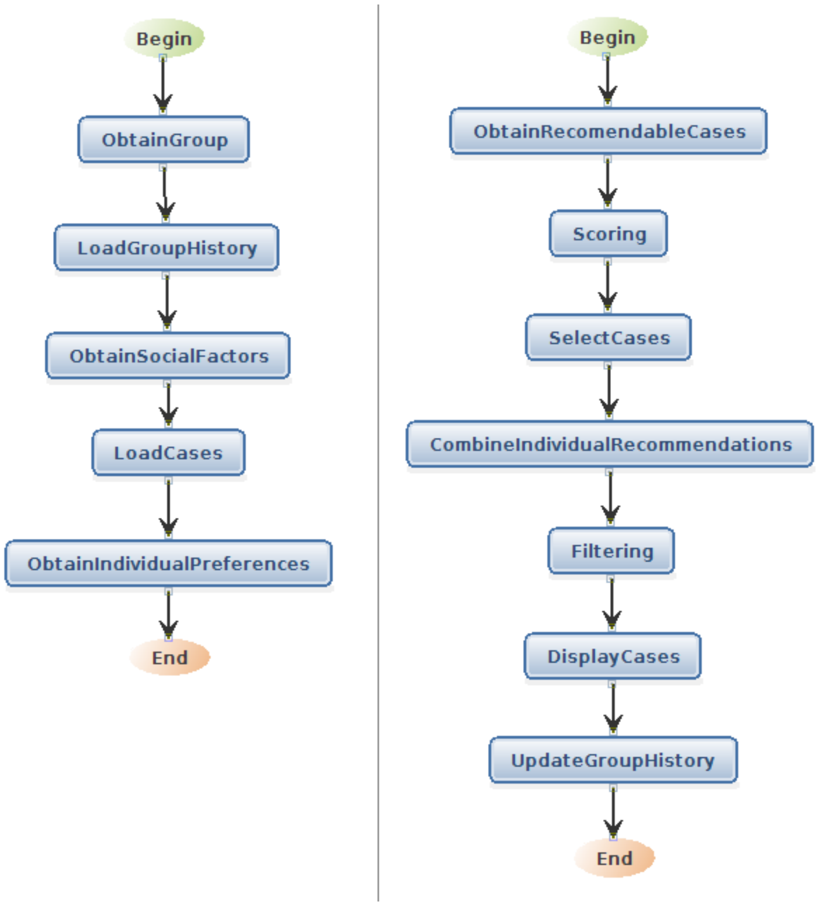


Figure 5.2: *Social templates*: *Pre-cycle* template (left) and *Cycle* template(right).

them. Note that these are fully detailed in (Quijano-Sánchez et al., 2013b, 2014c). Firstly, we will describe Figure 5.2 left, that shows the *pre-cycle template*, which is formed by the following *tasks*:

ObtainGoup Consists of obtaining the *id* of each user $u \in G_a$, being $G_a = \{u : 1 \ldots n\}$ the active group of users and $|G_a| > 1$. G_a (previously defined in Chapter 4, Section 4.1 for social group recommender systems) is the set of people that intend to realize an activity together. As ARISE and our *Social Recommenders Design Templates* are meant to be tools not only for the design and development of group recommenders systems but also for individual recommenders systems, we extend the definition of G_a for social individual recommender systems. This definition is understood as the people who belong to the circle of trusted people in the social environment of the user receiving the recommendation. For both options, the group is defined in the framework of Social Networks. Some of the different *methods* that can be used to obtain G_a are:

- Through the creation of an event to perform an activity together (for social group recommender systems).
- Calculating the group of closest friends in the Social Network (for social individual recommender systems). To do so, the method obtains a trust value (as we will explain bellow in the *ObtainTrustFactors task*) with all the user's friends in Facebook that also use the system being implemented.

LoadGroupHistory Corresponds to *Memory & Satisfaction* module in ARISE. Assume a case base CB in which each case $c \in CB$ records a previous recommendation event. This *task* consists of retrieving the case c that corresponds to the active user u or group of users G_a. Note that this *task* is optional and can be skipped if developers of the new system do not want to build a system with memory of past recommendations.

ObtainSocialFactors Consists of a *generic task* that encapsulates the following *subtasks*:

- **ObtainUsersPersonality**: Corresponds to *Personality* module in ARISE. Consists of obtaining the personality of each user u. As we remember, the idea of including users' personality, denoted p_u, was introduced in Chapter 3, Section 3.2 and applied in the different methods of the SRM. To obtain this factor, users must complete a personality test on registration with the recommender system. This *task* can be fulfilled by the following *methods*:
  - The Thomas-Killmann Conflict Mode Instrument (TKI) (Thomas and Kilmann, 1974) that proposes 30 situations where the user has to think about how s/he will react. (For example, this *method* was used in (Quijano-Sánchez et al., 2010))
  - TKI's alternative movie metaphor, that consists of displaying two well known movie characters with opposite personalities for each of five possible categories (personality aspects). One character represents the essential characteristics of one category, while the other one represents all the opposite ones. What the user has to do is to choose with whom of each pair of characters s/he feels more identified. To do so, s/he will move an arrow that indicates her/his degree of resemblance with the character. (For example, this is the *method* used in *HappyMovie* (Quijano-Sánchez et al., 2014b) and in *HappyShopping* (Quijano-Sánchez et al., 2013b) as we will see en next chapter, Sections 6.2 and 6.6).
  - Any other personality test from which the p_u can be defined as a single numeric value of range $(0, 1]$, where 0 signals a very cooperative person and 1 signals a very selfish person.

- **ObtainTrustFactors**: Corresponds to *Trust* module in ARISE. Consist of obtaining the tie strength, our trust factor $t_{u,v}$, between users u and v ($u \neq v \in G_a$). This factor, that measures closeness between users, can be estimated with the distance in the Social Network, the number of friends in common, relationship duration, and so on. In order to elicit this information directly from social networks (avoiding with this action tedious questionnaires that users might resent) we can use social networks like Facebook, Tuenti or Google+, that is *Human social networks*. For example, as we mentioned in Chapter 3, Section 3.3, when we introduced the idea of including our trust factor, in (Gilbert and Karahalios, 2009) 74 Facebook variables were identified as potential predictors of tie strength. On the other hand, in (Quijano-Sánchez et al., 2014b) we presented a method (offered in our templates as *implemented method*) to compute $t_{u,v}$ by automatically eliciting needed variables and information directly from Facebook.

LoadCases Consists of obtaining items i in the domain catalogue $D = \{i : 1 \dots m\}$.

ObtainIndividualPreferences Corresponds to *Explicit Individual Preferences* module in ARISE. It can be based on a combination of implicit data, i.e. according to the user's patterns of use (e.g. Ardissono et al. (2004); Zimmerman et al. (2004)) or, on explicit data, where the user briefly, and throughout the usage of the system, specifies their preferences to the system (e.g. Billsus and Pazzani (1999); Mccarthy et al. (2004); Quijano-Sánchez et al. (2010)). For example, a system which sells books may recommend new books for a user to buy based on which books they have looked at or bought in the past (implicit rating), or how they have actively rated books (explicit rating). The *implemented method* that we provide for this *task* consists of obtaining the ratings $r_{u,i}$ that each user u in G_a assigns to items i in D. Ratings are on a Likert scale, e.g. 1 = terrible and 5 = excellent.

We continue templates explanation with the other presented template, the *cycle template*, shown in the Figure 5.2 right. This template is principally designed for its use in social group recommender systems building processes, however it can also be used for social individual recommenders leaving the last four *tasks* unimplemented. The *cycle template* is formed by the following *tasks*:

ObtainRecommendableCases Corresponds to *Product Data* module in ARISE. Consists of obtaining all the candidate target items i in the recommendation catalogue $I = \{i : 1 \dots n\}$. For example, for *HappyMovie*, as we will see in next chapter, we have built a *Web Crawler* that parses a leisure guide web[3] and retrieves all the movies and movie sessions being displayed in Spain's cinemas. In this template, we provide this *Web Crawler* as a *method* that implements this *task* as it can easily be adapted to other leisure activity

[3]http://www.guiadelocio.com/

domains offered by this web like restaurants, theatres, concerts or museums for instance[4].

Scoring Corresponds to *Individual Estimation* module in ARISE. Consists of obtaining predicted ratings $\hat{r}_{u,i}$ for each active user $u \in G_a$ and target item $i \in I$. Some of the different *methods* that can be used to implement this *task* are the traditional individual recommender methods seen in Chapter 2, Section 2.1.1, or the social recommendation method seen in the previous chapter, Section 4.3:

- *Collaborative recommenders* (Ekstrand et al., 2011; Koren and Bell, 2011; Herlocker et al., 2002).

- *Content-based recommenders* (Lops et al., 2011).

- *Hybrid recommenders* (Burke, 2002).

- *Asking other G_a users to give an estimated rating for the product i* (Costello et al., 2006), this method relies heavily on explicit feature-level feedback from users.

- *Influence based recommenders* (Quijano-Sánchez et al., 2013c, 2010), that modify non-social predictions $\hat{r}_{u,i}$ obtained with one of the above *methods* with the *personality* and *trust* factors computed in previous *tasks*. This method, detailed in the previous chapter: IBR (Section 4.3, Equation 4.5), is used in the case study of these templates, in *HappyShopping*'s recommendations, as we will see in in next chapter, Section 6.6.

SelectCases Consists of selecting for each active user $u \in G_a$ the k items in I whose predicted ratings $\hat{r}_{u,i}$ are highest. For example, in *HappyShopping* (Section 6.6), we use $k = 4$. Note that the next three *tasks* are specific for social group recommendations, and therefore the *method* that implements this *task* will need to have a *display cases* option for single individual recommender implementations.

CombineIndividualRecommendations Corresponds to ARISE module in ARISE. Consists of obtaining a group prediction, $\hat{r}_{G_a,i}$, aggregating group members predicted ratings, $\hat{r}_{u,i}$ for each $u \in G_a$ and $i \in I$ (see Equation 4.2 in previous chapter). The methods that implement these are the ones presented in the previous chapter, DBR (Section 4.2, Equation 4.3), IBR (Section 4.3, Equation 4.5) or memory based ones (Section 4.6) among others.

[4]We are aware that this is limited to Spain's leisure offers, but believe that it could easily be adapted to other leisure webs in other countries. Therefore we have included the *Web Crawler* as a possible *method* that implements this *task*.

Filtering Consists of selecting the k' items in I that have the highest predicted ratings for the group. For example, in *HappyMovie* as we will see in next chapter, we used $k' = 3$.

DisplayCases Consists of displaying to each user u receiving the recommendation the k' items proposed by the group recommender.

UpdateGroupHistory Corresponds to *Memory & Satisfaction* module in ARISE. Consists of revising the case c that corresponds to the active user u (for individual recommenders) or to the active group of users G_a (for group recommenders) with the new recommendation and retaining it in the case base CB for future recommendations. Note that this *task* is optional and can be skipped if developers do not want to build a system with memory of past recommendations.

5.4 Conclusions

In this chapter we have validated our third hypothesis: ***"H3: It is possible to generalize our SRM in a way that it is applicable to different domains and in a way that other recommender systems developers can reuse it"***. To do so, we have designed our generic architecture ARISE. ARISE is a theoretical organization of the components required to build social recommenders according to our *SRM*. The common and key factor in all the different types of recommenders that can be built in different domains using this generic architecture is the inclusion of social factors. In next chapter we will present two use cases that are based on ARISE's architecture. These use cases have been build in different domains, movies and clothing domains, and have different goals, as one is an individual social recommender application (*HappyShopping*, Section 6.6) and the other one is a group social recommender application (*HappyMovie*, Section 6.2). The different goals and domains between the two designed use cases will allow us to conclude that ARISE is indeed a valid generic architecture to build social recommender systems in different domains.

After designing ARISE, our next goal to validate this third hypothesis has been to design a tool that could ease the work of other recommender developers. We have created a set of *Social Recommenders Design Templates* that are a software developing methodology and represent an intermediate step between ARISE and any social application that can be built following its structure. To do so, we propose a CBR approach, where when a developer wants to create a new social recommender application s/he only has to retrieve a similar previously designed system (our templates). The developer can therefore reuse this previously designed system and revise if all the system's requirements are fulfilled by the existing templates (i.e. cases). Once the desired *implemented methods* offered in the templates have been selected an initial version of the system is automatically generated. At this point, if adaptation is needed in order to cover all the new system's functionality, the templates can be used as base-system where new *methods* can be added. In next chapter, when presenting our use case *HappyShopping*, Section 6.6, we will detail an experiment where 3 developers were asked to use our *Social Recommenders Design Templates* and develop a new social recommender application from scratch. In it, developers will later reflect whether they preferred to have the templates to assist them, in which case we could conclude that our *Social Recommenders Design Templates* are indeed useful for the recommenders community, or, whether they didn't find that our *Social Recommenders Design Templates* eased and quickened their work.

Chapter 6

Use Cases in a Social Network

6.1 Introduction

As mentioned in previous chapters, social factors (such as *personality* or *trust*) are difficult to estimate. This difficulty is even more stressed if we establish as goal the design of a non-intrusive dynamic recommender system. For example, in other works (Golbeck, 2006b; Avesani et al., 2005) that also use social factors, users' trust in these concrete examples, users are asked to explicitly state their trust with other users. The abuse of this elicitation process through questionnaires can result tedious and therefore generate rejection from users. Nowadays, the collaborative Web provides a useful tool to solve this issue: Social Networks.

Social Networks allow users to interact with each other and develop social relationships through Internet. Hence, there are some works that have pointed out that social factors could be inferred from them (Mislove et al., 2010; Konstas et al., 2009). For example, we could measure the strength of a tie between two users by computing the number of interchanged messages or friends in common. Besides, Social Networks could also be used as an experimental environment to build recommender systems, due to their design, that eases interaction and information interchange between users, their significance, nowadays nearly everyone uses some type of Social Network (as seen in Chapter 2, Section 2.2.1) and also due to their working dynamics, for example the creation of events to perform group activities that Facebook offers could be useful for the integration of group recommendations in them.

In the previous chapter we presented a generic platform that allows us to reuse our *Social Recommendation Model* (*SRM*), ARISE (Architecture for Recommendations Including Social Elements) and a methodology to perform this reuse through a software developing process based on templates that conceptualize the *SRM*. In this chapter our goal is to validate ARISE, the *Social Recommenders Design Templates* and the hypothesis: ***"H4: It is possible to validate and evaluate our generic architecture*** ARISE

through different concrete applications in different domains". To do so, we have built two use cases in the social network Facebook[1]: (1)*HappyMovie*, that is a social group application that follows ARISE's architecture and implements our *SRM*. The development of this application provides us with means to test the viability of the semi-automatic social factor elicitation process proposed in Chapter 3, and also as means to test the social algorithms presented in Chapter 4 and users' reactions towards them; (2) *HappyShopping*, that serves us to prove the validity of our model in other domains and the usefulness to the recommenders community of ARISE and the *Social Recommenders Design Templates* presented in the previous chapter.

6.2 Use case in the movies domain embedded in a social network: *HappyMovie*

To illustrate and validate the capabilities of our *SRM* we have instantiated our model into a real-life recommender system: *HappyMovie*[2], which is a particular instantiation of our generic architecture ARISE for the movie group recommendation domain in the social network Facebook. *HappyMovie* serves us as a use case and experimental environment where we are able to evaluate our *SRM* with real data.

In order to ease the social factors elicitation process required by our model, we have designed *HappyMovie* as an application in a social network, from which we can profit from the knowledge stored in it. With this step we move our theories for making recommendations to groups, which have already been proven in simulated environments (Quijano-Sánchez et al., 2009, 2010, 2013c), to the instantiation of our model in a real-life scenario: the social network Facebook. There are several reasons for this choice: First, Facebook allows users to create events and invite their friends to join activities, so our system can help them in the organization of such events. Second, users' activity in Social Networks can be easily tracked, fact from which we benefit and use it to automatically elicit the knowledge required to compute users' trust without bothering our users with excessive questionnaires, which can slow down the generation of the recommendation and/or lead to users' rejection. And finally, as Facebook provides the possibility to perform tests, polls and plain games in an interactive and dynamic way it is a perfect environment to obtain information about the user's personality, the last factor required by our model. Note that although we believe that Facebook is the most suitable network to work with, other social networks like Tuenti or Google+, also classified as *Human social networks* –from which

[1]Facebook passes 1.19 billion monthly active users.
[2]https://happymovie.fdi.ucm.es

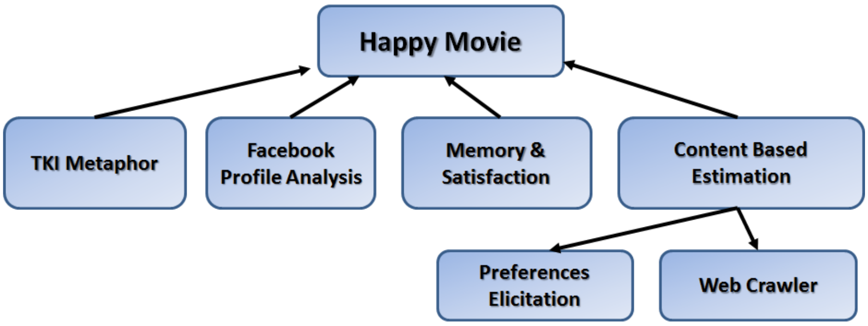


Figure 6.1: *HappyMovie*'s architecture.

we can estimate tie strength between users–, might also be adequate for the development of this type of application.

In the contributions presented in Chapters 11, (Quijano-Sánchez et al., 2011e), 15, (Quijano-Sánchez et al., 2011b) and 23, (Quijano-Sánchez et al., 2014b) we detail different aspects of *HappyMovie* that, as we mentioned above, is a particular instantiation of ARISE's architecture. Figure 6.1 shows *HappyMovie*'s module architecture. Next sections describe each of these modules as a concrete instantiation (focused on a Facebook application that provides group recommendations to groups of people that wish to go together to the movies) of its ARISE's corresponding high-level designed module. Note that ARISE was described in Section 5.2 and its architecture was reflected in Figure 5.1.

6.3 *HappyMovie*'s modules

As described in the previous chapters our *SRM* integrates social factors to improve the estimation and modelling of real decision-making processes followed by groups of people when deciding a joint activity.

Our goal with *HappyMovie* is to evolve and integrate recommender systems into the Social Web (defined in Section 2.2.1) –more concretely Facebook– where personal relations can be analyzed and infered to enhance the process of making recommendations to groups. Within this environment, we are able to elicit much of the information needed to apply our *SRM* directly from the chosen Social Network. As we previously mentioned, previously (Golbeck, 2006b; Bischoff, 2010) the acquisition of such social data had to be performed by means of several questionnaires. The integration in a social network eases this process and provides a lot of valuable feedback to evaluate and improve our proposal.

Next subsections detail each of the modules in which *HappyMovie* is divided.

6.3.1 TKI metaphor

To determine the personality, *HappyMovie* users perform an adaptation of the Thomas-Kilmann Conflict Mode Instrument (TKI) test (Thomas and Kilmann, 1974), which as we mention in Chapter 3, is a leading instrument used by individuals and businesses for identifying their ability to handle conflicts effectively. As we mentioned in Section 3.2, it consists of 30 different situations with two possible answers. Depending on the answers, a score is assigned for 5 existing personality modes (see Figure 3.1) organized according to two dimensions: *assertiveness* and *cooperativeness*. However, when we asked our users about the test (Quijano-Sánchez et al., 2011b), they described it as tedious and long. To make the application more easy going we studied the possibility of using a movie metaphor as an alternative personality test. Consequently, in *HappyMovie* we have developed this alternative metaphor that lightens this activity.

Our interactive metaphor consists of displaying two movies characters with opposite personalities for each of the five existing conflict-handling modes. One character represents the essential characteristics of the mode, while the other one represents all the opposite ones. The user has to move an arrow showing her/his degree of similarity to the characters being presented. In Quijano-Sánchez et al. (2011b) we concluded that it is possible to replace the TKI personality test with the movie metaphor test because it provides an statistically confirmed accurate estimation of the personality mode. This alternative test enhances significantly the usability and interest for the application. A screenshot of *HappyMovie*'s personality test is presented in Figure 6.2.

6.3.2 Facebook profile analysis

This module obtains the inter-personal trust or social tie between users. This factor can be estimated following different approaches, being most of them manual extractions or long questionnaires (Golbeck, 2006a), task that our users resented and found very tedious. Hence, we propose its elicitation from Social Networks. In this section we detail how the computation of the trust between two users can be automatically computed thanks to embedding the group recommender application in a social network. The process consists of calculating the inter-personal trust by analysing users' profiles and interactions in the social network. Users in Facebook can post a huge amount of personal information that can be extracted to compute the trust with other users: likes and interests, personal preferences, pictures, games, etc.

The use of trust and other social knowledge obtained from Social Networks in the development of recommender systems is not new (Golbeck, 2006b; Avesani et al., 2005). Therefore, we reviewed several existing works (Gilbert and Karahalios, 2009; Golbeck, 2006a) that identify the variables

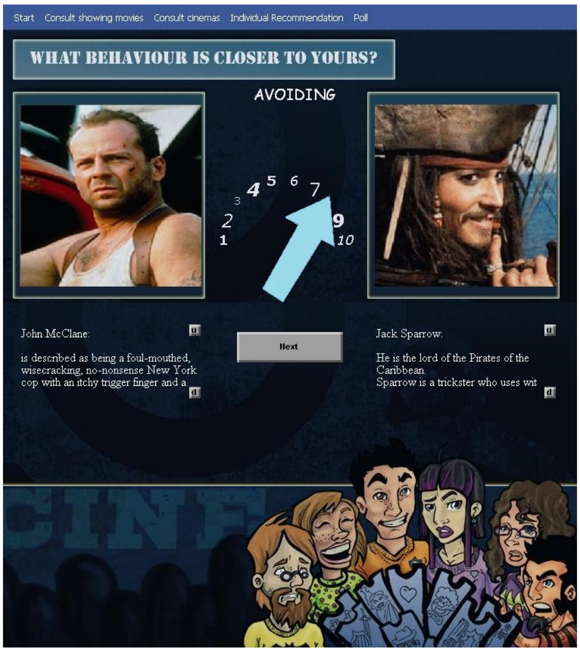


Figure 6.2: *HappyMovie*'s personality test.

to be analysed. In order to move from theory to practice it is important to take into account that these variables are not easy to quantify and are dependant of the Social Network's API. In *HappyMovie*, as detailed in Quijano-Sánchez et al. (2014b), to compute the trust between users u and v $\{t_{u,v} : u, v \epsilon U, u \neq v\}$ we use a weighted sum of the following variables: t_1) **Intimacy**, that represents how much users interact outside the Social Network. To compute it we evaluate the percentage of pictures they appear together; t_2) **Intensity**, that represents how much users interact inside the Social Network. To compute it we count the number of interactions in the Social Network; t_3) **Duration**, that represents how long they have known each other. We compute it as a structural variable that measures the number of common friends; and t_4) **Reciprocal Services**, that represents how similar users' profiles are, in terms of common interests (music, movies, etc). To compute it we evaluate the percentage of common posted information in the Social Network.

The trust calculation is done every time a user joins an event with the rest of users also attending to it. These values are not stored, but repetitively calculated as Facebook profiles keep changing and so does the trust between two people.

6.3.3 Memory & satisfaction

HappyMovie stores all the recommendations that have been made for every user and every group. This feature avoids repeating past recommendations and also ensures a certain degree of *fairness/justice* in the long run. Frequently, a group will expect to use the application several times, thereby getting a bigger sample of recommendations. However, our *SRM* tends to always favour the same users (because they have stronger personalities or because they are closer friends with other members). Therefore, we could end up with a situation where we have some dissatisfied users because we take their opinions less into account for the group's sake. In order to avoid a situation of high deviation in the satisfaction levels of the group, we must take into account users' satisfaction regarding past recommendations. It would be desirable that future recommendations favour dissatisfied users so that all of them reach a proper level of satisfaction. To address this issue, *HappyMovie* uses our proposal of recommendations based on memory detailed in Section 4.6.

6.3.4 Content-based estimation

We have chosen a *content-based* approach to estimate the rating a user would assign to a new movie (Lops et al., 2011; Pazzani and Billsus, 2007). We have chosen this option over its alternative approach, *collaborative* recommenders (Ekstrand et al., 2011), because the movies to be recommended are the most recent movies on cinemas, so it is difficult to have user ratings. Hence, we could not use those ratings as collaborative recommenders do. This module produces for every user u in the active group G_a a set $\{\hat{r}_{u,i} : u \epsilon G_a, i \epsilon I\}$ with the individual predicted ratings for all the target movies in the recommendation catalogue I.

6.3.5 Preferences elicitation

This module implements a preferences test where users indicate their taste in movies. The ratings here obtained are used by the individual recommender, that estimates the different movies to be recommended according to users' preferences in actors, genre, etc. For example, if a user has voted with 3 stars a certain movie, as we can see for example in Figure 6.3, we could consider that s/he likes that type of movies, so later, the individual recommender will analyse the characteristics of this movie and try to find a similar one. In order to complete the test, users must rate at least 40 movies through a Likert scale. Users are allowed to run this test on demand to modify or increase their ratings. The more ratings users give the more accurate their personal profile will be, and therefore the individual recommender will perform better. This test returns a set of real ratings $r_{u,i}$ for every user u in group G_a and item i in the test set in the movies domain D.

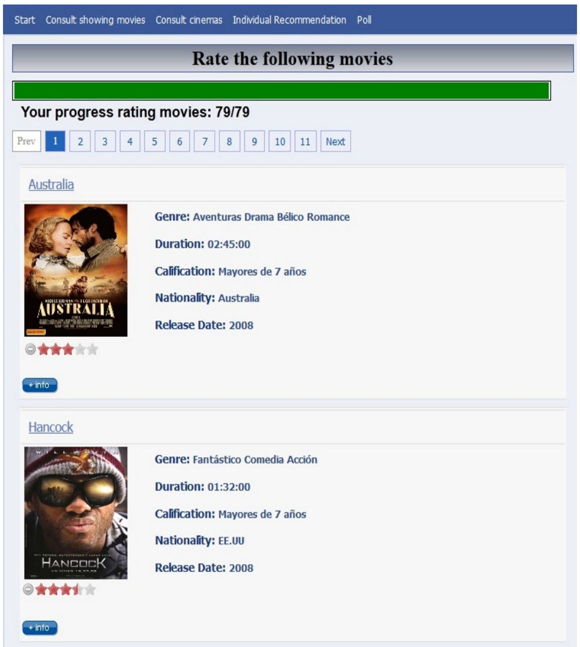


Figure 6.3: *HappyMovie*'s preferences test.

6.3.6 Web crawler

We have built a Web Crawler that parses the Spanish leisure web *La Guía del Ocio*[3] and retrieves all the movies and movie sessions being displayed in Spain's cinemas. This Web Crawler obtains a full technical datasheet for each of the movies being displayed. Each specific characteristic of the movie is a field that the individual recommender compares. For example, in our particular case study these characteristics are the movie's main actors, director and synopsis, between others. The retrieved set of movies, with all their specific information, is the target movie listing I containing the items i sent to the individual and group recommenders.

6.3.7 *HappyMovie*

This module combines all the information provided by the rest of the modules and offers a group recommendation. We have implemented all recommendation methods based on social factors presented in Chapter 4 along with the different aggregation functions also presented in that chapter. This allows us to use *HappyMovie* as a tool to test any of the proposed methods in our *SRM*.

Next we describe the general guidelines of *HappyMovie*'s functionality.

[3]http://www.guiadelocio.com/

Figure 6.4: *HappyMovie*'s main page.

6.4 Functional description of *HappyMovie*

HappyMovie's main page (Figure 6.4) offers three different activities: *perform preferences test*, *perform personality test* and *create new event*. The personality and preferences tests must be answered before being able to create events. Complementarily, users can receive invitations to join events. Through this main page users will be able to access all the events they participate. Once users answer the two mentioned tests they have full access to *HappyMovie*'s functionality (as their individual personal profile is now created and the system does not need further information). This functionality includes options such as the creation of new events, inviting friends to events or retaking the preferences test, among many other possibilities:

- **Accept invitations**: Users can accept or refuse pending invitations.
- **Create an Event**: Users can create new events indicating when and where it will take place and the deadline to join the event (Figure 6.5). New events appear in the main page.
- **Events**: This page (Figure 6.6) displays all the information of the event: attendees, celebration place, date and time, deadline to join, wall of the event, etc. However its main purpose is to present the best three recommendations for the current group. This tentative recommendation is updated when the movie listing from the selected city changes and/or when a new user enters or leaves the event.

When a new event is created, users can invite friends from a list containing all the user's Facebook contacts. Users are also able to erase themselves from

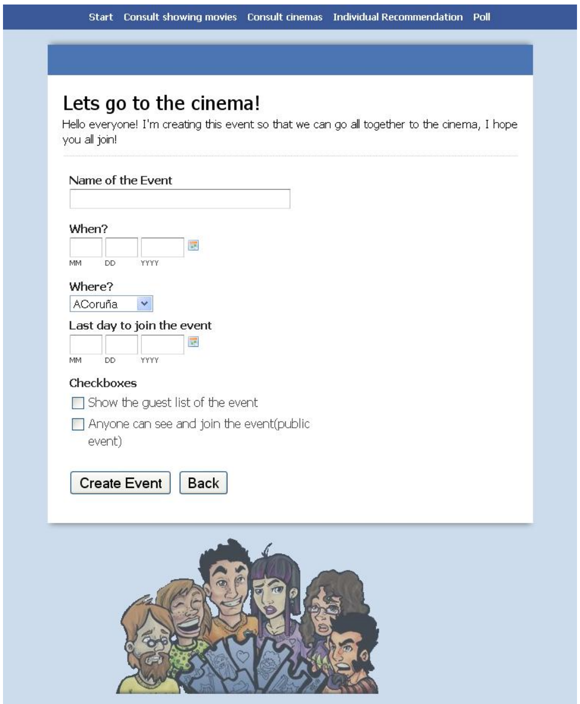


Figure 6.5: *HappyMovie*'s create event page.

the event. However, when the deadline date is reached these two options are disabled, leaving the group fixed as it was. In this moment the final three proposed movies are displayed. At this point users can vote these final three recommendations to agree on the best movie to watch together. This action also provides *HappyMovie* the information required to update users satisfaction value.

We here detail some extra *HappyMovie* functionalities, that can be accessed at any time from the tabs toolbar situated on top of each of *HappyMovie*'s pages (as for example we can see in Figure 6.4).

Individual Recommendation: Provides the best 5 movies that the individual recommender has found for a selected city and the current movie listing.

Consult showing movies: Lists all the movies being played at the selected city. Note that every time that a movie is presented in *HappyMovie* users can see its title, poster, calculated percentage of genres and some extra details. Additionally, there is always a "more info" button that takes users to an individual page where all the details of the movie (synopsis, actors, age certificate, etc) are shown (See Figure 6.7). Besides, there is always an extra button "Cinemas playing this movie" that takes users to a page where all the cinemas of the selected city and all the different sessions that they have are presented.

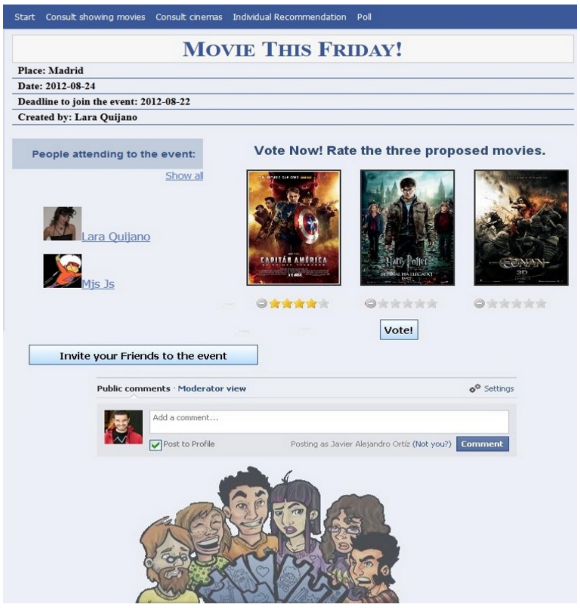


Figure 6.6: *HappyMovie*'s event page.

Consult cinemas: Lists all the existing cinemas in the selected city along with its geographical location and movie listings. This page presents the details and location through *Google Maps*.

Poll: This page contains several questionnaires about *HappyMovie*. These questionnaires allow us to obtain users' feedback regarding their satisfaction with the presented recommendations and the system in general and consequently improve our application. We will detail some of the questionnaires used for our experiments regarding *HappyMovie* in next section.

6.5 Experimental evaluation

In order to verify our *SRM* and *HappyMovie*'s usability, in the contribution presented in Chapter 23, (Quijano-Sánchez et al., 2014b) we have run several experiments where 60 users (25 females and 35 males) have tested our applications functionality. These experiments, which we will here just summarize but that are fully detailed in (Quijano-Sánchez et al., 2014b), mainly consisted of running a functional evaluation of the application and later answering to a questionnaire regarding users' perception of *HappyMovie*'s recommendations and usability. Also, these very same users answered one last questionnaire this time regarding what they valued most from a group recommender application. This last experiment was set to study what should the next generation of group recommender systems try to improve and how to obtain valuable feedback from users. The results of this study regarding

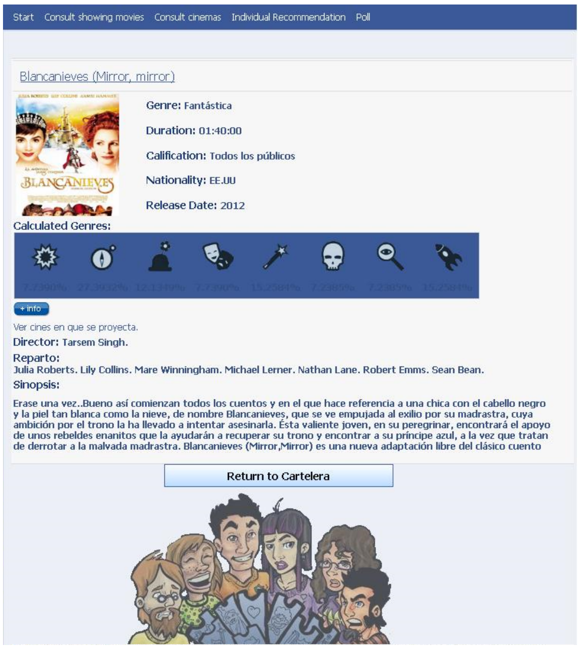


Figure 6.7: *HappyMovie*'s movies.

users' feedback can be found in the contribution presented in Chapter 20, (Quijano-Sánchez and Bridge, 2013).

Coming back to our initial questionnaire, in (Quijano-Sánchez et al., 2014b) users were asked to test *HappyMovie*'s functionality and answer some questions. More precisely they were asked to perform the following steps:

Step 1. Answer the personality test through the movie metaphor (Figure 6.2).
Step 2. Answer the preferences test (Figure 6.3).
Step 3. Check the accuracy of the recommended movies presented by the "Individual Recommendation" tab.
Step 4. Group themselves in groups of 3 people and create an event to go to the movies together.
Step 5. Check the 3 best movies that the *Social Recommender* has found for the group.

Next, users from each group had to argue as a group and decide if they liked and would follow the group recommendations. Finally, they were asked to individually answer the following questions, with a five star Likert scale[4]:

Q1. Usefulness (u): "*I find the application useful (0 being not useful at all and 5 being very useful)*".

Q2. Decision process (dP): "*It is useful because it speeds up the group*

[4] We ran the experiment with students whose first language was Spanish. The questions that we show here are paraphrases into English of the Spanish questionnaire.

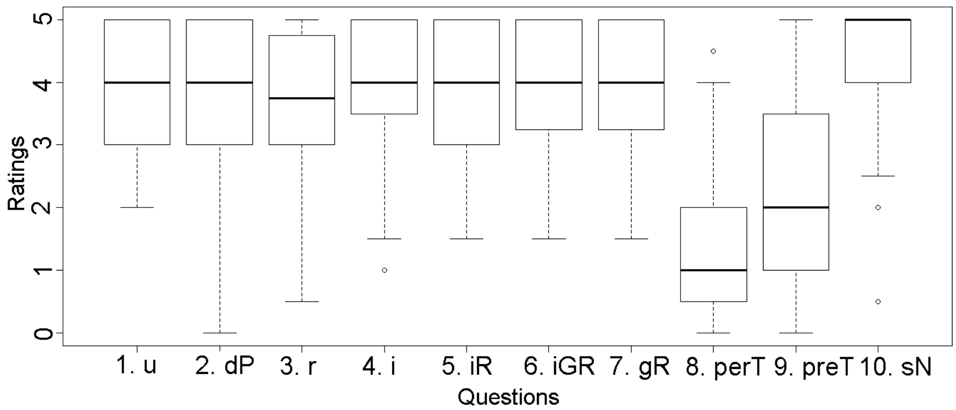


Figure 6.8: Results to our *HappyMovie* questionnaire.

decision-making process (0 being very little and 5 being a lot)".

Q3. Reusability (r): "*I will use the application to go to the movies with my friends (0 being very little and 5 being a lot)*".

Q4. Usability (i): "*The application is intuitive and easy to use (0 being not at all intuitive and 5 being very intuitive)*".

Q5. Individual Recommendation (iR): "*I like the individual recommendation of the system (0 being barely and 5 being a lot)*".

Q6. Individual Group Recommendation (iGR): "*I individually like the group recommendation of the system (0 being barely and 5 being a lot)*".

Q7. Group Recommendation (gR): "*As a group we like the group recommendation of the system (0 being barely and 5 being a lot)*".

Q8. Personality Test (perT): "*Was it easy to answer to the personality test? (0 being very easy and 5 being not easy at all)*".

Q9. Preferences Test (preT): "*Was it easy to answer to the preferences test? (0 being very easy and 5 being not easy at all)*".

Q10. Social network (sN): "*Do you like having the application in a social network? (0 being not at all positive and 5 being very positive)*".

Figure 6.8 shows the general results of the test which as we can see are very good. They reflect that users like the application (as reflected in answers 1 u, 2 dP and 4 i), they intend to use it again (answer 3 r) and more importantly they think both recommendations, individual and group, are the key point and the best feature in the application (as we can conclude from answers 5 iR, 6 iGR and 7 gR that have the highest mean and lowest standard deviation). Besides, as reflected on answers 8 perT and 9 preT, users do not resent from the questionnaires which was one of our goals in order to have a dynamic application and easy to use so that the probability of users using the application frequently could increase.

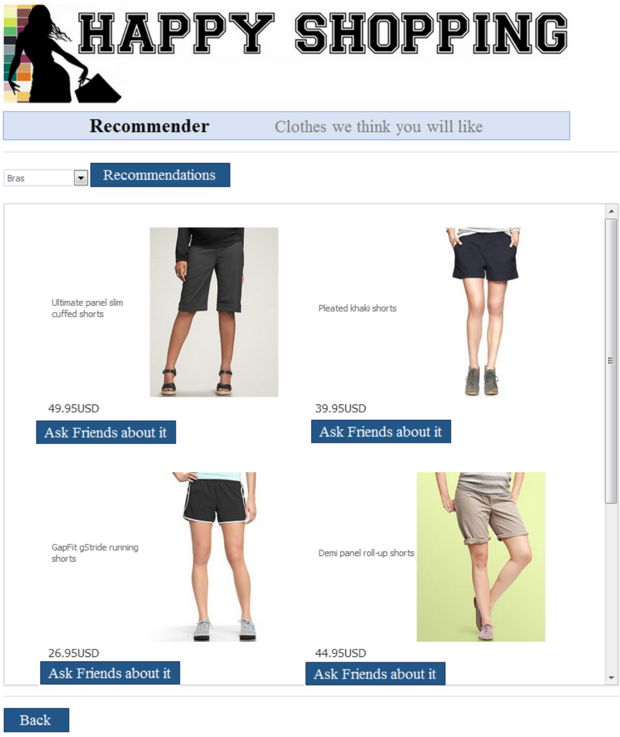


Figure 6.9: *HappyShopping*'s recommendation page.

6.6 Use case in the clothing domain *HappyShopping*

As mentioned in this chapter's introduction, our second goal regarding use cases in a social network was to test the applicability of ARISE (detailed in Section 5.2) in different domains and the usability of the proposed *Social Recommenders Design Templates* (detailed in Section 5.3). To do so, in the contribution presented in Chapter 21, (Quijano-Sánchez et al., 2013b, 2014c) we performed an experiment and asked 3 external developers to use our *Social Recommenders Design Templates* and build a new social recommender application based on ARISE. However, we asked them to build it in a new domain, clothing, as we have already tested our *SRM* in the movies domain with *HappyMovie*. The result of this experiment has been *HappyShopping*, where Figure 6.9 shows and example of the recommendations that this application presents.

HappyShopping is a Facebook application that recommends pieces of clothing to a user. We here point out the difference with *HappyMovie*, that is also a Facebook application but that implements our model in order to perform social group recommendations. This difference, having two different

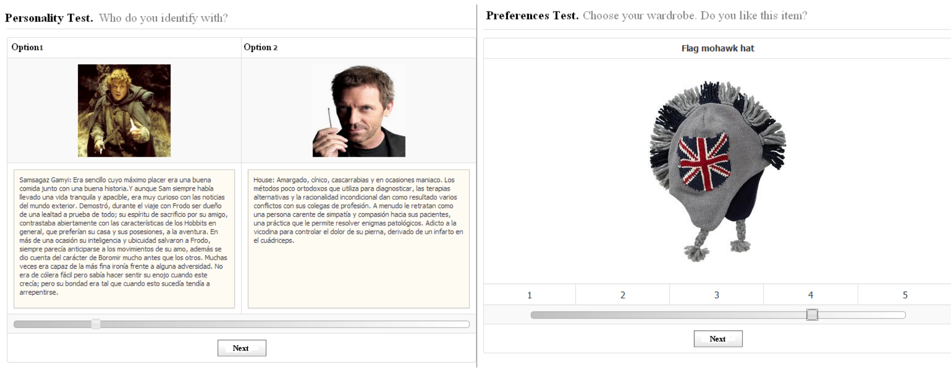


Figure 6.10: Personality test (left) and Preferences test (right) in *HappyShopping*.

social applications one being and individual recommender (*HappyShopping*) and the other one being a group recommender (*HappyMovie*), allows us to validate ARISE as a valid social architecture that comprises our *SRM*.

Moreover, *HappyShopping* exploits the fact that before and after consuming and item users might listen to their friends' opinions. And that these friends, that are believed to be trustworthy, may influence the user (this is the *IBR* approach seen in Section 4.3). This approach is a concrete instantiation of ARISE's generic *Individual Estimation Module* (see Figure 5.1) and the *Social Recommenders Design Templates*'s *Scoring* task (see Figure 5.2). Firstly, equally to *HappyMovie* and following our *SRM* represented in ARISE, the recommendation process followed by *HappyShopping* requires that users answer a personality test. This test (shown in Figure 6.10 left) follows the same approach of *HappyMovie*'s *TKI Metaphor* module (Section 6.3.1) and implements the TKI test (Thomas and Kilmann, 1974) that, as explained in Section 3.2 allows us to identify assertive personalities and cooperative personalities. Note that we here use this personality definition to measure the degree in which users might influence or be influenced. Secondly, and again following ARISE's architecture (this time the *Explicit Individual Preferences* module) and following the *Social Recommenders Design Templates*'s *tasks* (the *ObtainIndividualPreferences task*), users have to explicitly identify products that are of her/his interest. These will form the users' "wardrobe" (see Figure 6.10 right). Finally, we model the impact of the opinions of the people that are close in the social environment of the user requesting the recommendation and therefore might influence in her/his opinion[5]. Along with the degree in which a user may be influenced according to her/his per-

[5]Note that here closeness refers to our trust measure and is again equally implemented as *HappyMovie*'s *Facebook Profile Analysis* module (Section 6.3.2), implementation that is also included in our *Social Recommenders Design Templates* (*ObtainTrustFactors* task) and that therefore can be reused.

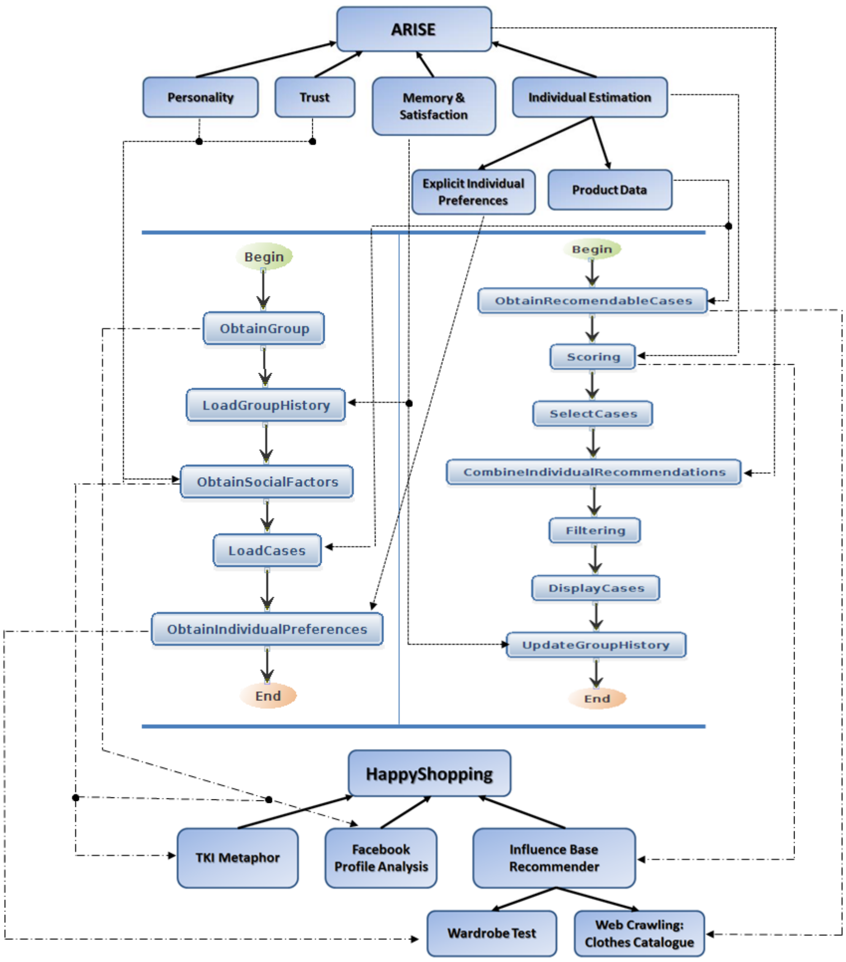


Figure 6.11: Relationship between the ARISE architecture, the *Social Recommenders Design Templates* and its instantiation in *HappyShopping*.

sonality. This is done through the *IBR* function explained in Section 4.3 and implemented in our *Social Recommenders Design Templates*' *Scoring* task.

To better understand how each module of ARISE defines a *task* of the templates and how *HappyShopping* implements these *tasks*, we introduce Figure 6.11. On top of the Figure we see ARISE's modules, each line that goes out of a module points the concrete *task* in the *Social Recommenders Design Templates* that defines it and each line that comes out of a *task* in the templates points the concrete module in *HappyShopping*'s structure that implements it.

After *HappyShopping* was finished, we performed an informal evaluation about its development cost. To do so, we asked to the 3 developers that built it about the effort and viability of using ARISE and our *Social Recommenders*

Design Templates. The answers of the questionnaire we ran can be found in the contribution presented in Chapter 21, (Quijano-Sánchez et al., 2013b, 2014c). Summing up, they answered that the templates had facilitated and quickened their work and that they preferred to have the templates to assist them. To measure the effort, we asked them how long it took them to build an initial version of the application, they answered that it took them 5 weeks to develop an initial version and 10 weeks to develop the final version of *HappyShopping*. If we compare these results with the time that took us to develop *HappyMovie*, that was more than 5 months, we can conclude that the usage of our social templates and ARISE has been a success.

6.7 Conclusions

In this chapter we have presented two use cases: *HappyMovie* and *HappyShopping*. Through them we have validated our generic architecture ARISE by being able to develop two different applications based on its structure. The applications are different in two aspects: (1) one is a social group recommender, *HappyMovie*, and the other one is a social individual recommender, *HappyShopping*. This fact allows us to conclude that ARISE is a valid architecture for the two existing types of recommenders, group and individual. (2) The applications have been developed in two different domains, movies, *HappyMovie*, and clothing, *HappyShopping*. This fact allows us to conclude that ARISE is a valid architecture for different domains. Although the two use cases have been developed in two different domains, movies and clothing, both domains comprise rateable and easily classifiable items. Because of this, we believe that ARISE is a valid architecture for many other domains, such as trips or books, as long as the items in the chosen domain have these characteristics: rateable and classifiable. These two properties match the goals of the two modules in ARISE that are domain dependent: *Product Data Module* and *Explicit Individual Preferences Module* (see Section 5.2). As for the applicability range of ARISE in different social networks, in both use cases we have limited ourselves to perform tests in Facebook, for implementation reasons mainly. Besides, as we previously mentioned, Facebook provides an ideal platform to develop applications because it allows users to interact, by creating events or completing interactive tests, it gives visibility and ease access, due to the relevance and number of users of the network, and it stores social information that allows us to compute the *trust* factor. As we saw in Chapter 2, Section 2.2.1, Facebook is what we call a *Human social network*. This is why, even though we have not proved it through experiments (task that we leave for future work), we believe that the reproductivity of our *SRM* in other social networks is limited to this type of networks, in contrast to *Content social networks* (like Twitter), since, as seen in Section 2.2.1 the first ones are of a more intimate nature making it easier to infer trust models and/or emotional contagion.

In this chapter we have also validated our *Social Recommenders Design Templates*. To do so, as we saw in the previous section and in papers (Quijano-Sánchez et al., 2013b, 2014c), we have performed an informal evaluation with 3 developers that used our templates to build *HappyShopping*. These developers were able to reuse our previously designed *methods* (included in the templates developing process explained in the previous chapter) and avoided starting the developing process from scratch. As future work we would like to develop more implemented *methods* for the templates in order to allow the templates to represent more heterogenous systems and also to give them more visibility, so that more developers use them and benefit from them.

Finally, and in a more general way, we have also been able to validate our *SRM*, after the positive evaluation that a significative sample of users gave when we asked them to perform and experimental evaluation of *HappyMovie*, seen in Section 6.5 and detailed in (Quijano-Sánchez et al., 2014b).

These facts allow us to validate this Thesis last hypothesis: ***"H4: It is possible to validate and evaluate our generic architecture*** ARISE ***through different concrete applications in different domains"***.

Next, in the last chapter of this PhD Thesis we will conclude the work here presented and introduce some lines of future work.

Chapter 7

Conclusions and future work

I'm glad to be with you, Samwise Gamgee, here at the end of all things.

Frodo Baggins

The main goal pursued in this PhD Thesis has been the improvement of group recommender systems through the inclusion of social factors. To do so, at the beginning of this PhD Thesis we formulated the hypothesis that, contrary to the more simpler aggregation strategies that have been mainly being used in group recommender systems (Lieberman et al., 1999; Crossen et al., 2002): ***"The real satisfaction of a group regarding a group recommendation cannot be accurately estimated using the simple aggregation of its members' individual preferences. Considering people as social entities that relate with each other allows the better estimation of their individual satisfaction regarding the result of the recommendation and, therefore, improves the global group satisfaction"***. To prove this hypothesis we have studied the "*Impact of social factors and organizations in group recommendation processes*" and, we have been able to conclude that indeed the inclusion of social factors to group recommendation processes improves the performance of existing group recommendation techniques. These conclusions are supported by this Thesis main contributions –i.e. the papers presented in Chapters 8 to 23–.

In this chapter we review the goals and main contributions completed in this work, which have allowed us to prove this Thesis hypothesis. We finish this document with some proposals for future work.

7.1 Conclusions

After introducing the work proposal in Chapter 1 we have verified the hypothesis presented in each chapter by completing the following goals:

- **Hypothesis 1 (H1)**: ***"There is a need to improve group recommender systems by better modelling decision-making processes, possibly through the inclusion of social factors".***
  Goal 1 (G1)→ Study the elicitation and usage of social factors in group recommendation processes and their ability to ease group decisions: In Chapters 2 and 3 we have studied the usage of social factors in group recommendation processes and their ability to ease group decisions. The result of this study is summarized in the following contributions:

  – **Contribution 1→ Study of existing recommender systems and different group and individual recommendation techniques**: In Chapter 2, we have studied the most representative recommender systems. This study has covered different recommendation techniques, both for individuals and groups, known to date and an overview of the main group recommender systems.

  – **Contribution 2→ Study of social factors in recommender systems and evaluation of Social Networks and the information that can be extracted from them**: In Chapter 2, we have studied the social factors that other researchers have so far included in recommender systems and the rising importance of Social Networks in the last few years, including the trust models that can be inferred from them and the different recommenders that have been designed using the information stored in them.

  – **Contribution 3→ Identification and study of people's group behaviour (in relation to conflict solving) according to their personality**: In Chapter 3 we have summarized our study of people's group behaviour (in relation to conflict solving) according to their *personality*. During this study we have proposed the inclusion, in group recommendation processes, of a factor that simulates the behaviour and reactions in conflict situations of the different group members (comprised in a factor that represents personality). This factor is a key element when personalizing the recommendations according to group characteristics. Besides, the personality factor has proven to be useful when trying to better model the decisions and argumentations that a group of people follow when trying to choose an item to consume together. The papers that cover this contribution are the ones included in Chapters 7, (Quijano-Sánchez et al., 2009) and 10, (Quijano-Sánchez et al., 2010).

  – **Contribution 4→ Identification of social factors that influence in people's trust and how to elicit them from Social Networks**: In Chapter 3 we have summarized our study of the different social factors that influence in people's *trust*. During this process we have studied which are the best variables to predict the *tie strength* between users (a.k.a. our trust factor), how to estimate this factor by automatically extracting information stored in Social Networks and how to use it in group recommendation processes. The papers that cover this contribution are the ones included in Chapters 10, (Quijano-Sánchez et al., 2010), 18, (Quijano-Sánchez et al., 2013c) and 23 (Quijano-Sánchez et al., 2014b).

- **Contribution 5→ Identification of other social factors that influence in group decision-making processes**: In Chapter 3 we have also presented other social factors, *homophily*, *persuasiveness* and *justice*, that influence in group decision-making processes and that therefore are worthy of studying when modeling these kind of processes. The papers that cover this contribution are the ones included in Chapters 12, (Quijano-Sánchez et al., 2011d), 19, (Recio-García et al., 2013) and 13, (Quijano-Sánchez et al., 2011c).

- **Hypothesis 2 (H2)**: ***"It is possible to develop group recommender systems that model groups' social behaviour by including social factors".***
Goal 2 (G2)→ Development of our *SRM* through the inclusion of the social factors identified in the previous goal: After proving the usefulness of including social factors that help us to better model decision making processes in conflict situations (Quijano-Sánchez et al., 2009, 2010, 2013c) we have developed our *Social Recommendation Model* (*SRM*). In Chapter 4 we have summarized our development of the *SRM* and the different social-based methods that it includes. The result of this development is summarized in the following contributions:

  - **Contribution 6→ Proposal of recommendations based on delegation, DBR (Delegation-Based Recommendations)**: In this method we propose a way of combining the *personality* and *trust* factors so that recommendations to each group member are based in the rest of the group members' preferences. The performed experiments with this technique, summarized in Chapter 4, prove that, among all the different social recommendation techniques that we propose, this one is the one that obtains better results. The papers that cover this contribution are the ones included in Chapters 12, (Quijano-Sánchez et al., 2011d), 13, (Quijano-Sánchez et al., 2011c) and 18, (Quijano-Sánchez et al., 2013c).

– **Contribution 7→ Proposal of recommendations based on influence, IBR (Influence-Based Recommendations)**: In this method we propose a way of combining our social factors of *personality* and *trust* so that recommendations to each group member are modified according to the influence that other group members have in her/him. The paper that covers this contribution is the one included in Chapter 18, (Quijano-Sánchez et al., 2013c).

– **Contribution 8→ Proposal of recommendations based on *coalitions***: In this method we propose the usage of our *SRM* main social factors, *personality* and *trust*, along with one additional factor, *homophily*. The *coalitions*-based technique es the identification of leaders inside a group and how these leaders can try to form coalitions that support their preferences. The paper that covers this contribution is the one included in Chapter 12, (Quijano-Sánchez et al., 2011d).

– **Contribution 9→ Proposal of recommendations based on Distributed Models and Argumentation**: In this method we study how to model group decision-making processes through dynamic argumentations. To do so, we propose the use of multi-agent systems that follow a social network topology, where each agent represents a group member. Besides, in this technique we add the *persuasiveness* social factor to the other two main social factors of *personality* and *trust*. The papers that cover this contribution are the ones included in Chapters 9, (Recio-García et al., 2010) and 19, (Recio-García et al., 2013).

– **Contribution 10→ Proposal of recommendations based on memory of past recommendations**: In this method we propose a way of combining the *personality* and *trust* factors with a new social factor, *justice*. To do so, we use a memory of past recommendations that avoids repeating recommendations and offers new recommendations that ensure an homogeneous satisfaction level between group members. The papers that cover this contribution are the ones included in Chapters 13, (Quijano-Sánchez et al., 2011c), 18, (Quijano-Sánchez et al., 2013c) and 23, (Quijano-Sánchez et al., 2014b).

– **Contribution 11→ Proposal of recommendations that solve the *cold-start* problem**: In this method we propose the use of social factors in group recommender systems to define social similarity measures between users and groups. These measures are later used to allocate to users who have very few ratings –they are in *cold-start* and it is therefore very difficult to perform good estimations for them–ratings from the most similar user from the most similar group. The papers that cover this contribution are the ones included in Chapter 16, (Quijano-Sánchez et al., 2012b, 2013a).

- **Contribution 12→ Proposal of social recommendations based on CBR (Cased-Based Reasoning)**: In this method we propose the use of social factors along with CBR techniques to fulfill our goal of defining similarity measures between users and groups. These measures, help us to later simulate the behaviour of the most similar users in the most similar groups, predicting this way how the members of the group receiving a recommendation will behave. This method is presented as an alternative to the previous methods (that present predefined equations), considering the possibility that each defined equation may not always be the most suitable for each different group. The paper that covers this contribution is the one included in Chapter 17, (Quijano-Sánchez et al., 2012a).
- **Contribution 13→ Evaluation of our *SRM* for the different existing aggregation strategies**: During this evaluation process we have studied existing simple aggregation techniques (Masthoff, 2004) and have implemented all of them for our *SRM*. Next, we have performed experiments with both, real and synthetic users, to test which aggregation technique works better for the different possible group configurations (small or big groups) and recommender configurations (*Non Social*, *personality*-based, *trust*-based or with both of our main social factors). The results of these experiments show that, in general, the aggregation technique that obtains the best results in our *SRM* is the *Average Satisfaction* and that those recommenders that include both of our main social factors, *personality* and *trust*, are the ones that obtain the best results. The papers that cover this contribution are the ones included in Chapters 14, (Quijano-Sánchez et al., 2011a) and 22, (Quijano-Sánchez et al., 2014a).
- **Contribution 14→ Evaluation of the proposed methods**: During this evaluation process we have proven that our *SRM* improves the performance of the recommender systems that do not use the social factors that we include (Equation 4.2). We have performed experiments with both, real and synthetic users, and have tested the precision and efficiency of our *SRM* in the movie recommendation domain. Besides, we have proven the viability of using synthetic data, as its results are equivalent to the ones obtained with real users. The paper that covers this contribution is the one included in Chapter 22, (Quijano-Sánchez et al., 2014a).

• **Hypothesis 3 (H3)**: ***"It is possible to generalize our*** *SRM* ***in***

a way that it is applicable to different domains and in a way that other recommender systems developers can reuse".
Goal 3 (G3)→ Provide a generic architecture and a development methodology that allows the instantiation of our *SRM*: After presenting our *SRM* and verifying its relevance regarding the improvement of group recommendation techniques, we wanted to provide a generic architecture and a development methodology that allows the instantiation of the proposed model. This development, that has been presented in Chapter 5, is summarized in the following contributions:

- **Contribution 15→ Proposal a reusable generic architecture,** ARISE: During this process we abstracted our *SRM*. To do so, we have modulated and organized our *SRM* so that it can be reused in other domains, besides of the already tested movies domain. The result of this process is our generic architecture ARISE. The papers that cover this contribution are the ones included in Chapters 21, (Quijano-Sánchez et al., 2013b, 2014c) and 22, (Quijano-Sánchez et al., 2014a).
- **Contribution 16→ A semi-automatic instantiation of the** ARISE **architecture through the usage of *Social Recommenders Design Templates*:** During this process we have designed a series of templates, called *Social Recommenders Design Templates*, that represent an intermediate step between ARISE and any other social application that can be built using its structure. The papers that cover this contribution are the ones included in Chapter 21, (Quijano-Sánchez et al., 2013b, 2014c).

- **Hypothesis 4 (H4)**: ***"It is possible to validate and evaluate our generic architecture*** ARISE ***through different concrete applications in different domains".***
Goal 4 (G4)→ Development of an application in a social network that validates our *SRM*: To validate the contributions presented in the previous goal we have developed two different use cases in social environments. Both of them have been developed in the social network Facebook for practical reasons. One in the movie group recommendation domain and other in the individual clothing recommendation domain. These use cases, that have been presented in Chapter 6, are summarized in the following contributions:

  - **Contribution 17→ Development of an application in the social network Facebook that implements** ARISE **and the proposed social recommendation techniques of our *SRM*, *HappyMovie*:** With this application we have proven the importance of social factors in group recommendation processes and the efficiency of our *SRM*. With the application as tool, we have been able to perform an evaluation with real users where we have verified users' acceptance of *HappyMovie*'s recommendations and the easy usage of the application. The papers that cover this contribution are the ones included in Chapters 11, (Quijano-Sánchez et al., 2011e), 15, (Quijano-Sánchez et al., 2011b), 20, (Quijano-Sánchez and Bridge, 2013) and 23, (Quijano-Sánchez et al., 2014b).

– **Contribution 18→ Development of an application in the social network Facebook, *HappyShopping*, that proves that the ARISE architecture is viable for other domains and that the proposed *Social Recommenders Design Templates* ease the develop of new social applications**: With this application we have proven that our *SRM*, that was initially tested only in the movies domain, is valid for other domains. Besides, our *Social Recommenders Design Templates* and the ARISE architecture have been used during this application's building process, proving this way their viability and usefulness. The papers that cover this contribution are the ones included in Chapter 21, (Quijano-Sánchez et al., 2013b, 2014c).

Summing up, the results that we have presented throughout this PhD Thesis, and that are in turn in the published papers presented as core of this document (part III), hold that:

1. Our *SRM* is indeed useful and improves the performance of group recommendation methods based on simple aggregation of preferences.

2. Our work leads to progress in the state of the art.

3. The methods that we have presented throughout the developed work are novel.

4. The results of the performed experiments both with real and synthetic users are significant.

5. The results of our research have been presented to the scientific community.

6. The result of our work proposes a new line of research that is currently being followed by other researchers (Gaillard et al., 2014; Kompan and Bieliková, 2014; Leonard, 2014; Christensen and Schiaffino, 2014), with more than 80 references, ruling out self-references, to our work.

Next, to conclude, we present some lines of future research work.

7.2 Future work

After the completion of this Thesis work, we propose 3 lines of future work that we feel are worth pursuing.

7.2.1 Adaptive social group recommenders

Throughout this PhD Thesis performed experiments (Quijano-Sánchez et al., 2010, 2013c, 2014a), we have observed that not all group compositions are equal and therefore not all the different recommendation techniques work equally well for groups of different sizes, personality, trust distributions, etc. Because of this, it would be interesting to design recommenders whose recommendation techniques adapt to different group configurations, being this way able to optimize the obtained results. As a result of this, a future line of work would be: *The study of groups' structure and composition for the usage of social adaptive group recommenders.* We understand as adaptive recommenders as the ones that automatically choose which group recommendation approach fits better for a given group based on the group's characteristics. This adaptability can be related to either our *SRM* methods (by choosing the *DBR* method, the *IBR* method, etc) or to the aggregation techniques applied in those methods (*Average Satisfaction*, *Least Misery*, etc).

Following this line of research in adaptive recommenders, another possible step in the improvement of group recommender systems could be the analysis of group behaviour according to the characterisation of the group, for example, age distribution inside the group. In this PhD Thesis work we have used as a starting-point the premise that the groups to be recommended were groups of friends performing joint activities. A completely different situation would be family group recommendation processes, in these situations, age difference (elder, kids) considerably varies the possible activities and the different priorities that must be taken into account when trying to satisfy the different group members. Following this line of thought, there is a need for adaptive recommenders that automatically limit the set of items to be recommended to a suitable age and stablish different weights and priorities following a study of how groups, for example with kids, tend to behave.

Also, related to establishing different priority weights to each group member, a line of work that we have not studied in this Thesis and that would be interesting to develop is groups' context dependency. This is, a group of friends may not have the same behaviour one day or another. Whether because it is someone's birthday in which case the preferences of this group member may have more weight – because it is a special date, Halloween for example, where it is typical to watch terror movies, in which case this kind of genre would have more weight –, or because of the general emotional state of the group members, where they could point out that they are not in a good mood and feel like watching a comedy for example.

7.2.2 Cased-based social group recommenders

In Chapter 4, Section 4.8, we have presented our study that uses past recommendations to groups in order to "replay" the behaviour of each user in a group. This way, we avoid using prefixed equations that may not always be equally productive in all possible group configurations. A possible future research line could be carrying on with this idea and studying group structures (the graph that they form) and according to them, develop techniques based on previous cases. This line of future work could be defined as: *The analysis and storage of behaviours and group recommendations for its later reuse in group recommender systems based in CBR*. Following this line of work, in (Quijano-Sánchez and Bridge, 2013) we set out as future work the necessity of building a case base (of groups and the joint activities they perform) of complex and detailed cases for its future reuse[1]. To do so, CBR (Cased-Based Reasoning) might be very well-suited to this task. After all, CBR is all about reasoning with experiences (Leake, 1996). A rich case structure could capture multiple aspects of an ongoing activity and its decision-making process. For example, the problem description part of the case could contain some or all of the following: (a) information about each member of the group, demographic information, personality information, and information about tastes, e.g. in the form of ratings; (b) information about relationships between group members; (c) the candidate movies, i.e. the ones from which the recommender made its recommendations; (d) predicted ratings for each group member and each candidate item; and even (e) predictions about the other dimensions (user satisfaction and the group experience). On the other hand, the solution part of the case could contain at least the item or items that were recommended and might contain more than this (e.g. the ranking of all the candidate items). Besides, it is well-known that groups tend to recur (with small variations) and groups structures (such as a parent and her/his children, or a group of university-age friends) also recur. Therefore, a CBR assumption (where similar problems have similar solutions) could perfectly fit.

7.2.3 Social explanations in group recommenders

Throughout this PhD Thesis work we have studied different group recommendation techniques that integrate social factors (our *SRM*). Besides, we have built a use case of our model in the social network Facebook, *HappyMovie*, which is an application for groups of friends that wish to go to the movies. As we have seen throughout this document, the system *HappyMovie* tries to alleviate certain existing limitations in group recommender systems, like the obtention of users' profiles (that requires users' time and effort) or

[1] Note that up to date there are no existing case bases where the cases are groups of people that have performed joint activities.

the way of presenting group recommendations (by introducing an accesible system embedded in a daily used social network that helps users and eases conflict situations).

A possible line of future research could be the evaluation of the impact that group explanations in social recommender systems could have in users response towards the recommendations (this is a novel proposal as we have not found any work that studies this area of recommender systems). To do so, we could present to the group receiving the recommendation different types of social explanations. This line of future work could be defined as: *The study of social explanations in group recommender systems*. In these explanations, we would try to explain why a recommender system predicts that a particular selected item is the one that best reconciles the group members' expectations. More concretely, we would evaluate the viability and usefulness of including either graphical or textual explanations and the impact that they have in users' acceptance and trust in the system.

Once the system provides a recommendation it is natural that group members should want to understand to some extent how a recommendation was arrived at - and in particular, how attractive a recommended item is likely to be to each individual group member -. Hence, some recommender systems provide along with the recommendation itself an explanation of how it is reached (Herlocker et al., 2000; Tintarev, 2007). An example of this type of system, that uses explanations to justify the proposed outcome, is Let's Browse (Lieberman et al., 1999). Explanations in recommender systems present multiple variations, they can go from a simple confidence value to a complex visualization of the pros and cons of a solution.

And additional contribution of this PhD Thesis performed work would be including in our *SRM* a model that explains users why has the system predicted that a particular item is the best choice for a group. This technique is totally novel, as to date no research had been performed in the explanations to groups area. To do so, firstly, we would study the state-of-the-art related to recommender systems that provide explanations. Next, we would propose different alternatives to provide explanations to the group of users receiving a recommendation taking into account different group aspects like users individual preferences predictions or the social factors studied in our *SRM*. This research line is novel in the recommenders' area since our goal is not only to extend explanations research to group recommenders (up to date the works in explanations have focused in explanations to individual users and not to groups) but also to include the social factors that this Thesis includes in an innovative way in the existing explanations techniques. This last factor poses a greater challenge as explanations of the social factors involved in decision-making processes (and the ones used in our *SRM*), like the *personality* of each individual or the *trust* between users should be treated as a delicate subject, where users sensibility and their relations are at stake.

Because of this, our goal should be to produce clear and useful explanations but also tactful explanations. With these approach could try to increase of users' acceptance of the recommendations and users' trust in the system.

7.3 Final conclusions

Throughout this PhD Thesis we have studied recommender systems, more concretely group recommender systems. After identifying some shortages that these systems suffer, when modelling social behaviour inside a group, we have proposed a model –SRM– based on the inclusion of social factors. The model is composed by a set of methods, a generic architecture that comprehends it, a set of templates that allow its reuse and two applications that instantiate it. Besides, we have been able to prove that the usage of our model leads to an improvement in group recommender systems. There is still a lot of work to do in the area of group recommender systems and in the area of social recommender systems as they are two rising research lines. However, this Thesis work has successfully resolved the posed hypothesis and identified new lines of future work. With it, we close this stage of our research and start a new journey.

Parte II

Contribuciones

Capítulo 1

Introducción

Los sistemas de recomendación para grupos de usuarios son actualmente una de las líneas de investigación más activas dentro del área de recomendadores. Los sistemas de recomendación clásicos –i.e. individuales– han demostrado su importancia y repercusión dentro de la industria al ser uno de los pilares centrales del comercio on-line (Adomavicius y Tuzhilin, 2005; Ricci et al., 2011). Por ejemplo, sitios Web tan populares como Amazon[1] utilizan estas técnicas para guiar al usuario en la selección de productos. Los recomendadores individuales se centran en el caso más simple: un único usuario consumiendo un producto. Sin embargo, ignoran el hecho de que algunos productos, sobre todo productos de ocio, se consumen en grupo, ya sea en pareja, en familia o con amigos. La elección de una película en una cita, de unas vacaciones familiares o de un restaurante donde cenar requiere un balance de las preferencias de los miembros del grupo. Esta clase de recomendaciones es la que llevan a cabo los sistemas de recomendación grupal (Jameson y Smyth, 2007). El proceso que normalmente siguen estos sistemas es la agregación de los *ratings*[2] (ya sean reales o estimados) de cada uno de los miembros del grupo (Jameson y Smyth, 2007; Baltrunas et al., 2010; Berkovsky y Freyne, 2010; Pessemier et al., 2013). Para esta tarea se utilizan típicamente funciones de agregación inspiradas en funciones de bienestar social desarrolladas en el campo de investigación denominado *Social Choice Theory* (Masthoff y Gatt, 2006). Sin embargo, mientras que este enfoque está ampliamente aceptado, ignora ciertos factores sociales que influyen en los procesos de toma de decisiones en grupo. Este trabajo de Tesis nace de esta carencia y propone un nuevo modelo de recomendación grupal, llamado *Modelo de Recomendación Social* (*MRS*), que incluye factores sociales.

Con la llegada de la Web Social (Open Diary, 1998[3]) y la creciente popu-

[1] `http://www.amazon.com/`

[2] Denominamos ratings a las valoraciones que hacen los usuarios sobre diferentes productos.

[3] `http://www.OpenDiary.com`

laridad de las redes sociales[4], donde los usuarios proporcionan explícitamente información personal a la vez que interactúan con otros y con el sistema, aparece la oportunidad para desarrollar nuevas estrategias de recomendación que se ayuden de estas nuevas fuentes de conocimiento social. En nuestro modelo planteamos la explotación de las redes sociales como plataforma de ejecución y/o fuente de información social, demostrando que dicha aproximación mejora el rendimiento respecto a otras técnicas de recomendación grupal. En nuestra propuesta, como veremos a lo largo de este trabajo de Tesis, destacamos dos aspectos sociales dentro de la dinámica de un grupo: el papel individual de cada componente del grupo, que depende de la personalidad o capacidad persuasiva de cada individuo. Y, por otro lado, las relaciones y comportamientos entre los componentes del grupo, que dependen de la confianza entre los componentes del grupo o del sentido de la justicia, y surgen cuando se estudia al grupo como una entidad propia.

Típicamente, los sistemas de recomendación grupal incluyen técnicas de predicción de ratings y de agregación (Jameson y Smyth, 2007). Siguiendo este esquema, el sistema funcionaría como sigue: Primero, para cada miembro del grupo, se utiliza un sistema de recomendación individual que predice una serie de ratings para un conjunto de productos candidatos. Segundo, el recomendador grupal agrega estos ratings: por ejemplo para cada producto candidato, puede utilizar la media de los ratings de los miembros del grupo, o el mínimo, o el máximo (Masthoff, 2004). Finalmente, el sistema recomienda al grupo los productos que obtengan el mayor rating después de la agregación. Sin embargo, esta aproximación no tiene en cuenta que los grupos de personas pueden tener características muy diferentes, ya sean estructurales como el tamaño, de preferencias similares o antagónicas, o de diferentes tipos relaciones sociales. Es aquí donde nace la hipótesis general que hemos formulado en esta Tesis Doctoral: ***"La satisfacción real de un grupo de personas respecto a una recomendación grupal no se puede estimar fielmente utilizando una agregación simple de las preferencias individuales de cada uno de sus miembros. La consideración de las personas como entidades sociales que se relacionan permite mejorar la estimación de su satisfacción individual respecto al resultado de la recomendación y, por lo tanto, mejorar la satisfacción global del grupo"***. Concretamente hemos estudiado dos factores fundamentales: *personalidad* y *confianza* y otros tres factores secundarios: *homofilia*, *persuasividad* y *justicia*. Estos factores definen a cada persona como un componente potencialmente influenciable de una comunidad (o grupo) determinado por su entorno, que en la mayoría de los casos son las Redes Sociales a las que pertenece. Para contrastar dicha hipótesis hemos diseñado y evaluado una serie de métodos que integran factores sociales en distintas técnicas de recomendación grupal. El conjunto de estos métodos de recomendación social

[4]Hoy en día (2015) existen mas de 200 redes sociales (y casi 2 billones de usuarios).

componen lo que denominamos el *Modelo de Recomendación Social* (*MRS*), contribución principal de esta Tesis. En él, simulamos el comportamiento individual de un individuo basándonos en la contrastada idea de que la relación entre un individuo y las redes a las que pertenece influye directamente en las relaciones en su vida (Christakis y Fowler, 2011). Además del desarrollo de estos métodos desde el punto de vista funcional, proponemos una arquitectura genérica llamada ARISE (*Architecture for Recommendations Including Social Elements*) que puede ser utilizada para instanciar el *MRS* propuesto en recomendadores concretos de distintos dominios y aplicaciones. Además, proporcionamos una metodología para el desarrollo de software cuyo propósito es facilitar la instanciación de la arquitectura ARISE para la construcción de nuevas aplicaciones de recomendación social. Esta metodología, siguiendo una línea de investigación previa (Díaz-Agudo et al., 2007), se basa en plantillas (Quijano-Sánchez et al., 2013b) que formalizan el comportamiento funcional de los sistemas de recomendación social y facilitan su configuración y desarrollo. El factor común y clave que tienen los diferentes tipos de recomendadores que se pueden construir en distintos dominios utilizando esta arquitectura genérica es la inclusion de factores sociales. Para ilustrar y validar las mejoras que supone la utilización de nuestro *MRS* y la viabilidad de nuestra plataforma genérica ARISE, lo hemos instanciado en dos sistemas de recomendación reales: el recomendador grupal *HappyMovie*[5], que es una aplicación en la red social Facebook que representa una instanciación particular de nuestra arquitectura genérica ARISE en el dominio de las películas. Y el recomendador individual *HappyShopping*, que es una aplicación en la red social Facebook que representa otra instanciación particular de ARISE siguiendo nuestra metodología de desarrollo basada en plantillas esta vez, en el dominio de la ropa. Además, también utilizaremos *HappyMovie* como caso de uso y entorno experimental donde poder evaluar nuestra arquitectura ARISE y nuestro *MRS* con datos reales.

1.1. Hipótesis y objetivos

A la hora de organizar el trabajo de investigación desarrollado a lo largo de esta Tesis Doctoral se formularon una serie de hipótesis (que desglosan la hipótesis general anteriormente descrita) a demostrar y, respectivamente, una serie de objetivos a cumplir para poder validar estas hipótesis. Estos objetivos han dado lugar a diferentes aportaciones en su área que han quedado reflejadas en los artículos publicados que se presentan como núcleo de este trabajo (Parte III de este documento).

A continuación desglosamos las hipótesis planteadas y sus correspondientes objetivos, las aportaciones que estos dieron lugar y los artículos resultantes:

[5]http://gaia.fdi.ucm.es/research/happymovie

Hipótesis 1 (H1)
Existe la necesidad de mejorar los sistemas de recomendación grupal por medio de un modelado más detallado de los procesos de toma de decisiones, posiblemente mediante la inclusión de factores sociales.

Objetivo 1 (O1)
Estudiar la obtención y el uso de los factores sociales en los procesos de recomendación grupal para facilitar la toma de decisiones en grupo.

- **Aportación 1**: Estudio de sistemas recomendadores existentes, incluyendo diferentes técnicas de recomendación individual y grupal.

  Contribuciones aportadas:

  - Capítulo 2. Estado del Arte.

- **Aportación 2**: Estudio de factores sociales en los sistemas de recomendación y evaluación de las redes sociales y la información que se puede extraer de ellas.

  Contribuciones aportadas:

  - Capítulo 2. Estado del Arte.

- **Aportación 3**: Identificación y estudio del comportamiento, respecto a la resolución de conflictos, de las personas en un grupo en función de su personalidad.

  Contribuciones aportadas:

  - Capítulo 8, (Quijano-Sánchez et al., 2009).
  - Capítulo 10, (Quijano-Sánchez et al., 2010).

- **Aportación 4**: Identificación de los factores sociales que influyen en la confianza entre personas y cómo obtenerlos a través de las redes sociales.

  Contribuciones aportadas:

  - Capítulo 10, (Quijano-Sánchez et al., 2010).
  - Capítulo 18, (Quijano-Sánchez et al., 2013c).
  - Capítulo 23 (Quijano-Sánchez et al., 2014b).

- **Aportación 5**: Identificación de factores sociales adicionales que influyen en los procesos de toma de decisiones en grupo.

  Contribuciones aportadas:

  - Capítulo 12, (Quijano-Sánchez et al., 2011d).

- Capítulo 19, (Recio-García et al., 2013).
- Capítulo 13, (Quijano-Sánchez et al., 2011c).

Hipótesis 2 (H2)
Es posible desarrollar sistemas de recomendación grupal que modelen el comportamiento social que tienen los grupos de personas mediante la inclusión de factores sociales.

Objetivo 2 (O2)
Desarrollar nuestro MRS mediante la inclusión de los factores sociales identificados en el objetivo anterior.

- **Aportación 6**: Propuesta de un método de recomendación basado en delegación, DBR (Delegation-Based Recommendations).

  Contribuciones aportadas:

  - Capítulo 18, (Quijano-Sánchez et al., 2013c).
  - Capítulo 22, (Quijano-Sánchez et al., 2014a).

- **Aportación 7**: Propuesta de un método de recomendación basado en influencia, IBR (Influence-Based Recommendations).

  Contribuciones aportadas:

  - Capítulo 18, (Quijano-Sánchez et al., 2013c).

- **Aportación 8**: Propuesta de un método de recomendación basado en coaliciones.

  Contribuciones aportadas:

  - Capítulo 12, (Quijano-Sánchez et al., 2011d).

- **Aportación 9**: Propuesta de un método de recomendación basado en modelos distribuidos y argumentación.

  Contribuciones aportadas:

  - Capítulo 9, (Recio-García et al., 2010).
  - Capítulo 19, (Recio-García et al., 2013).

- **Aportación 10**: Propuesta de un método de recomendación basado en memoria.

  Contribuciones aportadas:

  - Capítulo 13, (Quijano-Sánchez et al., 2011c).
  - Capítulo 18, (Quijano-Sánchez et al., 2013c).
  - Capítulo 23, (Quijano-Sánchez et al., 2014b).

- **Aportación 11**: Propuesta de un método de recomendación para resolver el problema del *cold-start*.

  Contribuciones aportadas:

  - Capítulo 16, (Quijano-Sánchez et al., 2012b, 2013a).

- **Aportación 11**: Propuesta de un método de recomendación social basado en CBR (Cased-Based Reasoning).

  Contribuciones aportadas:

  - Capítulo 17, (Quijano-Sánchez et al., 2012a).

- **Aportación 13**: Evaluación de nuestro MRS utilizando las diferentes técnicas de agregación existentes. Contribuciones aportadas:

  - Capítulo 14, (Quijano-Sánchez et al., 2011a).
  - Capítulo 22, (Quijano-Sánchez et al., 2014a).

- **Aportación 14**: Evaluación de los métodos propuestos.

  Contribuciones aportadas:

  - Capítulo 22, (Quijano-Sánchez et al., 2014a).

Hipótesis 3 (H3)
Es posible generalizar nuestro MRS de forma que sea aplicable a diferentes dominios y de forma que otros desarrolladores de sistemas de recomendación sean capaces de reutilizarlo.

Objetivo 3 (O3)
Proporcionar una arquitectura genérica y una metodología de desarrollo que permita la instanciación de nuestro MRS en distintos dominios.

- **Aportación 15**: Propuesta de una arquitectura genérica reutilizable: ARISE.

  Contribuciones aportadas:

  - Capítulo 22, (Quijano-Sánchez et al., 2014a).
  - Capítulo 21, (Quijano-Sánchez et al., 2013b, 2014c).

- **Aportación 16**: Instanciación semi-automática de la arquitectura ARISE por medio de *Plantillas de Diseño de Recomendadores Sociales*.

  Contribuciones aportadas:

  - Capítulo 21, (Quijano-Sánchez et al., 2013b, 2014c).

Hipótesis 4 (H4)
Es posible validar y evaluar nuestra arquitectura genérica ARISE por medio de distintas aplicaciones concretas en diferentes dominios.

Objetivo 4 (O4)
Desarrollo de una aplicación para validar nuestro *MRS* en una red social, en el dominio de las recomendaciones de películas para grupos y en el dominio de las recomendaciones de ropa para individuos en entornos sociales.

- **Aportación 17**: Desarrollo de una aplicación en la red social Facebook que implementa ARISE y recoge las técnicas de recomendación basadas en factores sociales propuestas en el *MRS*: *HappyMovie*.

  Contribuciones aportadas:

  - Capítulo 11, (Quijano-Sánchez et al., 2011e).
  - Capítulo 15, (Quijano-Sánchez et al., 2011b).
  - Capítulo 23, (Quijano-Sánchez et al., 2014b).

- **Aportación 18**: Desarrollo de una aplicación en la red social Facebook que demuestra que la arquitectura ARISE es viable para otros dominios y que las *Plantillas de Diseño de Recomendadores Sociales* propuestas facilitan el desarrollo de nuevas aplicaciones sociales: *HappyShopping*.

  Contribuciones aportadas:

  - Capítulo 21, (Quijano-Sánchez et al., 2013b, 2014c).

1.2. Estructura de la tesis

La memoria de esta Tesis se ha organizado en torno a los objetivos planteados en la sección anterior (O1-O4). Cada capítulo (del 3 al 7) corresponde con uno de los estos objetivos y cada sección con un resumen de cada una de las aportaciones (publicaciones) presentadas en la Parte III. Las secciones resumen el contenido de los artículos citados en ellas y los artículos representan la investigación realizada para alcanzar cada objetivo.

En el Capítulo 2, primero describimos el marco teórico de las recomendaciones en general y de las recomendaciones para grupos en particular. En este capítulo se expone en que consisten las recomendaciones grupales y se realiza un análisis general de trabajos previos relacionados (incluyendo trabajos que resaltan los beneficios de estudiar vínculos sociales, de hacer uso de las redes sociales y de utilizar la información que éstas contienen). En el Capítulo 3 se realiza un estudio sobre cómo mejorar los procesos de recomendación en el que realizamos una revisión sobre las ventajas de incluir factores sociales a los procesos de recomendación grupal junto con algunas propuestas de cómo obtener dichos factores sociales (O1). A continuación, en el Capítulo 4, introducimos nuestro *MRS* y presentamos diferentes propuestas de sistemas de recomendación grupal que lo utilizan (O2), estas propuestas son los diferentes métodos que componen el modelo. Además, se presentan diferentes experimentos que evalúan y validan nuestro *MRS*. El siguiente capítulo, Capítulo 5, aborda la reutilización y generalización de nuestro método a través del diseño de una arquitectura genérica y una metodología de desarrollo basada en plantillas de sistemas de recomendación social (O3). Más adelante, en el Capitulo 6, se expone como prueba de concepto un caso de estudio –*HappyMovie*– en el dominio de las películas donde se aplican las nuevas técnicas estudiadas y una evaluación sobre la eficiencia de éstas. Seguidamente, se expone un segundo caso de estudio en un dominio diferente que nos permite asegurar la viabilidad y reproducibilidad de nuestra arquitectura genérica y su metodología asociada (O4). Finalmente, en el Capítulo 7 se presentan las conclusiones obtenidas tras este trabajo de investigación así como las líneas de trabajo futuro.

Siguiendo la exposición anterior el trabajo de Tesis se estructura en los siguientes capítulos:

Capítulo 2. Estado del arte. Este capítulo presenta una visión general de los sistemas recomendadores individuales y para grupos. Seguidamente, se explica la importancia de las redes sociales en los últimos años y como de ellas se puede extraer información social para mejorar los sistemas de recomendación grupal. Por último, se presentan algunos ejemplos de las investigaciones realizadas en el campo de la recomendación social en los últimos años.

Capítulo 3. Estudio de la obtención y uso de los factores sociales en los procesos de recomendación grupal para facilitar la toma de decisiones en grupo. En este capítulo verificamos nuestra hipótesis H1 completando nuestro objetivo O1. Para ello, identificamos los diferentes factores sociales que incluye nuestro *MRS*, motivamos las ventajas de introducirlos y presentamos algunas ideas de como extraerlos.

Capítulo 4. Desarrollo de los métodos de recomendación social grupal que forman el *MRS*. En este capítulo verificamos nuestra hipótesis H2 completando nuestro objetivo O2. Para ello, se presenta nuestro *MRS* y diferentes enfoques que lo utilizan.

Capítulo 5. Arquitectura genérica y metodología de desarrollo para la instaciación del modelo. En este capítulo verificamos nuestra hipótesis H3 completando nuestro objetivo O3. Para ello, se explica el diseño de nuestra arquitectura genérica ARISE y se presentan nuestras *Plantillas de Diseño de Recomendadores Sociales*, que representan los pasos que se deben seguir para reutilizar nuestro *MRS*, ya sea con otros datos o en otros dominios.

Capítulo 6. Pruebas de concepto en una red social. En este capítulo verificamos nuestra hipótesis H4 completando nuestro objetivo O4. Para ello, se detalla nuestra aplicación *HappyMovie*. Esta aplicación es un sistema de recomendación grupal integrada en la red social Facebook, en el dominio de las películas, cuyo objetivo es la presentación de diferentes películas que el sistema estima que agradarán al grupo que ha solicitado la recomendación de acuerdo a la preferencias individuales de los miembros del grupo y sus relaciones sociales. Finalmente, se presenta un caso de estudio secundario, *HappyShopping*, esta vez en el dominio de la ropa. Con esta segunda aplicación demostramos la utilidad de usar ARISE y las *Plantillas de Diseño de Recomendadores Sociales* en un dominio diferente al que fueron diseñados.

Capítulo 7. Conclusiones y Trabajo Futuro. En este capítulo exponemos las conclusiones que hemos obtenido tras la realización de esta Tesis y comentamos las líneas de investigación que seguiremos en el futuro.

Capítulo 2

Estado del arte

2.1. Sistemas recomendadores

Los sistemas recomendadores nacen con el propósito de facilitar la toma de decisiones en temas/dominios en los que las posibilidades de elección son muchas y variadas. Actúan sugiriéndonos buenos productos y/o servicios bien sea para comprar o consumir (véase (Adomavicius y Tuzhilin, 2005; Ricci et al., 2011) como resumen). En la actualidad podemos encontrar recomendadores para todo tipo de productos: viajes, libros, películas, restaurantes, coches y un largo etcétera (Jung, 2012; McCarthy, 2002; Batet et al., 2012; Vaz et al., 2012; Jameson, 2004). Sin duda, un campo de gran aplicación de estos sistemas son las actividades relacionadas con el ocio. Algunas de estas actividades son típicamente realizadas en grupo en vez de en forma individual, luego tiene sentido realizar no sólo recomendaciones individuales sino recomendaciones a todo un grupo de personas que vaya a realizar alguna actividad conjuntamente (Jameson y Smyth, 2007).

En este capítulo presentamos los distintos tipos de recomendadores que existen, las diferencias entre ellos en función de su diseño y objetivos y las carencias que presentan, factor que ha motivado la realización de esta Tesis. Seguidamente introducimos el concepto de factores sociales, cuya inclusión a los sistemas de recomendación grupales será la principal aportación de esta Tesis. Para ello revisaremos primero el impacto que las redes y medios sociales han tenido en los últimos años, sus funciones, la información que se puede extraer de ellas y el uso que otros sistemas de recomendación han hecho de ellas.

2.1.1. Sistemas de recomendación para individuos

Existen dos grandes familias de recomendadores para individuos en base a la fuente de conocimiento: *basados en contenido*, aquellos que realizan la recomendación en base a la descripción de los elementos (Pazzani y Billsus, 2007), y *colaborativos* aquellos que utilizan información de los usuarios junto con las valoraciones que los usuarios hacen de los elementos (Bridge et al., 2005). Entre estos dos extremos, existen también múltiples aproximaciones *híbridas* (Burke, 2002) que mezclan características de recomendación *colaborativas* y *basadas en contenido*.

Estos enfoques (*colaborativos*, *basados en contenido* y *híbridos*) tienen diferentes puntos fuertes y debilidades. Un punto débil que la mayoría de los recomendadores presenta es el llamado problema del *cold-start* (Herlocker, 2000; Schafer et al., 2007b). Este problema ocurre cuando el sistema no tiene suficiente información sobre un usuario nuevo como para deducir nada sobre sus gustos. Otro punto débil, en este caso relacionado sólo con los recomendadores *basados en contenido*, que algunos sistemas presentan son recomendaciones que son demasiado *uniformes*. Por ejemplo, el sistema recomendador que tenía Amazon inicialmente sufría del llamado *efecto portafolio* (Linden et al., 2003; Burke, 2002), esto es que se ofrecían recomendaciones tan similares entre si que no servían al usuario. En lo referente a los diferentes puntos débiles que los recomendadores presentan podemos encontrar un resumen en la Tesis de Tintatev (Tintarev, 2009).

Los recomendadores individuales también se pueden clasificar atendiendo a las siguientes características:

- Quién toma la iniciativa. Podemos distinguir dos tipos en base a quién lleva la iniciativa en la recomendación. Así, podemos tener un recomendador *reactivo*, donde es el usuario quien lleva la iniciativa realizando una consulta al sistema. Es el caso del sistema TAAABLE (Cordier et al., 2012), que recomienda recetas de cocina que puedan servir al usuario tras una búsqueda inicial por parte de éste. Por otro lado está el recomendador *proactivo*, donde el que lleva la iniciativa es el recomendador, realizando una propuesta inicial al usuario basada en el historial pasado del usuario, en valoraciones asociadas a los elementos, o en cualquier otra estrategia previamente seleccionada (McGinty y Smyth, 2003a).

- Dinámica de recomendación. También aquí distinguimos dos tipos: *single-shot* y *conversacional* (Smyth, 2007). *Single-shot* son aquellos en los que sólo se muestra un conjunto de elementos recomendados al usuario y este tiene la oportunidad de elegir uno o descartarlos. Si la recomendación no agradara al usuario, este debería empezar de nuevo para obtener nuevos elementos. Los recomendadores *conversacionales* son aquellos en los que la recomendación se entiende como un proceso iterativo, en el que el usuario puede ir refinando sus requisitos hasta obtener un elemento adecuado para él. Existen dos estrategias de conversación. *Navegación-por-propuesta* y *navegación-por-pregunta* (Shimazu, 2001, 2002). En la primera, un conjunto de elementos es mostrado al usuario, a partir del cual éste podrá refinar sus requisitos. En la navegación-por-preguntas el sistema recoge los requisitos del usuario a partir de un conjunto de preguntas cuidadosamente seleccionadas.

- Capacidad de personalización. Es decir, si en el proceso de recuperación intervienen o no las características/preferencias/necesidades del usuario (de Gemmis et al., 2011). Los perfiles de usuario pueden contener información sobre el historial de navegación, las preferencias, las necesidades del usuario, o lo que se crea conveniente. La capacidad de personalización está relacionada con cómo se maneja toda esta información, es decir, si es tenida en cuenta o no a la hora de realizar las recomendaciones.

- Determinación de la calidad de los elementos. Existen recomendadores que siguen una aproximación tradicional de la similitud entre elementos y recomendadores que apuestan por innovar en la similitud introduciendo una medida de calidad en los elementos (McGinty y Smyth, 2003b). Esta medida está relacionada con la *diversidad* de los elementos recuperados en la recomendación. Se define la diversidad del conjunto recuperado como la disimilitud existente entre cada par de elementos del conjunto. Para los recomendadores centrados en asegurar una cierta calidad en la recomendación un elemento mejorará su calidad cuanto más disimilar sea a los ya recuperados, siempre y cuando siga manteniendo la similitud con la consulta.

A continuación, miramos más en profundidad los sistemas recomendadores *colaborativos* y los *basados en contenido*.

2.1.1.1. Sistemas recomendadores colaborativos

Los recomendadores *colaborativos* (Ekstrand et al., 2011; Koren y Bell, 2011; Herlocker et al., 2002; Candillier et al., 2007) son aquellos que no necesitan información sobre las características del producto, ya que utilizan en su lugar las valoraciones de otros usuarios a dichos productos. En general, es el proceso de filtrado de información o modelos, que usan técnicas que implican la colaboración entre múltiples agentes, fuentes de datos, etc. El filtrado colaborativo es un método para hacer predicciones automáticas (filtrado) sobre los intereses de un usuario mediante la recopilación de las preferencias o gustos de información de muchos usuarios (colaborador). El mayor inconveniente de esta técnica de recomendación radica en la necesidad de adquirir valoraciones para los elementos de los usuarios. Para resolver este problema, frecuentemente se utilizan las técnicas *Long tail* (Anderson, 2007), que se refieren a la propiedad estadística de que una gran parte de la población de usuarios se mantiene en una cola de distribución de probabilidad como la observada en una distribución de Gauss. Esta técnica ha sido utilizada por Amazon[1] y Netflix[2] entre otras aplicaciones comerciales. El número de valoraciones necesarias se reduce si se cuenta con perfiles de usuario, en cuyo caso el recomendador utilizará esta información para hacer recomendaciones al usuario actual en función de lo que otros usuarios con un perfil similar hayan valorado. Esto permite hacer recomendaciones al usuario actual de elementos que no están en su perfil pero sí en el de usuarios parecidos a él. Los sistemas de filtrado colaborativo se dividen generalmente en dos subgrupos: *basados en memoria* y *basados en modelos* (Lee et al., 2012).

Los sistemas *basados en memoria*, memorizan la matriz de ratings y proponen recomendaciones basadas en la relación entre el usuario a recomendar, un artículo y el resto de la matriz de ratings. Sin embargo, Los *basados en modelos* parametrizan un modelo desde la matriz de ratings y luego realizan las recomendaciones basándose en este modelo. Lo sistemas *basados en memoria* más populares son los *métodos de vecindad*, que predicen ratings utilizando usuarios cuyos ratings se parecen a los del usuario que recibe la recomendación (*recomendación basada en usuarios* (Breese et al., 1998)), o a los productos que se parecen al producto a recomendar (*recomendación basada en productos* (Sarwar et al., 2001)). Se basan, en que si una persona A tiene la misma opinión que una persona B sobre un tema, A es más probable que tenga la misma opinión que B en otro tema diferente que la opinión que tendría una persona elegida al azar. O alternativamente, si dos productos tienen ratings similares entre una pequeña muestra de usuarios, los dos productos también tendrán ratings similares con el resto de la muestra. Los *métodos de vecindad* varían considerablemente dependiendo de la forma en la que se calcula la media ponderada de los ratings. Algunos ejemplos de medidas de similitud que realizan este computo entre ratings son, la *correlación de Pearson* (Herlocker, 2000), el *vector coseno* (Pham et al., 2011) o la *Diferencia-Cuadrática-Media* (DCM) (Herlocker et al., 2002).

Por otro lado, los sistemas *basados en modelos*, ajustan un modelo paramétrico a la muestra de entrenamiento que se usará mas tarde para predecir ratings y realizar recomendaciones. Los métodos *basados en modelos* incluyen los *basados en clusters* (Ungar y Foster, 1998; Connor y Herlocker, 2001), *clasificadores Bayesianos* (Miyahara y Pazzani, 2000) y métodos basados en regresión (Vucetic y Obradovic, 2005). El método *Slope-One* (Li et al., 2011; Lemire y Maclachlan, 2007) ajusta un modelo linear a la matriz de ratings, consiguiendo cálculos rápidos y una precisión razonable. Una clase relativamente reciente y competente de modelos de filtrado colaborativo es la basada en factorización de matrices de bajo rango. El método SVD (Billsus y Pazzani, 1998; Karatzoglou y Weimer, 2010) factoriza la matriz de ratings como un producto de dos matrices de bajo rango (perfil de usuario y perfil de producto) que son usadas para estimar los datos que faltan.

[1] `http://www.amazon.com/`
[2] `http://www.netflix.com/`

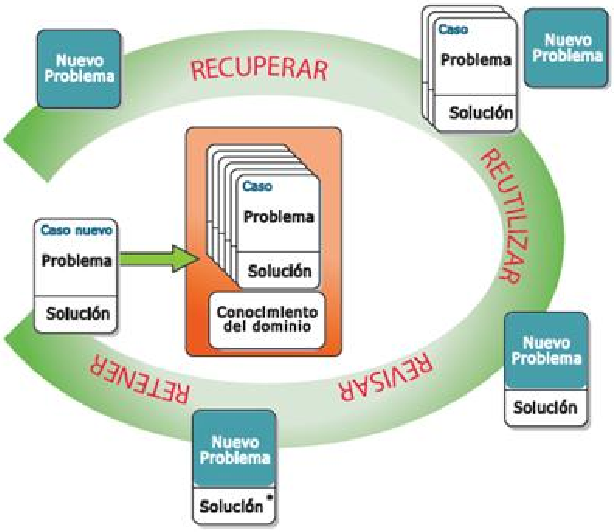


Figura 2.1: Ciclo CBR, Aamodt y Plaza 1994.

Un método alternativo es la Factorización de Matrices No-Negativas (NMF) (Lee y Seung, 2000) que se diferencia del anterior porque fuerza las matrices de bajo rango para que la factorización tenga datos no-negativos. Algunas variaciones recientes son la Factorización de Matrices Probabilística (PMF) (Salakhutdinov y Mnih, 2007), la PMF Bayesiana (Salakhutdinov y Mnih, 2008) o la PMF No-Linear (Lawrence y Urtasun, 2009) entre otras.

En contraposición a los recomendadores *colaborativos*, encontramos los recomendadores *basados en contenidos*, que pasamos a describir a continuación.

2.1.1.2. Sistemas recomendadores basados en contenido

Los sistemas recomendadores *basados en contenido* (Lops et al., 2011) utilizan descripciones de los productos a recomendar y proporcionan un conjunto de soluciones formado por los productos cuyas descripciones más se ajusten a la consulta que realice el usuario. Los recomendadores *basados en casos* son un tipo especial de recomendadores *basados en contenido* en los que cada caso que se recupera representa una recomendación anterior de un producto similar al que el usuario desea. Tienen un origen común con las técnicas de razonamiento basado en casos (CBR, del inglés Case-based Reasoning), veáse Figura 2.1. Los sistemas CBR (Leake, 1996) cuentan con una base de casos que se compone de problemas resueltos anteriores, junto con la solución tomada. De este modo los nuevos problemas se resuelven adaptando soluciones pasadas, usadas para resolver problemas similares.

A continuación, explicamos las diferencias entre los recomendadores individuales y los que se estudian en esta Tesis, los recomendadores grupales.

2.1.2. Sistemas de recomendación grupal

Los sistemas de recomendación se han ocupado tradicionalmente de hacer recomendaciones de elementos a usuarios individuales. Ha sido recientemente cuando se ha empezado a trabajar en el desarrollo de técnicas que permitan proponer recomendaciones a grupos de usuarios simultáneamente (Jameson y Smyth, 2007). Este tipo de sistemas plantea problemas específicos a las técnicas de recomendación, como son la necesidad de adquirir las preferencias del grupo, ayudar al grupo en el proceso de toma de decisiones de cuál es la mejor opción y explicar al grupo las razones de una recomendación. En referencia a los métodos de adquisición de información, la mayoría de los recomendadores de grupos utilizan métodos casi indistinguibles de los que se aplican en los sistemas recomendadores para individuos. Básicamente se pueden dividir en:

- Adquisición de preferencias sin especificación explícita. Muchos sistemas recomendadores para grupos no requieren que sus usuarios especifiquen explícitamente sus preferencias. El sistema puede funcionar con información adquirida implícitamente sobre los usuarios. Un ejemplo de ello es Let's Browse (Lieberman et al., 1999) que se trata de un sistema que recomienda páginas web a grupos de dos o más personas que están navegando juntas en la red.

- Especificación de preferencias explícita. Otros tipos de recomendadores para grupos, sí que requieren una especificación explícita de las preferencias de los usuarios. Por ejemplo en Travel Decisiom Forum (Jameson et al., 2004) que se trata de un sistema que ayuda a un grupo de usuarios a acordar unas determinadas características de unas vacaciones que planeen juntos. Otro sistema recomendador menos explícito sobre la especificación de preferencias es PolyLens (O'Connor et al., 2001) se trata de un sistema que recomienda películas a grupos de usuarios. Es una extensión del sistema MovieLens (Schafer et al., 2007a), que se basa en técnicas de filtrado colaborativo, donde los usuarios no describen explícitamente sus preferencias en el dominio de las películas, sino que puntúan las películas individualmente en una escala de 1 a 5 estrellas. Este sistema hace recomendaciones a grupos agregando las preferencias de los miembros del grupo con la estrategia de minimizar el malestar, basándose en la hipótesis de que la felicidad del grupo será igual a la del menos feliz de sus miembros.

Una vez que el sistema ha adquirido el conocimiento necesario sobre los gustos de los usuarios, surgen diferentes tipos de sistemas de recomendación grupal en función de cómo se decida gestionar estas preferencias y en función de los objetivos y características particulares de cada sistema. Ejemplos de diferencia de objetivos a la hora de gestionar las preferencias de los usuarios son: los sistemas que se centran en las preferencias negativas, como en Adaptive Radio (Chao et al., 2005), o: los sistemas que comparten información sobre las preferencias especificadas, como podemos ver en Travel Decision Forum (Jameson et al., 2004) o en CATS (McCarthy et al., 2006). Los primeros sólo tienen sentido si el procedimiento que se está utilizando para generar las recomendaciones está diseñado principalmente para evitar que el producto seleccionado sea especialmente contrario a los gustos de cualquier miembro del grupo. En cuanto al segundo tipo de sistemas, se puede dar el caso de que en un sistema recomendador para grupos interese que cada miembro pueda conocer las preferencias de los otros miembros del grupo. Por ejemplo para aprender de otros miembros del grupo, para ahorrar tiempo a la hora de especificar sus propias preferencias, para asimilar los motivos de los demás componentes y así llegar a un consenso más fácilmente o para poder prever actitudes y comportamientos de los otros componentes del grupo. Fácilmente podemos ver cómo de este enfoque surge un problema de manipulación, por ejemplo si una persona no quiere que salga un producto en particular, podría calificarlo como "odiado" y asegurarse así que nunca salga recomendado. Esta situación ocurría en MusicFX según describen sus autores en (McCarthy y Anagnost, 1998).

Dependiendo del tamaño y la homogeneidad del grupo, puede ser difícil encontrar una recomendación que sea adecuada para cada miembro del grupo de manera individual (Jameson, 2004). En la mayoría de los casos, el recomendador debe escoger aquella opción que satisfaga al mayor número de usuarios del grupo, de acuerdo con sus preferencias individuales. Por tanto, se necesita algún tipo de método de agregación para combinar la información sobre las preferencias individuales de los usuarios de forma que el sistema pueda obtener la recomendación idónea para el grupo en sí. Existen tres aproximaciones básicas para resolver este problema: (a) mezclar las recomendaciones que se harían por separado a cada uno de los miembros del grupo. Es un método simple de agregación donde se seleccionan los productos más prometedores para cada miembro del grupo y se combinan utilizando la unión o la intersección en una única lista. Este método fue considerado por PolyLens (O'Connor et al., 2001); (b) agregar las valoraciones para cada usuario. Para cada producto candidato y por cada miembro del grupo, el sistema predice cómo ese componente evaluaría dicho producto, luego agrega todas predicciones de un mismo producto en un solo rating y devuelve una colección de candidatos que tengan las valoraciones grupales previstas más altas. Este método es el más popular a la hora de diseñar recomendadores grupales y por consiguiente el que hemos adoptado en nuestro trabajo. En la siguiente subsección revisaremos diferentes funciones de agregación utilizadas y varios sistemas que adoptan este método. Un ejemplo de este comportamiento se puede encontrar en el Pocket RestaurantFinder (McCarthy, 2002); y (c) construir un modelo de las preferencias del grupo. En este enfoque el sistema usa la información sobre las preferencias individuales de los componentes del grupo para diseñar un modelo que prediga directamente las preferencias del grupo como conjunto en sí. Let's Browse (Lieberman et al., 1999) es un ejemplo de este tipo de sistemas.

2.1.2.1. Sistemas recomendadores para grupos que utilizan agregación de preferencias

Una vez se ha elegido un enfoque general (mezcla de recomendaciones, agregación de preferencias o construcción de un modelo), si éste es la agregación de preferencias hay que decidir qué procedimiento computacional se va a usar para la agregación. Hay varios objetivos que deben tenerse en cuenta a la hora de elegir uno, como la satisfacción total, la comprensibilidad o el grado de igualdad, estos variarán dependiendo de la situación dada. A continuación vemos algunos ejemplos, nótese que en ellos $\hat{r}_{u,i}$ es la predicción del rating para cada usuario $\{u : 1 \ldots n\}$ perteneciente al grupo G_a, para un producto i y $\hat{r}_{G_a,i}$ es el rating final que obtiene el producto i para un grupo activo G_a:

- *Maximizar la satisfacción media.* En esta función de agregación se computa una media de la satisfacción predicha para cada miembro del grupo y se usa como base de la selección de productos candidatos. La función a utilizar sería:

$$\hat{r}_{G_a,i} = 1/n * \sum_{u=1}^{n} \hat{r}_{u,i} \tag{2.1}$$

- *Minimizar la miseria.* Aunque la satisfacción media sea alta, si una solución deja a un componente del grupo especialmente disconforme, se puede considerar como una situación indeseada. En PolyLens (O'Connor et al., 2001) la minimización de la miseria es el único criterio que se aplica. Es posible tener en cuenta este factor como una restricción que debe cumplir cada solución, por ejemplo, considerando que el rating más bajo debe ser siempre superior a un cierto umbral. La función básica a utilizar sería:

$$\hat{r}_{G_a,i} = \min_{u \in G_a} \hat{r}_{u,i} \tag{2.2}$$

- *Minimizar la penalización*, asegurar un grado de justicia. En esta función se considera una solución que satisfaga a cada persona de igual modo. Esta situación es en general preferible a una solución que satisfaga a un cierto número de componentes a expensas de otros. La función a utilizar sería:

$$\hat{r}_{G_a,i} = 1/n * \sum_{u=1}^{n} \hat{r}_{u,i} - \omega * desviacionEstandar(\hat{r}_{u,i}) \qquad (2.3)$$

Donde ω es el peso que refleja la importancia del grado de justicia.

Estas estrategias de agregación han sido criticadas (Chen et al., 2008; Masthoff y Gatt, 2006) porque combinan las valoraciones de los usuarios siempre de la misma forma, sin tener en cuenta las interacciones concretas entre los miembros de un grupo determinado. Nuestra aproximación es una propuesta para resolver este problema. Otras soluciones a este problema son las que se proponen en (Chen et al., 2008), donde los autores usan algoritmos genéticos para determinar el peso óptimo que cada rating individual debe tener en la predicción de la valoración del grupo. Esta solución tiene el inconveniente de que se ha de disponer de valoraciones previas del grupo para otros productos. O las presentadas en (Masthoff y Gatt, 2006), donde se refleja la tendencia de algunos trabajos recientes a incluir consideraciones sociales en la relación de los miembros del grupo a la hora de hacer las recomendaciones. En este trabajo, se utiliza la idea de combinar la satisfacción individual con el *contagio emocional* aplicada a la recomendación de videoclips a un grupo de usuarios. Se considera que la elección del mejor video a escuchar a continuación viene determinada en parte por el último video seleccionado anteriormente. Esta evolución se computa como una función de satisfacción que tiene en cuenta el estado afectivo del individuo. El estado de un individuo influye a su vez en el estado afectivo de los demás miembros del grupo, produciendo así el contagio afectivo que se toma en consideración durante el proceso de recomendación.

Otro factor que proporciona una variación en la agregación de preferencias es la inclusión de distintos perfiles que reflejan la personalidad de cada usuario involucrado en el proceso de recomendación (Recio-García et al., 2009). En este trabajo, precursor de esta Tesis, se propone realizar recomendaciones a grupos, teniendo en cuenta las diferentes personalidades de los miembros de un grupo. En la recomendación final, las preferencias de cada individuo tienen diferentes pesos dependiendo de la manera en que cada miembro del grupo reaccionaría en una situación de conflicto.

Una vez el sistema realiza la recomendación es natural pensar que los componentes del grupo a recomendar deseen saber en cierto modo cómo se llegó a la recomendación, y en particular cuan de atractiva es dicha recomendación para cada uno de ellos como individuos. Por ello muchos sistemas recomendadores acompañan cada solución con una explicación de la recomendación (Herlocker et al., 2000; Tintarev, 2007). Un ejemplo de sistema que utiliza explicaciones para justificar las soluciones propuestas es Let's Browse

(Lieberman et al., 1999). Las explicaciones en los sistemas de recomendación presentan múltiples variaciones pudiendo ir desde un simple índice de la confianza del sistema a una visualización compleja de los pros y contras de una solución. Sin embargo, no existe ninguna garantía de que se vayan a tomar ninguna de las recomendaciones realizadas, no importa cuán de apropiada o convincente sean las recomendaciones o la explicación del sistema.

Con las recomendaciones a grupos se prevén debates extensivos y negociaciones. Para ello algunos sistemas tienden a no proporcionar una única decisión final, sino un medio para argumentar entre los miembros del grupo y llegar a un consenso como se explica en (Jameson et al., 2004).

Resumiendo, podemos concluir que existe una necesidad de adaptar los procesos de recomendación a la composición del grupo (Jameson y Smyth, 2007; Masthoff, 2004). Esta creencia está respaldada por algunos trabajos recientes que han centrado sus estudios en analizar la eficacia de las recomendaciones grupales de acuerdo a diferentes aspectos, como el tamaño del grupo y la similaridad entre miembros del grupo (Baltrunas et al., 2010), o en estudiar diferentes distribuciones de relevancia/peso de las preferencias de los miembros del grupo de acuerdo a su comportamiento o rol dentro del grupo (Berkovsky y Freyne, 2010). Nuestra propuesta redunda en la creencia general de adaptar y mejorar las recomendaciones grupales mediante la inclusión de factores que añadan información adicional sobre el grupo en sí, la composición grupal y los miembros del grupo. Más concretamente, hemos basado nuestra investigación en el estudio de las redes sociales, los factores sociales y modelos de confianza que pueden ser inferidos de ellas y cómo estos factores pueden favorecer a la mejora de los recomendadores grupales. La importancia de incluir información social a los sistemas de recomendación es una reciente teoría que está empezando a ser aceptada en la comunidad de investigación sobre recomendadores y que ha dado lugar a un nuevo tipo de recomendadores: los recomendadores sociales.

2.1.2.2. Medidas de evaluación en sistemas de recomendación grupal

Los sistemas de recomendación grupal requieren de funciones de evaluación que midan la precisión de las recomendaciones proporcionadas. Este es un factor clave para poder validar los resultados producidos. Para ello, un objetivo común es comparar los resultados obtenidos por el sistema de recomendación grupal $rec(G_a)$ con las preferencias/decisiones reales del grupo $fav(G_a)$ (Quijano-Sánchez et al., 2014a). Para poder elegir una función de evaluación adecuada se deben tener varios factores en cuenta:

El primero es la limitación de la longitud de la lista $fav(G_a)$. Los grupos reales de personas sólo están interesados en unos pocos productos que realmente quieren consumir y consecuentemente necesitamos limitar la lista de productos a recomendar (Shi et al., 2012). Por tanto, no es conveniente usar medidas generales como precisión o exahustividad (Billsus y Pazzani, 1999; Maybury et al., 2004). El segundo factor a tener en cuenta es si $rec(G_a)$ en el dominio estudiado (películas, ropa, etc) es una colección ordenada o no. Por ejemplo, en nuestro caso particular nos centraremos en recomendadores que propongan una colección de k productos sin ninguna clase de ordenación -estos son más tarde evaluados por los miembros del grupo para tomar una decisión final-. Esta característica descarta varias medidas de evaluación que comparan el orden de los resultados y la validación de listas como el Error Medio Absoluto (MAE) (Herlocker et al., 2004; Adomavicius y Tuzhilin, 2005) o el Normalized Discounted Cumulative Gain (nDCG) (Baltrunas et al., 2010). En este caso, donde no se consideran listas ordenadas, hay algunas métricas en el campo de Extracción de Información (IE) (Tomlinson, 2006) que son adecuadas.

Por ejemplo la *precision@n* evalúa cuantos elementos en $rec(G_a)$ están en $fav(G_a)$. Esta clase de evaluación se puede ver desde un punto de vista distinto: normalmente estaremos interesados en tener al menos uno de los productos de $rec(G_a)$ en la lista $fav(G_a)$. Esta medida se llama *acierto@n* y devuelve un 1 si al menos tenemos un acierto en las primeras n posiciones (Quijano-Sánchez et al., 2013c). Por tanto, se puede usar *acierto@n* para evaluar sistemas de recomendación grupal mediante el cálculo del porcentaje de recomendaciones donde existe al menos un acierto en $fav(G_a)$. Por ejemplo, una precisión del 90 % utilizando *acierto@n* representa que el recomendador ha sugerido al menos un producto correcto para el 90 % de los grupos evaluados. De hecho, *acierto@n* es equivalente a tener $precision@n > 1/n$. También podría definirse una métrica de 2*acierto@n* (equivalente a $precision@n > 2/n$), que representa cuantas veces $fav(G_a)$ contiene al menos dos productos de $rec(G_a)$. Obviamente, esta medida es mucho más restrictiva.

A continuación revisamos la creciente importancia de las redes sociales, la información útil para la comunidad de recomendadores que se puede extraer de ellas y algunos de los trabajos en desarrollo en el area de la recomendación social.

2.2. Factores sociales en los sistemas de recomendación

Las redes sociales online, como por ejemplo Facebook, proporcionan una fuente muy rica de información de la que podemos hacer uso para recomendar una variedad enorme de productos, ya sean artículos de noticias, películas, libros, o un largo etcetera. Mientras que los sistemas recomendadores han sido extensamente investigados desde mediados de los 90 (McCarthy y Anagnost, 1998; Lieberman et al., 1999), el estudio de los sistemas basados en recomendación social es un área totalmente nueva (Jiliang Tang y Liu, 2013). Uno de los factores clave que utilizan los recomendadores basados en aspectos sociales es el estudio de las múltiples dimensiones que existen dentro de las redes sociales de cada usuario. Entre estas dimensiones, destacan la confianza social, sus intereses, y la similitud entre usuarios. En (Gartrell et al., 2010), sus autores recogen estas dimensiones con el objetivo de desarrollar una nueva clase de sistemas recomendadores que recoja todas estas dimensiones. Otro ejemplo de recomendadores sociales son aquellos que estudian las relaciones sociales; existen varios enfoques que proponen sistemas de recomendación social basados en agrupaciones de confianza (Ma et al., 2009), propagación de confianza (Jamali y Ester, 2009), o directamente recomendadores basados en la confianza entre usuarios (Zhang et al., 2009).

Los recientes trabajos basados en sistemas de recomendación social han reportado una mejora significativa en los resultados de las recomendaciones (Golbeck, 2006b; Nepal et al., 2012; Pera et al., 2010; Yang et al., 2012; Bao et al., 2012; Hu et al., 2012). Por otra parte, hemos de reflejar que también ha habido intentos fallidos de utilizar recomendaciones sociales (IBM, 2012; Quora, 2012). Como veremos a continuación, aunque paralelamente a nuestra investigación se están realizando trabajos que se centran en la investigación de recomendadores sociales, la mayoría de ellos utiliza estas herramientas sociales para el caso concreto de las recomendaciones a individuos, sin explotar los aspectos sociales de las relaciones grupales y su aplicabilidad en las recomendaciones grupales. Existen trabajos que sí se centran en recomendaciones grupales sociales, como por ejemplo (Gartrell et al., 2010). En este trabajo, se presenta un modelo teórico donde utilizan información social como es la intensidad de la relación o el grado de experiencia de cada usuarios a la hora de realizar una recomendación grupal. Sin embargo se encuentran limitados por la necesidad de obtener esta información por medio de largos cuestionarios (la mayoría presenciales y guiados). Esto no es efectivo a la hora de proponer una herramienta que sea utilizable y de provecho para grupos de usuarios. Es por ello que en esta Tesis hemos decidido dar una paso más en la investigación de sistemas recomendadores y utilizar la información que hoy en día se encuentra en las redes sociales para inferir diferentes factores sociales (en nuestro caso la personalidad y la confianza principalmente) y utilizarlos junto con las técnicas ya existentes de recomendación grupal para diseñar una aplicación en un red social que realiza recomendaciones a grupos de usuarios automáticamente a través de la creación de eventos. De este modo ofrecemos un servicio fácilmente accesible y dinámico (a nivel de interacción con el usuario para obtener datos necesarios).

A continuación, expondremos una breve introducción a las redes sociales existentes y como son un entorno perfecto para generar modelos de confianza, cuyo uso en los sistemas de recomendación ha sido uno de los precursores de los sistemas de recomendación social.

2.2.1. Redes sociales

Los medios de comunicación sociales o simplemente medios sociales (social media en inglés), son plataformas de comunicación en línea donde el contenido es creado por los propios usuarios mediante el uso de las tecnologías de la Web 2.0, que facilitan la edición, la publicación y el intercambio de información. Estos medios hacen referencia a interacciones entre usuarios a través de las cuales crean, comparten, y/o intercambian información e ideas dentro de comunidades virtuales o redes. Genéricamente podríamos llamar a esta nueva web que presenta redes sociales y alberga información sociales la web Social.

El servicio que ofrecen las redes sociales se centra en construir y reflejar las relaciones sociales entre personas que por ejemplo comparten intereses comunes y/o actividades. Esencialmente consisten en la representación de cada usuario, a menudo mediante un perfil, sus conexiones sociales y una variedad de servicios adicionales.

Desde el punto de vista de nuestra investigación, las redes sociales proporcionan una medida de confianza entre los diversos usuarios que forman parte de ellas y nos aportan toda una estructura de red construida entre ellos. Un enlace entre dos usuarios de la red social simboliza una afinidad entre estos dentro de la temática de la red. Redes sociales como Facebook[3], LinkedIn[4], Instagram[5], Twitter[6] o Lastfm[7] (entre otras muchas), tienen como objetivo el intercambio de información entre sus usuarios. La temática de las redes sociales es amplia, variando desde la laboral al intercambio de fotos y música. La expansión de las redes sociales tiene su auge en los últimos años coincidiendo con la expansión de Internet, donde han cobrado una gran importancia[8]. Los usuarios de las redes sociales buscan un lugar donde encontrar gente similar a ellos dentro de la temática de la red y con quien poder compartir sus ideas. Existen dos clasificaciones principales de redes sociales:

- **Por su público objetivo y temática**:
    - *Redes sociales Horizontales*: Son aquellas dirigidas a todo tipo de usuarios y sin una temática definida. Los ejemplos más representativos son Facebook y Twitter.

[3] http://www.facebook.com
[4] http://www.linkedin.com
[5] http://instagram.com
[6] https://twitter.com
[7] http://www.last.fm/
[8] Existen mas de 200 redes sociales en la actualidad (véase http://en.wikipedia.org/wiki/List_of_social_networking_websites) y casi 2 billones de usuarios (véase http://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/).

    - *Redes sociales Verticales*: Están concebidas sobre la base de un eje temático agregador. Su objetivo es el de congregar en torno a una temática definida a un colectivo concreto. Dentro de este tipo pueden ser *profesionales* como LinkedIn, o de *ocio* como Lastfm.

- **Por el sujeto principal de la relación**:

    - *Redes sociales Humanas*: Son aquellas que centran su atención en fomentar las relaciones entre personas uniendo individuos según su perfil social y en función de sus gustos, aficiones, lugares de trabajo, viajes y actividades. Estas redes son de un carácter más íntimo, lo que permite inferir modelos de confianza y contagio emocional. Ejemplos de este tipo de redes son Facebook, Tuenti o Google+.
    - *Redes sociales de Contenidos*: Las relaciones se desarrollan uniendo perfiles a través de contenido publicado, los objetos que posee el usuario o los archivos que se encuentran en su ordenador. Los ejemplos más significativos son Twitter o Flickr.

La investigación multidisciplinar ha mostrado que las redes sociales operan en muchos niveles. En su forma más simple, una red social es un mapa de todos los lazos relevantes entre todos los nodos estudiados. Se habla en este caso de redes "sociocéntricas" o "completas". Otra opción es identificar la red que envuelve a una persona (en los diferentes contextos sociales en los que interactúa); en este caso se habla de "red personal". El análisis de redes sociales se está utilizando para una enorme cantidad de temas donde se ha aprovechado la información contenida en las redes sociales para construir sistemas con distintos propósitos:

- La evolución de las redes sociales a veces puede ser simulada mediante el uso de modelos basados en agentes que proporcionan información sobre la interacción entre las normas de comunicación, propagación de rumores y la estructura social (Yahja y Carley, 2005).

- La teoría de Difusión de innovaciones (Rogers, 1983) explora las redes sociales y su rol en la influencia de la difusión de nuevas ideas. La idea es entendida como una explicación acerca de como una innovación es comunicada a través de ciertos canales, a través del tiempo, entre los miembros de un sistema social y cómo esta "nueva idea" es aceptada y divulgada entre sus miembros de la red social.

- Un estudio ha descubierto que la felicidad tiende a correlacionarse en redes sociales (Fowler y Christakis, 2008). Cuando una persona es feliz, los amigos cercanos tienen una probabilidad un 25 por ciento mayor de ser también felices. Además, las personas en el centro de una red social tienden a ser más felices en el futuro que aquellos situados en la periferia.

- En McDonald (2003) se muestran diferentes sistemas que han utilizado las redes sociales como mecanismo para recomendar personas con las que colaborar o cómo utilizar visualizaciones de redes sociales para poder aprovechar las diferentes colaboraciones que surgen entre usuarios de un mismo entorno laboral.

A continuación explicaremos uno de los muchos usos que se le puede dar a la información almacenada en las redes sociales, la creación de modelos de confianza.

2.2.2. Modelos de confianza

Las investigaciones actuales indican que las personas tienden a confiar más en recomendaciones que provienen de gente en la que confían (amigos) que en recomendaciones basadas en valoraciones anónimas (Sinha y Swearingen, 2001). Este elemento social es incluso más importante cuando se realizan recomendaciones grupales cuyo propósito es ayudar a los usuarios a elegir un producto para consumir en grupo. Cuando en el día a día nos encontramos con esta clase de toma de decisiones vemos que generalmente siguen un proceso de argumentación donde cada usuario defiende sus preferencias y rebate las opiniones de otros. Aquí, la confianza entre usuarios es un factor clave ya que juega un papel fundamental a la hora de ceder y aceptar propuestas para alcanzar una decisión consensuada. Por ello, cuando modelamos estos procesos de toma de decisiones en los sistemas de recomendación grupal sería lógico que este factor de confianza se tuviese en cuenta.

Existe una gran cantidad de trabajos que se centran en la generación de modelos de confianza. Estos estudios se han visto incrementados aun más con el alzamiento de la web colaborativa actual (Web 2.0), que ha estimulado la idea de una web de confianza (WOT, Web of Trust) (Golbeck, 2006b; O'Donovan y Smyth, 2005; Victor et al., 2008). La WOT representa la confianza entre usuarios, que se ve modelada en una red online. Hay algunos enfoques específicos que utilizan una red social personalizada para recomendar elementos. Un ejemplo de esto es FilmTrust (Golbeck, 2006b) que utiliza este tipo de redes de confianza entre usuarios en el ámbito de las preferencias cinematográficas. El trabajo en (Golbeck, 2006a) presenta un estudio sobre cómo inferir las relaciones de confianza en redes sociales. El reto de calcular la confianza es ser capaces de determinar cuánto confía una determinada persona en la red social en otra. Ciertamente, las inferencias sobre la confianza no serán tan precisas como una valoración directa. En este trabajo, Golbeck presenta un algoritmo para inferir la confianza de las redes con sistemas de valoración continuos llamado TidalTrust, que mejora la precision de otros algoritmos que calculan la confianza en un 10 %. Sin embargo algunas de estas redes de confianza son bastante difíciles de obtener, pues requieren preguntas directas a los usuarios para crear la red y esto puede ocasionar rechazo.

Otro enfoque prometedor, y el que se sigue en esta Tesis es la recolección de información relativa a la confianza en redes sociales existentes como Facebook o Twitter. Estas redes contienen información implícita que se podría usar para mejorar el proceso de recomendación. Esta opción tiene como principal ventaja el ser completamente transparente a los usuarios. Los usuarios no necesitan proporcionar información implícita sobre su confianza con otros usuarios, ya que este conocimiento se extrae implícitamente de la interacción diaria en la red social. Sin embargo este método tiene un claro inconveniente y es que cada usuario del grupo a recomendar tiene que pertenecer a una red social. No obstante, la creciente popularidad de esta clase de aplicaciones web minimiza este riesgo. Es más, está siendo cada vez más común la creación de eventos (como ir al cine) a través de redes sociales, luego, se podrían incluir en este tipo de sitios web técnicas de recomendación para grupos, idea que se implementa en esta Tesis.

2.2.3. Recomendadores sociales

A continuación presentamos algunos ejemplos de recomendadores que utilizan información social en sus procesos de recomendación. En las redes sociales, los perfiles de usuario capturan los "propósitos" de los usuarios. Varios sistemas de recomendación han sido capaces de ayudar a los usuarios en procesos de toma de decisiones mediante el uso de este tipo de información social (Jiliang Tang y Liu, 2013). Estos recomendadores sociales varían en sus tipos de recomendaciones enormemente, por ejemplo existen recomendadores que guían a los usuarios en la tarea de seleccionar en qué gente deben confiar por medio de la utilización de la información almacenada en Twitter (Tavakolifard et al., 2013) o también existen recomendadores que sugieren a una empresa qué personas debe contratar utilizando el grafo social de LinkedIn (Posse, 2012). Otro ejemplo es (Konstas et al., 2009), en este trabajo utilizan la información almacenada en la red social musical *last.fm*, para capturar vínculos de amistad y etiquetas sociales que les permiten mejorar la precisión de las recomendaciones.

Hoy en día las redes sociales tienen publicados perfiles de usuario compuestos por diferentes atributos, como son la localización geográfica, intereses, etc. Es esta descripción informativa la que usan los sistemas recomendadores como base a la hora de agrupar usuarios, compartir contenido o sugerir a otros usuarios con los que interactuar. Sin embargo, en la práctica estos perfiles están incompletos y no todos los usuarios proporcionan información en todos los campos posibles, por tanto hay atributos que quedan vacíos. En (Mislove et al., 2010) proponen utilizar la información almacenada en diferentes redes sociales para inferir los atributos que faltan en los perfiles de usuario.

Otros enfoques usan el concepto de la web social semántica que engloba todos los procesos en los que las interacciones sociales en la web conllevan a la creación de representaciones del conocimiento explícitas y semánticamente muy ricas. Podemos ver la web social semántica como una web de sistemas de conocimiento colectivo capaces de proporcionar información útil basándose en las interacciones entre personas, una información, que es más rica cuantas más personas participan en ella. Tradicionalmente, el análisis y estudio de el conocimiento existente en la web social ha estado basado principalmente en procesos de extracción manual (Golbeck, 2006b; O'Donovan y Smyth, 2005; Victor et al., 2008). Este tipo de análisis, que básicamente es un barrido de información, require el uso de sistemas inteligentes para filtrar la inagotable información que se puede obtener de esta web social. Sin embargo, estudios recientes han desarrollado recomendadores que obtienen esta clase de información automáticamente (Nachawati et al., 2012).

Todos estos estudios utilizan algunos de los factores que se proponen en esta Tesis, como son el uso de factores sociales (Konstas et al., 2009; Tavakolifard et al., 2013), más concretamente el uso de factores sociales en recomendadores grupales (Gartrell et al., 2010), el estudio de la confianza entre usuarios en sistemas de recomendación (Ma et al., 2009; Jamali y Ester, 2009; Zhang et al., 2009), extracción automática de conocimiento almacenado en las redes sociales (Nachawati et al., 2012), etc. Sin embargo, no hemos encontrado ningún trabajo que integre, evalúe y extraiga automáticamente de las redes sociales, factores sociales en sistemas de recomendación grupal. A lo largo de esta Tesis estudiaremos cómo utilizar la información y los recursos que ofrecen las redes sociales para calcular nuestros factores sociales, cómo usarlos para modelar la composición de un grupo y cómo utilizarlos para mejorar los sistemas de recomendación grupal, pues demostraremos que por medio de ellos podremos generar un sistema que simule más fielmente las argumentaciones que tienen lugar en los grupos en la realidad y por tanto que genere recomendaciones más parecidas a las decisiones que los grupos tomarían finalmente.

En resumen, los sistemas de recomendaciones social están aún en una etapa temprana de desarrollo, y existen muchas cuestiones y puntos clave que suponen un desafío y que requieren investigación. Es por eso que consideramos necesario debatir y plantear nuevos enfoques que pueden mejorar el potencial de los sistemas de recomendación grupal y hacer de la recomendación social un área aplicable a un rango aún mayor de aplicaciones.

Tras el estudio del estado del arte realizado a lo largo de este capítulo se planteó la primera hipótesis de esta Tesis: ***"H1: Existe la necesidad de mejorar los sistemas de recomendación grupal por medio de un modelado más detallado de los procesos de toma de decisiones, posiblemente mediante la inclusión de factores sociales"***. En el siguiente capítulo demostraremos dicha hipótesis y plantearemos algunas ideas iniciales para nuestro *Modelo de Recomendación Social* (MRS).

Capítulo 3

Estudio de la obtención y uso de los factores sociales en los procesos de recomendación grupal para facilitar la toma de decisiones en grupo

3.1. Vision general

Los sistemas de recomendación se están convirtiendo en unas de las tecnologías más habituales a la hora de ayudar a usuarios en la labor de encontrar productos de su interés en la web (Mika, 2011). Existe un amplio espectro de productos como libros, música, juegos, viajes, etc que son difíciles de descubrir en la web debido a la abrumadora cantidad de información disponible. Los sistemas recomendadores (Jameson y Smyth, 2007) facilitan la labor de los usuarios a la hora de encontrar artículos y proporcionan una experiencia más rica e interactiva que las clásicas interfaces basadas en catálogos de productos.

Inicialmente, los sistemas recomendadores estaban enfocados únicamente a usuarios individuales (Ekstrand et al., 2011; Lops et al., 2011). Sin embargo, con el nacimiento de la web colaborativa (conocida como Web 2.0) se ha promocionado la organización de diferentes actividades realizadas en grupo en la web, como ir a ver una película, ir a un restaurante, escuchar una estación de radio o viajar con amigos. Un claro ejemplo de esto son los eventos organizados a través de redes sociales como Facebook. En estas situaciones los sistemas recomendadores juegan un papel muy significativo ya que conseguir que un grupo de usuarios llegue a un mutuo acuerdo sobre una actividad dada no es una tarea trivial (incluso cuando se realiza a través de argumentaciones presenciales entre personas). Para facilitar esta tarea, el número de sistemas recomendadores que se encargan del desafío que supone realizar recomendaciones grupales ha aumentado considerablemente (O'Connor et al., 2001; McCarthy, 2002). Estos recomendadores, como por ejemplo LET'S BROWSE o FlyTrap (Lieberman et al., 1999; Crossen et al., 2002), están basados generalmente en la simple agregación de las preferencias individuales de cada miembro del grupo. Sin embargo, nosotros consideramos que una recomendación grupal no es una mera agregación de preferencias. Afirmación que en los últimos años han hecho otros investigadores en sistemas de recomendación (Jameson y Smyth, 2007; Masthoff, 2004), confirmando que existe una creencia generalizada sobre la necesidad de adaptar los procesos de recomendación a la composición del grupo.

Las personas son individuos sociales y, por tanto, existen elementos sociales que tienen un gran impacto en los procesos de toma de decisiones (Rutherford y Ahlgren, 1991; Goode, 2000; Wanga et al., 2006). Nuestra propuesta tiene en cuenta este hecho y sugiere que la satisfacción general del grupo no es siempre la agregación de las preferencias de sus miembros. La influencia que los grupos tienen sobre las decisiones de un individuo es obvia. Nos referimos comúnmente a esta situación como *contagio emocional*: el efecto de los estados afectivos que los individuos tienen sobre otras personas en un grupo (Masthoff y Gatt, 2006; Barsade, 2002; Hatfield et al., 1994). Este *contagio* es generalmente proporcional a la *fuerza del vínculo* o la *confianza* entre individuos, dado que los amigos más cercanos tienen más influencia que otros no tan cercanos (Victor et al., 2008; Golbeck, 2006a; O'Donovan y Smyth, 2005). Además, el principio de *homofilia*, expone que las personas tienden a confiar y a ser amigos con aquellas personas con las que comparten intereses (Burt, 1982; Miller Mcpherson y Cook, 2001; McPherson y Smith-Lovin, 1987). Sin embargo, la influencia de un individuo dentro de un grupo también depende del grado de *conformidad* de cada individuo (Masthoff y Gatt, 2006). Se ha demostrado que las personas modifican sus opiniones para adaptarse a aquello que opina la mayoría (Masthoff y Gatt, 2006). El grado de conformidad está contrarrestado con el comportamiento que tiene cada individuo en situaciones conflictivas. En estas situaciones, la *personalidad* influye en la aceptación de las propuestas de otras personas (Recio-García et al., 2009). Además, se ha demostrado que las decisiones de un individuo cambian cuando éste recibe cierto tipo de argumentos en contra y que se mantienen en caso contrario. Esta reacción individual ante la *persuasión* es una parte esencial a la hora de entender la agregación de preferencias en grupos de personas y puede explicar la ventaja de estudiar el comportamiento grupal como una entidad propia en contraposición al estudio del comportamiento individual de cada persona en un grupo en los procesos de toma de decisiones (El-Shinnawy y Vinze, 1998; Penczynski, 2014). Finalmente, se debe tener en cuenta el concepto de *justicia* (Liang et al., 2007; Masthoff y

Gatt, 2006) a largo plazo. Esto implica la existencia de un equilibrio entre los productos recomendados y la satisfacción de los usuarios a los que se les proporcionó recomendaciones menos afines a sus preferencias por el bien del grupo.

La mayoría de los estudios sobre recomendaciones grupales consideran las preferencias de cada miembro del grupo con el mismo grado de importancia e intentan satisfacer las preferencias de cada individuo por igual (Jameson y Smyth, 2007; Baltrunas et al., 2010; Berkovsky y Freyne, 2010; Pessemier et al., 2013). Sin embargo, nosotros creemos que todos estos factores sociales (contagio emocional, influencia, personalidad, ...) deben incluirse en los modelos de recomendación, para así poder representar fielmente el comportamiento grupal que siguen las personas a la hora de elegir un producto. Aunque el proceso de modelar esta clase de conocimiento social puede parecer natural, existe una gran limitación en este proceso: los factores sociales son muy difíciles de estimar. Hasta ahora era casi imposible obtener estos factores sin tener que molestar a los usuarios con métodos intrusivos como cuestionarios. Sin embargo, hoy en día, la web colaborativa proporciona una herramienta que se puede utilizar para aliviar este problema: la redes sociales. Las redes sociales permiten a los usuarios interactuar y desarrollar relaciones sociales en un entorno basado en ordenadores. De hecho, varios trabajos han señalado que los factores sociales se pueden inferir a través de éstas (Golbeck, 2006b; Bischoff, 2010). Por ejemplo, se puede estimar el vínculo entre usuarios observando el número de mensajes que se intercambian o el número de amigos en común.

Tras este estudio inicial podemos demostrar nuestra hipótesis H1: ***"Existe la necesidad de mejorar los sistemas de recomendación grupal por medio de un modelado más detallado de los procesos de toma de decisiones, posiblemente mediante la inclusión de factores sociales"***. En lo restante de este capítulo explicamos cómo inferir este tipo de factores sociales directamente de las redes sociales. Nuestro objetivo es utilizar este conocimiento social para modelar los procesos de toma de decisiones en grupo y así proporcionar recomendaciones que se ajusten mejor a la realidad. Para ello, previamente al proceso de recomendación, se analizarán dos factores sociales principales (aunque como veremos más adelante en ocasiones se tendrán en cuenta otros adicionales): la *personalidad* de cada individuo, es decir, la forma en la que los usuarios reaccionarán ante una situación de conflicto en el proceso de decisión de un producto a consumir, si son abiertos de mente o no, colaboran en la toma de la decisiones, etc. Y la *confianza* entre los componentes del grupo, quién se fiará de la opinión de quién, quién influye a quién, etc.

3.2. Identificación y estudio del comportamiento, respecto a la resolución de conflictos, de las personas en un grupo en función de su personalidad

Cuando nos enfrentamos a una situación en la que las inquietudes de las usuarios son incompatibles surgen *situaciones conflictivas*. Aquí, conflicto se entiende como las diferencias que impiden un acuerdo. Más concretamente, para interacciones grupales, se define como acciones opuestas y competitivas que son incompatibles: acción o estado antagónico (en cuanto a divergencia de ideas, intereses o personas) (Mer, 2002). En situaciones de conflicto las personas tienen diferentes expectaciones y comportamientos (Masthoff y Gatt, 2006) y éstas se deben de tener en cuenta. Cuando empezamos nuestra investigación para mejorar los procesos de recomendación grupales, decidimos estudiar los diferentes comportamientos que tienen las personas en situaciones conflictivas de acuerdo a su personalidad.

En esta sección presentamos un método de recomendación grupal que distingue diferentes tipos de individuos en un grupo de acuerdo a su personalidad. Esta novedosa técnica propone la caracterización de usuarios por medio del test Thomas-Kilmann Conflict Mode Instrument (TKI) (Thomas y Kilmann, 1974). El TKI es un test diseñado para la medición del comportamiento humano en situaciones conflictivas. Hemos elegido este test en concreto ya que es un instrumento puntero en los estudios de determinación de conflictos. También es usado por asesores de Recursos Humanos y Desarrollo Organizativo para facilitar el aprendizaje sobre cómo los distintos estilos de manejar el conflicto afectan a dinámicas personales y grupales. Proporciona un valor tangible y estimable[1]. Y, al contrario que otros test similares, como *Ego Gram* Berne (1964), *Pen Model* Barrett et al. (1998) o NEO-PI-R Paul T. Costa y McCrae (1995), es fácil de interpretar (veáse (Quijano-Sánchez et al., 2014b) como ejemplo de discusión sobre la adecuación del test elegido). En nuestro caso, utilizamos el TKI para construir un perfil con la personalidad del usuario (p_u). Esto se hace a través de 30 preguntas con dos posibles respuestas[2]. Este perfil, describe el comportamiento de una persona en situaciones conflictivas por medio de dos dimensiones básicas: *autoritarismo* y *cooperacionismo*. Estas dos dimensiones de comportamiento se utilizan para definir los cinco tipos de personalidad existentes en situaciones conflictivas: competitiva, colaborativa, evasiva, complaciente y comprometida (véase Figura 3.1). Los detalles del cálculo del factor de personalidad (p_u) se pueden encontrar en las aportaciones presentadas en los Capítulos 10, (Quijano-Sánchez et al., 2010) y 18, (Quijano-Sánchez et

[1] `http://www.kilmanndiagnostics.com/catalog/thomas-kilmann-conflict-mode-instrument`
[2] Un ejemplo del test se encuentra `http://www.lara.warhalla.com/`

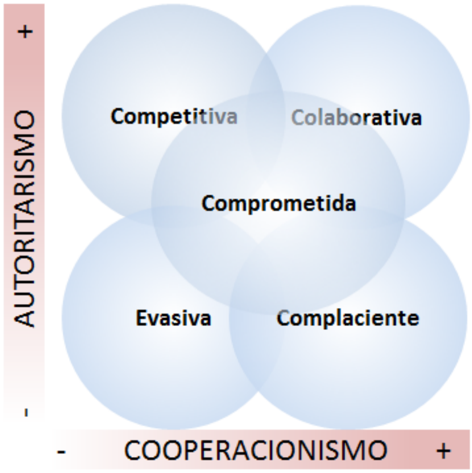


Figura 3.1: Tipos de personalidad TKI

al., 2013c).

Las aportaciones presentadas en los Capítulos 8, (Quijano-Sánchez et al., 2009) y 10, (Quijano-Sánchez et al., 2010) presentan un método inicial de nuestro *Modelo de Recomendación Social* (*MRS*)[3], llamado *Personality Based Recommendations* (*PBR*), que consiste en recomendar productos a un grupo mediante la obtención de los diferentes roles que las personas juegan cuando interactúan en procesos de toma de decisiones y que resumimos en la Ecuación 3.1.

$$pbr(\hat{r}_{u,i}, G_a) = \frac{1}{|G_a| - 1} \cdot \sum_{u \in G_a \wedge v \neq u} (\ \hat{r}_{u,i} + (p_u - p_v)\ ) \qquad (3.1)$$

En esta ecuación, $|G_a|$ representa el número de componentes que tiene el grupo activo G_a que recibe la recomendación (utilizamos este valor para normalizar el resultado). p_u y p_v son los valores de la personalidad de los usuarios u y v del grupo (en nuestro caso estos valores se computan usando el test TKI). Y $\hat{r}_{u,i}$ es el rating estimado para el usuario u y el producto i, que se ve incrementado si tras restar p_u y p_v observamos que la personalidad de u es más fuerte (asertiva) que la de v. $\hat{r}_{u,i}$ se estima por medio de cualquier algoritmo de recomendación individual deseado, ej. *colaborativo*, *basado en contenido*, etc.

En nuestra propuesta, la inclusión del factor de *personalidad* en el proceso de recomendación grupal sigue el siguiente razonamiento: el autoritarimo penaliza negativamente las diferencias entre las selecciones preferidas de otros miembros (pues esas selecciones no satisfacen sus propias inquie-

[3]Nuestro *MRS* (explicado en el capítulo 4) engloba a todos aquellos métodos de recomendación social que utilicen factores sociales.

tudes), mientras que el cooperacionismo recompensa las diferencias con las selecciones preferidas por otros miembros (ya que no es su selección pero es suficientemente buena para los otros componentes y por tanto para el grupo). Los experimentos realizados determinaron que la composición de la personalidad en el grupo influía ligeramente en la eficacia de la recomendación para el grupo. Esta mejoría sólo se alcanzada para cierto tipo de configuraciones grupales, como se comprobó al compararlo con diferentes algoritmos de recomendación grupal (Quijano-Sánchez et al., 2010, 2013c).

Una vez hemos estudiamos la caracterización individual de las personas en un grupo usando una valor que mide su personalidad, pasamos a estudiar otros factores relacionados con la estructura del grupo y cómo interactúan los usuarios entre ellos. Mediante la inclusión del factor de *personalidad* hemos evaluado el comportamiento social individual de las personas. Sin embargo, para modelar los procesos de toma de decisiones en grupo es importante, como vimos en la introducción de este capítulo, estudiar al grupo como una entidad propia y analizar su estructura y comportamiento como un todo y no sólo como la unión de sus miembros. Por tanto, necesitábamos explorar otros factores sociales, que explicamos a continuación.

3.3. Identificación de los factores sociales que influyen en la confianza entre personas y cómo obtenerlos a través de las redes sociales.

En las redes sociales, los usuarios publican en sus perfiles grandes cantidades de información personal que se puede analizar para calcular la *fuerza del vínculo* entre usuarios: gustos e intereses, información personal, fotos, juegos, etc. Estos factores son característicos de las *Redes sociales Humanas* (veáse 2.2.1 para una definición) y no se pueden extrapolar a otra clase de redes (Wu et al., 2010), como podría ser Twitter, que son *Redes sociales de Contenidos*. El uso de la confianza y otros conocimientos sociales obtenidos a través de las redes sociales en el desarrollo de sistemas recomendadores no es nuevo (Golbeck, 2006b; Avesani et al., 2005). A lo largo de este trabajo de Tesis, hemos revisado varios trabajos (Gilbert y Karahalios, 2009; Golbeck, 2006a) que identifican diferentes factores a ser analizados. En la mayoría de estos trabajos, la obtención de la confianza se hace consultando directamente a los usuarios, tarea que pueden llegar a resentir. Por ello, hemos ideado una manera no intrusiva de valorar la confianza entre dos usuarios. Esta estimación de la confianza se hace mediante la extracción de diferentes factores sociales existentes en los perfiles de los usuarios dentro de la red social. Es importante notar para poder pasar de la teoría a la práctica, que estos factores no son fácilmente cuantificables y que están limitados por el poder de extracción que las APIs de las redes sociales proporcionan.

Investigaciones anteriores han declarado que la *confianza* y la *fuerza del vínculo* son conceptualmente diferentes pero que existe una correlación entre ellas (Levin et al., 2004). Granovetter (1973) define la *fuerza del vínculo* como una (probablemente linear) combinación de cuatro factores: cantidad de tiempo, intensidad emocional, intimidad (confidencias mutuas), y servicios recíprocos que caracterizan el vínculo. La literatura revisada identifica estos cuatro factores como unas de las dimensions principales de variables predictivas. Con estas variables como guía, en Gilbert y Karahalios (2009) identificaron 74 variables de Facebook como predictores potenciales de la *fuerza del vínculo*. En este artículo, presentaron un diagrama que mostraba porcentajes que indicaban el poder de predicción de siete dimensiones elegidas para el cálculo de la *fuerza del vínculo*[4] y tres variables descriptivas de cada una de estas dimensiones. De todas las variables predictivas que Gilbert y Karahalios (2009) presenta, hemos seleccionado las que eran más representativas de cada dimensión principal y que también fuera posible extraerlas de los perfiles de usuario en la red social elegida (como hemos dicho anteriormente, estamos limitados por el poder de extracción que las redes sociales nos ofrecen). Una vez elegidas las diferentes variables involucradas en el cálculo de la *fuerza del vínculo* (entre las que se encuentran fotos en común, gustos e intereses, amigos en común o comentarios en el muro entre otras), las combinamos utilizando una media ponderada y obtenemos lo que llamamos nuestro factor de confianza $t_{u,v} \in (0, 1]$, que es una predicción de la *fuerza del vínculo* entre los usuarios u y v, donde el 0 representa vínculos débiles (personas de poca confianza) y el 1 vínculos fuertes (personas de gran confianza)[5].

Las aportaciones presentadas en los Capítulos 10, (Quijano-Sánchez et al., 2010), 18, (Quijano-Sánchez et al., 2013c) y 23 (Quijano-Sánchez et al., 2014b) presentan el estudio y el proceso de extracción que sigue nuestro método a la hora de calcular el factor de confianza. Somos conscientes de que la estimación de un valor que mide la *fuerza del vínculo* entre dos personas está influenciada directamente por la información extraída de la red social elegida (que en nuestro caso de estudio, Capítulo 6, ha sido Facebook), que hay varias formas de calcular una estimación de este valor, y que no es tan preciso como realizar una evaluación directa. Sin embargo, aunque no podemos concluir esta sección con un diseño único del cálculo del factor de

[4]Nótese que la cuatro dimensiones más decisivas de Gilbert y Karahalios (2009) son las mismas que Granovetter (1973) utiliza en su definición de *fuerza del vínculo* y por tanto, son las que nosotros hemos optado por utilizar, ya que la literatura revisada no ofrece una definición cerrada y mucho menos una especificación de cuantos niveles existen a la hora de medir la fuerza de un vínculo.

[5]Nótese que en el trabajo de esta Tesis hemos contemplado nuestro factor de confianza $t_{u,v}$ como la *tie strength fuerza del vínculo* entre dos usuarios desde un punto de vista general e independiente de cualquier dominio de recomendación. Un estudio diferente sería el de intentar computar $t_{u,v}$ como la confianza en las recomendaciones de alguien en un dominio en concreto, análisis que dejamos como posible línea de trabajo futura.

confianza, después de los estudios presentados en (Quijano-Sánchez et al., 2013c) o (Quijano-Sánchez et al., 2014b) entre otros, podemos afirmar que es una estimación útil, a efectos de la recomendación grupal, de la confianza entre usuarios.

3.4. Identificación de factores sociales adicionales que influyen en los procesos de toma de decisiones en grupo

Hasta ahora, hemos presentado el proceso de elicitación de nuestros dos factores sociales principales. El resultado de esta extracción ha sido la obtención de los factores de *personalidad* y *confianza*. Estos factores son la base de los métodos que pertenecen a nuestro *Modelo de Recomendación Social* (*MRS*), que como dijimos en el Capítulo 1 se trata de nuestro conjunto de algoritmos que integran factores sociales en distintas técnicas de recomendación grupal. Nótese que, además de los factores sociales de *personalidad* y *confianza*, a lo largo de esta Tesis Doctoral hemos estudiado otros tres factores sociales. Primero la *homofilia*, que es un factor extensamente estudiado en las ciencias sociales (Burt, 1982; Miller Mcpherson y Cook, 2001; McPherson y Smith-Lovin, 1987), referido a la idea de que los usuarios dentro de un sistema social tienden a asociarse más con aquellos usuarios a los que son similares y cuyo estudio en la difusión de información es importante en diversos contextos como en la comprensión de los *roles sociales* de los usuarios (Choudhury et al., 2010a,b). Este factor social se estudia en la aportación 12, (Quijano-Sánchez et al., 2011d). El segundo factor social adicional es la *persuasividad*, que se define como un intento deliberado de una parte en influir las actitudes o comportamientos de la otra parte para conseguir un objetivo predeterminado (Kaptein y van Halteren, 2013) y que cuando se aplica a procesos de *argumentación* (o de recomendación grupal) se contempla como una serie de mensajes persuasivos entre las partes para alcanzar un consenso (Todorov et al., 2002). Este factor social se estudia en la aportación presentada en el Capítulo 19, (Recio-García et al., 2013). Y el tercer factor es la *justicia*, que referida como factor social se entiende como la distribución equitativa de bienestar, oportunidades y privilegios dentro de una sociedad (Oxf, 2010) y que cuando se aplica al concepto de recomendación grupal puede referirse como la distribución homogénea de la *satisfacción* dentro un grupo (Liang et al., 2007; Masthoff y Gatt, 2006). Este factor social se estudia en la aportación presentada en el Capítulo 13, (Quijano-Sánchez et al., 2011c).

3.5. Conclusiones

El objetivo principal de esta Tesis es la mejora de las técnicas que típicamente se usaban a la hora de recomendar distintos tipos de productos a grupos. Como hemos mencionado anteriormente, previos estudios en sistemas recomendadores consideran por igual las preferencias de cada miembro de un grupo (McCarthy y Anagnost, 1998; McCarthy, 2002; Lieberman et al., 1999). Sin embargo, hay que considerar que no todos los grupos son iguales, éstos tienen diferentes características, como el tamaño o pueden estar formados por personas con preferencias similares o antagonistas. Nuestro enfoque presupone que la satisfacción general de un grupo respecto a una recomendación no se maximiza mediante la agregación simple de las preferencias de sus miembros. Lo novedoso de nuestro enfoque es el proceso de elicitación de factores sociales y su posterior inclusión en el proceso de recomendación grupal.

En este capítulo hemos presentado 5 factores sociales (*personalidad*, *confianza*, *homofilia*, *persuasividad* y *justicia*), que son los que se usarán en nuestro MRS. Además, es a través de los factores sociales presentados en este capítulo que mediremos el *Impacto de los factores y organizaciones sociales en los procesos de recomendación para grupos* y contrastaremos la hipótesis general formulada en esta Tesis: ***"La inclusión de elementos sociales mejora el rendimiento de las técnicas de recomendación grupal"***. En el siguiente capítulo detallaremos los métodos que pertenecen al MRS y una serie de experimentos realizados, tanto con usuarios reales como sintéticos, donde aplicando diferentes métodos de recomendación grupal (como agregación de preferencias sin factores sociales, inclusión del factor de *personalidad*, inclusión del factor de *confianza* o inclusión de otros factores sociales) mostramos una notable mejora en los resultados de las recomendaciones cuando se incluyen factores sociales (aportaciones presentadas en los Capítulos 18, (Quijano-Sánchez et al., 2013c), 22, (Quijano-Sánchez et al., 2014a) y 23, (Quijano-Sánchez et al., 2014b)).

Capítulo 4

Desarrollo de los métodos de recomendación social grupal que forman el MRS

4.1. Introducción

En el capítulo anterior, hemos explicado los beneficios potenciales de incluir factores sociales en los procesos de recomendación grupales. Las aportaciones presentadas en los Capítulos 18, (Quijano-Sánchez et al., 2013c), 22, (Quijano-Sánchez et al., 2014a) y 23, (Quijano-Sánchez et al., 2014b) demuestran que la inclusión de factores sociales, mejora considerablemente los resultados que previamente se obtenían con los métodos tradicionales de recomendación grupal. El objetivo de este capítulo es validar nuestra segunda hipótesis: ***"H2: Es posible desarrollar sistemas de recomendación grupal que modelen el comportamiento social que tienen los grupos de personas mediante la inclusión de factores sociales"***. Para ello, se explicarán los diferentes métodos que hemos diseñado para aplicar factores sociales en procesos de recomendación grupal. Este enfoque, que hemos llamado *Modelo de Recomendación Social* (MRS), utiliza junto con técnicas de recomendación individual y grupal tradicionales variables que miden la *personalidad* de cada componente del grupo y/o la *fuerza del vínculo* entre los miembros del grupo, es decir los factores p_u y $t_{u,v}$ explicados en el capítulo anterior. Además, como introducimos al final del capítulo anterior, también hemos estudiado la inclusión de factores sociales adicionales al MRS (en este caso secundarios pues acompañan a nuestros dos factores principales), estos son la *homofilia*, la *persuasividad* y la *justicia*.

En este capítulo explicaremos diferentes formas de diseñar y mejorar los procesos de recomendación grupal a través de la inclusión de estos cinco factores sociales. Para ello partiremos del pilar básico de cualquier método de recomendación grupal: la obtención de las preferencias individuales de los usuarios. Estas preferencias individuales, se pueden obtener de diferentes formas: se puede pedir directamente a los usuarios que valoren diferentes productos (Costello et al., 2006) o, lo mas común, se puede utilizar un recomendador individual (como los descritos en el Capítulo 2.1.1) que le proporcione al sistema estimaciones de las preferencias de cada usuario. En nuestro método utilizamos recomendadores individuales tanto *colaborativos* como *basados en contenido* según la necesidad del sistema. Estos recomendadores tienen como salida la estimación del rating $\hat{r}_{u,i}$ que cada usuario u perteneciente a un grupo activo G_a daría a cada item i del catálogo de productos a recomendar P. Una vez obtenidas las preferencias individuales de todos los miembros del grupo, el siguiente paso es elegir una función de agregación. La elección de una función de agregación adecuada es un elemento clave en el éxito de la recomendación generada para un grupo. Por este motivo (como veremos en la Sección 4.9), hemos revisado diferentes técnicas de agregación existentes y realizado un estudio de cuales son mejores para cada tipo de recomendación grupal (ya sea tradicional o social) y cada configuración grupal (grupos grandes, pequeños, etc) (Quijano-Sánchez et al., 2011a, 2014a).

Existen diferentes técnicas de agregación de preferencias individuales (Masthoff, 2004), siendo las estrategias de *Satisfacción media* (Abramowitz y Stegun, 1964) (donde se escoge la media), *Minimizar la miseria* (Masthoff, 2004) (donde se escoge el mínimo) y *Maximizar la satisfacción* (Masthoff, 2004) (donde se escoge el máximo) las más comúnmente usadas. Nuestro MRS esta basado en enfoques de agregación simple de preferencias. Estos enfoques (Masthoff y Gatt, 2006; O'Connor et al., 2001) obtienen una predicción para un grupo activo G_a utilizando una función de agregación ($\sqcup$ en la siguiente ecuación) que agrega los ratings individuales ($\hat{r}_{u,i}$) que han sido estimados para cada usuario u respecto al item i:

$$\hat{r}_{G_a,i} = \bigsqcup_{\forall u \in G_a} \hat{r}_{u,i} \tag{4.1}$$

Esta ecuación proporciona un rating estimado $\hat{r}_{G_a,i}$ para un determinado producto i y un grupo activo G_a. Tras la obtención de este resultado, los recomendadores grupales modelados por nuestro MRS, al igual que la mayoría de recomendadores grupales, proponen los k productos en P que tengan un mayor rating grupal estimado.

Este proceso de agregación simple (Ecuación 4.1) es común para la mayoría de los recomendadores grupales en la literatura. Sin embargo, es en este punto donde nosotros incluimos los factores sociales y por tanto nos desviamos de las técnicas tradicionales de recomendación grupal para reflejar así mejor los diferentes matices que existen en cada grupo distinto y que por tanto hacen que los procesos de toma de decisiones, es decir las recomendaciones, no sean siempre un proceso lineal sino variable en función de ciertos factores. Para reflejar las características del grupo, nuestro MRS modifica los ratings proporcionados por el recomendador individual $\hat{r}_{u,i}$ con nuestros factores sociales. De este modo, nuestro MRS se puede definir como el conjunto de métodos que siguen la Ecuación 4.2 en vez de la Ecuación 4.1.

$$\hat{r}_{G_a,i} \quad = \bigsqcup_{\forall u,v \in G_a \wedge v \neq u} FuncionSocial(\ \hat{r}_{u,i}\ ,\ p_u\ ,\ t_{u,v}\ ,\ fs\ ) \tag{4.2}$$

Donde $\hat{r}_{G_a,i}$ es el rating estimado para un determinado producto i y un grupo activo G_a, p_u está en un rango de $(0,1]$, donde el 0 representa una personalidad cooperativa y el 1 una personalidad asertiva y $t_{u,v}$ está en un rango de $(0,1]$, donde el 0 representa una confianza leve y el 1 una gran confianza entre los miembros del grupo. El último elemento, fs, es un conjunto de factores sociales que pueden o no ser incluidos dentro de la función social ($\{fs_1, .., fs_n\}$). En nuestro caso particular $n = 3$, y se refiere a los factores sociales secundarios estudiados hasta el momento: *homofilia*, *persuasividad* y *justicia*.

Hemos ideado diferentes *Funciones Sociales* (véase Ecuación 4.2) que se encargan de combinar los ratings predichos ($\hat{r}_{u,i}$) con los factores de *personalidad* (p_u), *confianza* ($t_{u,v}$) y otros. En cada una de las siguientes secciones de este capítulo (Secciones 4.2-4.8) explicaremos una forma distinta de combinar estos factores sociales y diferentes motivaciones y técnicas para realizarlo, i.e., los diferentes métodos que componen nuestro MRS.

4.2. Propuesta de un método de recomendación basado en delegación, DBR (Delegation-Based Recommendations)

En esta sección explicamos nuestro método de recomendación basada en delegación (*DBR*). Este método (que aquí sólo resumimos) se explica extensamente en las aportaciones presentadas en los Capítulos 18, (Quijano-Sánchez et al., 2013c) y 22, (Quijano-Sánchez et al., 2014a) entre otras. El enfoque basado en delegación sigue las ideas presentadas en (Golbeck, 2006a), donde el rating individual estimado de una persona está basado en los ratings dados por otros usuarios. La idea detrás de este enfoque es que la opinión de un usuario se crea basándose en las opiniones de su entorno social. El promedio de éstas opiniones se calibra dependiendo del nivel de confianza con cada amigo. Adicionalmente, en nuestra propuesta, la personalidad de cada amigo también se tiene en cuenta modificando así la opinión base. Básicamente, en el turno de cada usuario $u \in G_a, |G_a| = n$, la opinión de este usuario no se tiene en cuenta, sino en los otros (n-1) turnos que es cuando este usuario influenciará las opiniones de otros. En vez de usar la información que nos da la opinión del usuario una única vez, este método la tiene en cuenta cada vez que otro usuario del grupo expresa su opinión.

El DBR para un usuario u en un grupo activo G_a y un producto i del catálogo de productos a recomendar P se calcula del siguiente modo[1]:

$$
\begin{aligned}
dbr(\hat{r}_{u,i}, G_a) &= \frac{1}{T} \sum_{v \neq u \in G_a} t_{u,v} [\, \hat{r}_{v,i} + \theta_{v,i} \cdot (p_v - p_u) \,] \qquad (4.3) \\
\text{donde} & \\
T &= \sum_{v \neq u \in G_a} t_{u,v}
\end{aligned}
$$

Como vemos en esta ecuación utilizamos las predicciones de las preferencias $\hat{r}_{v,i}$ para cada amigo v y producto i. Este rating estimado se aumenta o disminuye dependiendo de las diferencias entre las personalidades de los dos amigos (u y v), $p_v - p_u$. De este modo, si el usuario v tiene una personalidad fuerte (o egoísta) su opinión tendrá mas impacto en la predicción para el usuario u. Sin embargo, tenemos que matizar que un usuario v con una personalidad fuerte y una opinión positiva del producto i, $\hat{r}_{v,i}$, intentará incrementar la valoración de u para ese producto. Pero, si en caso contrario v tiene una mala opinión del producto, intentará que la opinión de u decaiga. Hemos modelado este comportamiento por medio del parámetro $\theta_{v,i}$, véase, suponiendo que $\hat{r}_{v,i}$ se encuentra en un rango de [a,b]:

$$
\theta_{v,i} = \begin{cases} 5 & \text{if } \hat{r}_{v,i} \geq \frac{b-a}{2} \\ -5 & \text{if } \hat{r}_{v,i} < \frac{b-a}{2} \end{cases} \qquad (4.4)
$$

Hemos elegido estos valores (5 y -5) para la constante $\theta_{v,i}$ ya que tras varios estudios sobre la composición de la personalidad en diferentes grupos (Quijano-Sánchez et al., 2011b, 2014a) hemos observado que la desviación típica de la personalidad en cada grupo es generalmente 0.2 y por tanto el impacto de $\theta_{r_{v,i}} \cdot (p_v - p_u)$ en la Ecuación 4.3 se aproximará a $\pm 5 * 0{,}2 \approx \pm 1$, que representa el rango más adecuado en comparación con otros estudiados ($\theta_{r_{v,i}} = 1, .., 10$).

Finalmente, la predicción del usuario v, que ha sido modificada en función de la diferencia de personalidades entre estos dos componentes del grupo (u y v), se calibra con la confianza entre los dos usuarios $t_{u,v}$. Debe resaltarse que esta ecuación no esta normalizada por el tamaño del grupo sino que usa

[1] La ecuación final que mostramos en esta Tesis es la que aparece en la aportación presentada en el Capítulo 22, Quijano-Sánchez et al. (2014a). Sin embargo, hemos de notar que el cálculo del *dbr* ha ido evolucionando (véase Tabla 4.1) y mejorando a lo largo de este trabajo de Tesis, entre otros motivos gracias a los comentarios recibidos por varios revisores de revistas y congresos. Por este motivo en trabajos previos (Quijano-Sánchez et al., 2010, 2013c) aparece una ecuación más simple a la aquí presentada.

Versión y artículo en que aparece	**Ecuación** DBR
V1. (Quijano-Sánchez et al., 2013c)	$dbr(\hat{r}_{u,i}, G_a) = \frac{1}{T} \sum_{v \neq u \in G_a} t_{u,v} [\, \hat{r}_{v,i} + p_v \,]$ donde $T = \sum_{v \neq u \in G_a} t_{u,v}$
V2. (Quijano-Sánchez et al., 2010)	$dbr(\hat{r}_{u,i}, G_a) = \frac{1}{T} \sum_{v \neq u \in G_a} t_{u,v} [\, \hat{r}_{v,i} + \alpha \cdot (p_v - p_u) \,]$ donde $T = \sum_{v \neq u \in G_a} t_{u,v}$ y $\alpha = 5$
V3. Quijano-Sánchez et al. (2014a)	$dbr(\hat{r}_{u,i}, G_a) = \frac{1}{T} \sum_{v \neq u \in G_a} t_{u,v} [\, \hat{r}_{v,i} + \theta_{v,i} \cdot (p_v - p_u) \,]$ donde $T = \sum_{v \neq u \in G_a} t_{u,v}$ y $\theta_{v,i} = \pm 5$

Tabla 4.1: Evolución del rating basado en delegación (*dbr*) a lo largo de esta Tesis. Nótese que el orden de las versiones no coincide con el orden de las fechas de publicación. Esto es debido a los retrasos en el proceso de publicación de artículos.

la confianza acumulada[2] (representada como T). Hemos elegido esta opción siguiendo el estudio de (Golbeck, 2006b), donde se utiliza un método de recomendación grupal usando la confianza de este mismo modo.

Aunque la idea de utilizar las preferencias de los otros miembros del grupo y no las del propio usuario dentro del método *DBR* podría parecer anti-intuitiva, hemos realizado varios experimentos con métodos alternativos, p. ej. el método *IBR* que explicaremos a continuación, y todos han proporcionado recomendaciones peores que el *DBR*. Vemos estas conclusiones reflejadas en la Figura 4.2 que presentaremos más adelante junto con los otros dos métodos de recomendación que en ella aparecen (IBR y $Coaliciones$). Una vez determinado que el método *DBR* era de nuestros métodos sociales el que mejores resultados obtenía, en Quijano-Sánchez et al. (2014a) realizamos un experimento para demostrar la importancia de incluir factores sociales a los procesos de recomendación grupal. En él, construimos 4 recomendadores grupales: un recomendador que utiliza técnicas de agregación simples, es decir, que no incluye factores sociales y otros tres que representan la inclusión gradual de nuestros dos factores sociales principales (*personalidad* y *confianza*). Estos experimentos nos permitieron demostrar, como se ver en la Figura 4.1, que el recomendador que tenía en cuenta nuestros dos factores sociales principales, i.e. nuestro método *DBR*, era el que mejores resultados obtenía. En estos experimentos comparamos la decisión que cada grupo observado (15 en total) nos proporcionó con los resultados obtenidos por los 4 recomendadores grupales. Para ello utilizamos una función de evaluación, llamada *acierto@3* (del inglés *success@3*) (explicada en el Capítulo 2, Sección 2.1.2.2), que valora si hay al menos un acierto[3] en las primeras *3* posiciones.

[2]Los valores de confianza son siempre mayores que 0 así que no tenemos problemas con esta normalización.

[3]Decimos que hay un acierto si el recomendador predice correctamente al menos un producto que se encuentre en la lista de los 3 productos elegidos por el grupo.

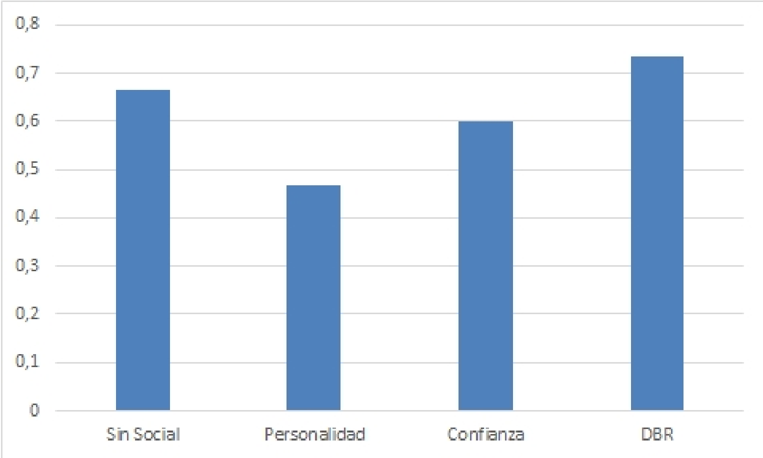


Figura 4.1: Porcentaje de aciertos de los 4 recomendadores grupales implementados utilizando como función de agregación la función satisfacción media.

Los 4 recomendadores grupales que comparamos fueron:

- *Sin social*. Un recomendador grupal estándar que utiliza la función de agregación *Satisfacción media* siguiendo la Ecuación 4.1).
- *Personalidad*. Que sólo utiliza los valores de personalidad y que implementa el método *Personality Based Recommendation* descrito en el Capítulo 3, Sección 3.2 (Ecuación 3.1).
- *Confianza*. Que implementa el método *Delegation-Based Recommendation* (*DBR*) presentado en esta sección (Ecuación 4.3), pero que sólo utiliza los valores de confianza $t_{u,v}$ (los valores p_u se anulan).
- *DBR*. Que representa el algoritmo completo con Confianza y Personalidad del método *DBR* (Equation 4.3).

En resumen, el método *DBR*, intenta simular el siguiente comportamiento: a la hora de elegir un producto en grupo, generalmente contrastaremos nuestras opiniones con aquellos miembros del grupo con los que tenemos más confianza. A su vez, esa confianza hace que tengamos más en cuenta las reacciones y comportamientos de esos usuarios (su personalidad), y que le demos más importancia a sus preferencias que a otras. Por ejemplo, podemos saber que un amigo (que tiene una personalidad fuerte/asertiva) se enfada fácilmente si no escogemos el producto que ella/el quiere. Este es el método que hasta ahora ha proporcionando mejores resultados de recomendación (cuando se ha comparado con los métodos *IBR* y *coaliciones* que explicaremos a continuación). Sin embargo, creemos que se podrían realizar más experimentos que permitiesen determinar de entre todos los métodos que pertenecen a nuestro MRS cuál es el que proporciona mejores resultados y, más importante, que condiciones o configuraciones grupales hacen que un método sea mejor que otro. Esta intuición nos lleva a la idea de recomendadores adaptativos, presentados en la sección de trabajo futuro del Capítulo 7.

4.3. Propuesta de un método de recomendación basado en influencia, IBR (Influence-Based Recommendations)

En esta sección explicamos el método *IBR*, que es otra de las estrategias de recomendación social que pertenecen al MRS. Este método (que aquí sólo resumimos) se detalla en el Capítulo 18, (Quijano-Sánchez et al., 2013c). El método *IBR*, al igual que el anterior (*DBR*), se centra en la inclusión de factores sociales a las técnicas de agregación de preferencias tradicionales. Sin embargo, en vez de centrarse en el concepto de delegación se centra en el concepto de influencia, por ello la llamamos técnica de recomendación basada en influencia. El enfoque basado en influencia simula la influencia que tiene cada miembro del grupo en una determinada persona. En vez de crear una nueva preferencia, parte de la suposición de que el usuario modificará sus preferencias por un producto dado, dependiendo de las preferencias de los otros miembros del grupo por ese mismo. Por ejemplo, si nuestra opinión de un producto esta estimada con un rating de 3 y tenemos un amigo que tiene un rating para el mismo producto de 5, podemos pensar en modificar nuestro rating a 4. Dependiendo de la confianza que nos inspire este amigo en concreto decidiremos el nivel de variación de nuestro rating (i.e. daremos 3.5 si es baja, o 4.5 si la confianza es alta). Esta modificación en nuestra opinión dependerá de nuestra propia personalidad. Si tenemos una personalidad muy fuerte (un valor del factor de personalidad alto) no estaremos dispuestos a cambiar nuestra valoración del producto, pero si en cambio tenemos una personalidad afable (un valor del factor de personalidad bajo), puede que las opiniones de otros nos influyan más. Estos conceptos se reflejan en la siguiente ecuación:

$$ibr(\hat{r}_{u,i}, G_a) = \hat{r}_{u,i} + (1 - p_u) \frac{\sum\limits_{v \in G_a \wedge v \neq u} t_{u,v} \cdot (\ \hat{r}_{v,i} - \hat{r}_{u,i}\ )}{|G_a| - 1} \tag{4.5}$$

Donde la estimación del rating del usuario u perteneciente al grupo activo G_a para el producto i, siguiendo nuestro método *IBR*, $ibr(\hat{r}_{u,i}, G_a)$, se calcula modificando la predicción del rating individual $\hat{r}_{u,i}$ en función de su diferencia con los ratings de los otros miembros del grupo ($\hat{r}_{v,i} - \hat{r}_{u,i}$). Esta diferencia esta calibrada con la confianza entre miembros del grupo ($t_{u,v}$) y la personalidad del usuario u (p_u).

Podemos encontrar los resultados de la evaluación del método *IBR* en los artículos (Quijano-Sánchez et al., 2013c) y (Quijano-Sánchez et al., 2011d) y además, un resumen de estos en la Figura 4.2.

4.4. Propuesta de un método de recomendación basado en coaliciones

En esta sección explicamos nuestro método basado en *Coaliciones* que es otro método de nuestro modelo. Este método sigue un nuevo enfoque para resolver las situaciones conflictivas (definidas en la Sección 3.2) por medio del modelado de las interacciones entre usuarios en los sistemas de recomendación grupal. Aquí, en vez de calcular una recomendación global para un grupo activo G_a basándonos solamente en las preferencias individuales de cada uno de sus componentes así como en sus personalidades y confianza mutua, estudiamos un modelo en el que cada usuario intenta negociar y convencer a los otros miembros del grupo sobre un producto en el que esta interesado. Para ello, utilizamos el concepto de *homofilia*, término utilizado para referirse a la tendencia que tiene todo individuo de asociarse y vincularse con gente que comparte sus mismos intereses. Esta característica se ha visto reflejada en muchas redes sociales (Miller Mcpherson y Cook, 2001; Lazarsfeld y Merton, 1954). En nuestro modelo, los usuarios con personalidades fuertes intentarán crear *alianzas* con otros usuarios que apoyen sus preferencias. De este modo, este tipo de usuarios, que llamaremos *influenciadores*, van obteniendo votos para conseguir que el producto en el que están interesados sea el elegido por el grupo. Estos *influenciadores* intentan, como la propia palabra indica, influenciar a otros usuarios y utilizar su posición de liderazgo para crear una alianza dentro del grupo.

Típicamente se caracteriza a los *influenciadores* como los "lideres" de un grupo o como personas con personalidades atractivas que tienen la habilidad de influir en las decisiones y opiniones de otros. En la práctica, podemos definir a estos individuos, también llamados *conectores* (Gladwell, 2000), como personas altamente conectadas con el entorno o individuos que unen dos sub-comunidades relativamente grandes. Este comportamiento social se ha investigado ampliamente en el campo de las ciencias sociales durante las últimas décadas (Burt, 1982; Miller Mcpherson y Cook, 2001).

Nuestro método basado en *coaliciones*, explicado en la aportación presentada en el Capítulo 12, (Quijano-Sánchez et al., 2011d), utiliza los factores de personalidad y confianza (definidos en el Capítulo 3) como un medio para definir *alianzas* dentro de un grupo. Definimos una alianza como un subgrupo que coincide en el mismo resultado. Un líder crea alianzas con otros

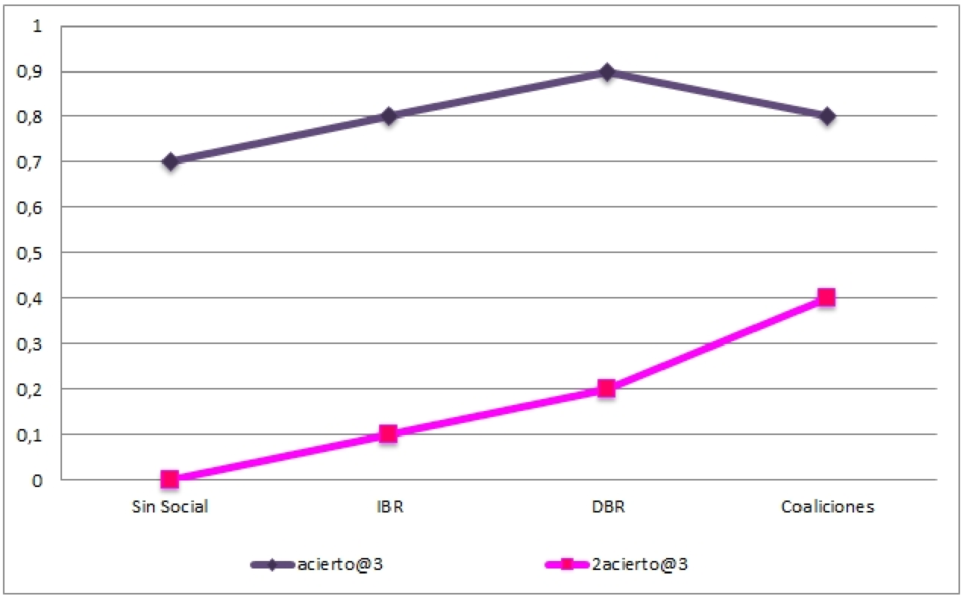


Figura 4.2: Porcentage de aciertos de los 4 recomendadores grupales implementados.

usuarios en los que confía para que le apoyen en la decisión de un producto. Nuestra recomendación propondrá el producto que haya elegido el líder que consiga la alianza con más miembros. Los líderes ($l \in L_a \subset G_a$) de un grupo activo G_a se identifican como aquellos cuya personalidad supera un cierto umbral de "personalidad fuerte". Los "posibles aliados" ($PA(l)$) son aquellos que tienen una confianza con el líder también superior a un cierto umbral de "gran confianza". Una vez identificados los líderes y sus "posibles aliados" empieza el proceso de formación de *alianzas*, para ello buscamos qué productos i dentro de un catálogo de productos P son los que elegiría cada líder. A estos productos los denominamos "favoritos del líder" $Fav(l)$ y los obtenemos seleccionando los k productos mejores $\hat{r}_{u,i}$ que nos devuelve el recomendador individual. Una vez que el líder tiene su lista de favoritos $Fav(l)$ le propone a cada usuario $v \in PA(l)$ cada producto $i \in Fav(l)$. Si el rating estimado para ese producto i, $\hat{r}_{v,i}$, es mayor que un umbral δ entonces añadimos al usuario v a la lista de aliados del líder, $LA(l)$. δ se calcula mediante una función que tiene en cuenta las valoraciones del usuario, su personalidad (el umbral es mayor cuanto mayor sea la personalidad del usuario, pues será mas difícil de convencer) y de la confianza con el líder (el umbral será menor cuanto más confíe el usuario en el líder). El recomendador propondrá la lista $Fav(l)$ que pertenezca al líder que consiga una lista $LA(l)$ con más aliados.

En cuanto a los resultados de este método, en la aportación presentada en el Capítulo 12, (Quijano-Sánchez et al., 2011d) realizamos experimentos utilizando la función de evaluación *acierto@3* (descrita en el Capítulo 2, Sección

2.1.2.2 y también utilizada en la Sección 4.2) donde comparamos los tres recomendadores sociales (pertenecientes al *MRS*) descritos hasta ahora: *DBR* (Ecuación 4.3), *IBR* (Ecuación 4.5) y un recomendador que utiliza nuestro método basado en *coaliciones* con un recomendador *No social* (Ecuación 4.1). Los resultados, que aquí resumimos en la Figura 4.2, muestran que al igual que las otras dos propuestas de recomendación social el método basado en *coaliciones* mejora los resultados (el porcentaje de aciertos global) del recomendador que no utiliza factores sociales. Cuando comparamos, utilizando la función *acierto@3*, el método basado en *coaliciones* con el método *DBR* vemos que, al igual que concluimos en la Sección 4.2, este último no mejora al *DBR*. Sin embargo, si utilizamos como función de evaluación una mas restrictiva, *2acierto@3*[4] vemos que el método basado en *coaliciones* mejora significativamente las otras técnicas de recomendación.

Los tres métodos propuestos hasta ahora: *DBR*, *IBR* y *Coaliciones*, se encargan de generar estáticamente recomendaciones grupales utilizando algunos de los factores sociales propuestos en esta Tesis. Sin embargo, en nuestra investigación hemos querido dar un paso más en el estudio del *Impacto de los factores y organizaciones sociales en procesos de recomendación grupal* teniendo en cuenta un factor social adicional, la *persuasividad*. Para ello, a continuación proponemos un método que utiliza la topología de red social en la que se organiza un determinado grupo que recibe una recomendación para realizar recomendaciones sociales siguiendo un proceso de argumentación dinámica.

4.5. Propuesta de un método de recomendación basado en modelos distribuidos y argumentación

En secciones anteriores hemos descrito los factores sociales principales estudiados en nuestro *MRS*: la *personalidad* (Sección 3.2) y la *confianza* entre los miembros del grupo (Sección 3.3). Ahora, continuamos nuestra investigación con el estudio de la *persuasividad* dentro de una arquitectura grupal que imite las conexiones de redes sociales en los procesos de recomendación. Nuestro objetivo en las aportaciones presentadas en los Capítulos 9, (Recio-García et al., 2010) y 19, (Recio-García et al., 2013) es mejorar las recomendaciones grupales teniendo en cuenta la topología de la red y los factores sociales de *personalidad*, *confianza* y *persuasividad*. Para realizar este estudio utilizaremos sistemas multiagentes con topología de red social, donde cada agente será un elemento persuasivo representando a un miembro del grupo. Cada uno de los agentes representando a cada usuario defenderá sus mejores intereses *argumentando* con los otros agentes a los que está conec-

[4]En vez de comprobar si el recomendador predice correctamente al menos un producto que se encuentre en la lista de los 3 productos elegidos por el grupo, comprobar si predice correctamente al menos dos productos.

tado. De nuevo, siguiendo nuestra propuesta de recomendaciones sociales, la recomendación final se verá influenciada por la personalidad de cada usuario y la confianza mutua.

En resumen, nuestro objetivo es incluir la posibilidad de generar procesos de argumentación dinámicos en vez de procesos de recomendación estática. Para ello, proponemos un nuevo método para nuestro MRS, que consiste en utilizar una arquitectura distribuida formada por agentes con capacidad de deliberación que debaten y defienden las preferencias del usuario al que representan para llegar a un consenso. En la red de agentes cada agente tiene definido la capacidad de persuasion (Kaptein y van Halteren, 2013; Todorov et al., 2002) de los otros agentes a los que está conectado y la personalidad del usuario al que representa. Este modelo está basado en la idea de utilizar las conexiones de los agentes colaboradores junto con el nivel de confianza existente con el agente a colaborar en cada momento (McDonald, 2003; Golbeck, 2006b; Ziegler y Golbeck, 2007). El objetivo principal de esta aportación es la mejora de los resultados en sistemas de decisión grupales por medio de un modelo distribuido con topología de red social, la inclusión de factores sociales, como la personalidad, la confianza y la persuasividad, y además un proceso de argumentación que permite a los usuarios discutir y defender sus opiniones delegando esta tarea en los agentes que les representan. En los experimentos llevados a cabo en las aportaciones presentadas en los Capítulos 9, (Recio-García et al., 2010) y 19, (Recio-García et al., 2013) hemos podido concluir que los modelos distribuidos y las técnicas de argumentación que incluyen la personalidad y la confianza mejoran la satisfacción de los usuarios envueltos en procesos de toma de decisiones en grupo.

Hasta ahora hemos visto diferentes técnicas de incluir nuestros factores sociales en métodos de recomendación grupal (tanto de forma estática: DBR, IBR y $Coaliciones$, como de forma dinámica: $ModelosDistribuidos$). Sin embargo, como vemos a continuación, en esta Tesis hemos estudiado la importancia de incluir un último factor social, la *justicia*, que nos permite homogeneizar la satisfacción de los usuarios con recomendaciones pasadas.

4.6. Propuesta de un método de recomendación basado en memoria

Hasta ahora, nos hemos centrado en la situación en la que el sistema hará recomendaciones sólo una vez. Pero frecuentemente, podemos esperar que un grupo utilice el sistema varias veces, y que por tanto obtenga una muestra mayor de recomendaciones. Las nuevas técnicas de recomendación que hemos propuesto hasta ahora (nuestros métodos incluidos en el MRS) siempre tienden a favorecer a los mismos usuarios, ya sea porque tienen personalidades más fuertes o porque tienen más confianza con otros usuarios del grupo. Es por esto que se podría dar el caso que a la larga tuviéramos usuarios insatisfechos por que sus opiniones se hayan tenido menos en cuenta por el bien del grupo. Para evitar situaciones en las que la satisfacción está descompensada entre los diferentes miembros de un grupo, proponemos un nuevo método para nuestro modelo donde tenemos en cuenta un factor social adicional: la *justicia*. A motivación para esta decisión es el caso hipotético de que una recomendación resulte muy buena para el grupo en general pero a costa de esa recomendación un usuario quede especialmente insatisfecho. En ese caso, sería deseable que las recomendaciones futuras favoreciesen a ese componente del grupo en concreto para que alcance un nivel apropiado de satisfacción.

Para poder resolver este tipo de situaciones, proponemos un sistema que tenga memoria sobre las recomendaciones pasadas. De este modo, si un miembro del grupo acepta una propuesta que no es muy de su agrado, en el próximo evento sus preferencias tendrán prioridad en el proceso de recomendación. Esto significa que su opinión tendrá más peso la siguiente vez. Estos pesos también han de ser calibrados según las diferentes personalidades que existen en un grupo. Por ejemplo, hay algunos usuarios que aunque no les guste el producto recomendado (le darían un rating bajo) puede que estén contentos con la recomendación, especialmente si creen que ha sido la opción que mejor resolvía los conflictos de intereses. Además, es probable que esta satisfacción sea mayor si el usuario tiene un tipo de personalidad cooperativa (poco egoísta), o si la recomendación favorece a los componentes del grupo con los que tiene más relación, viéndose sus opiniones influenciadas por motivos de *contagio*[5] o *conformidad*[6] Masthoff y Gatt (2006). Nuestra propuesta modela este comportamiento compensando inmediatamente a los usuarios cuyos intereses se han visto perjudicados y tienen personalidades "fuertes" puesto que comprendemos que los usuarios con personalidades más apacibles están más predispuestos a ceder más veces.

Creamos una variable s_u que refleja el nivel de satisfacción de un usuario u, donde un usuario que esté completamente contento con las recomendaciones tendrá la satisfacción a 1. Y, sin embargo cuanto más aumente su malestar más irá decreciendo este valor, llegando a 0 en el peor de los casos. El siguiente paso es incluir este factor que representa la *justicia* dentro de un grupo en nuestro *MRS*. Por ejemplo, si utilizamos la propuesta *DBR* (Sección 4.2) la ecuación a seguir sería:

[5]Influencia en el estado afectivo de un individuo debido al contagio del estado de otros dentro de un mismo grupo (Barsade, 2002; Hatfield et al., 1994).

[6]Acción que causa que un individuo cambie su opinión con respecto a algo debido bien a la presión grupal (querer agradar a otros) o bien a la mayor confianza en el juicio de otros que en la de uno mismo (Deutsch y Gerard, 1955).

$$
\begin{aligned}
dbr_m(\hat{r}_{u,i}, G_a) &= \frac{1}{T} \sum_{v \neq u \in G_a} t_{u,v} [\, \hat{r}_{v,i} + \theta_{r_{v,i}} \cdot (p_v - p_u) \,] + m_v \\
\text{donde} & \\
T &= \sum_{v \neq u \in G} t_{u,v} \\
m_v &= \alpha(1 - s_v) p_v
\end{aligned}
\tag{4.6}
$$

Donde m_v es el factor de memoria de recomendaciones pasadas. El parámetro $\alpha \epsilon (0,1)$ se utiliza para modificar el impacto de la memoria en el dbr_m, y tiene un valor positivo o negativo en función del valor de $\hat{r}_{v,i}$ del mismo modo que utilizábamos $\theta_{r_{v,i}}$ en el cálculo del *dbr* (Sección 4.2). Nótese que inicialmente a cada usuario se le asigna un valor $s_v = 1$. Por tanto, la primera vez que un grupo recibe una recomendación el factor m_v queda anulado en la ecuación, ya que no es necesario pues no existen recomendaciones anteriores. Finalmente, podemos observar como S_u se calibra en función de la personalidad del usuario v (p_v) para reflejar la importancia de satisfacer a ese usuario en concreto.

Tras la obtención del producto/s a recomendar (Ecuación 4.2), el último paso a tomar en los procesos de recomendación que utilizan un sistema de memoria de recomendaciones pasadas es actualizar la variable s_u para cada miembro del grupo ($u \epsilon G_a$) en función de su satisfacción individual con la recomendación grupal resultante ($\hat{r}_{G_a,i}$). Esto se realiza mediante la siguiente ecuación:

$$s_u(t) = (1 - \delta) \cdot is_u(t) + \delta \cdot s_u(t-1) \tag{4.7}$$

Donde $s_u(t)$ es la satisfacción acumulada del usuario u con respecto al grupo G_a. $is_u(t)$ es la satisfacción instantánea, calculada tras recibir la última recomendación. Y, $\delta \in$ [0.,1] se utiliza para ajustar el impacto del valor anterior de satisfacción $s_u(t-1)$.

Nótese que este proceso se ha diseñado y se ha evaluado completamente en la aportación presentada en el Capítulo 23, (Quijano-Sánchez et al., 2014b) además de estudiarse en las aportaciones presentadas en los Capítulos 13, (Quijano-Sánchez et al., 2011c) y 18, (Quijano-Sánchez et al., 2013c).

4.7. Propuesta de un método de recomendación para resolver el problema del cold-start

Un problema que comúnmente presentan los recomendadores colaborativos es el denominado problema del *cold-start* (Herlocker, 2000; Schafer et al., 2007a). Este problema (como introdujimos en el Capítulo 2, Sección 2.1.1.2) se da cuando los recomendadores individuales tienen dificultades en proporcionar buenas predicciones a usuarios que son nuevos en el sistema y que por tanto disponen de pocas valoraciones en su perfil. Esta carencia de valoraciones hace muy difícil la tarea de encontrar el usuario más similar, pues como recordamos (Capítulo 2, Sección 2.1.1.1) es la metodología que siguen los recomendadores colaborativos. Los recomendadores grupales heredan este problema porque utilizan recomendadores individuales. Algunas soluciones que se han propuesto para el problema del *cold-start* en recomendadores individuales son: incluir promedios de las opiniones populares, pedir más valoraciones a los usuarios o utilizar recomendadores híbridos que utilicen también recomendadores basados en contenido (Schafer et al., 2007a).

Nuestra aportación presentada en el Capítulo 16, (Quijano-Sánchez et al., 2012b, 2013a) es una solución que utiliza técnicas de razonamiento basado en casos (CBR, del inglés Cased Based Reasoning) (Leake, 1996) y información relativa al grupo para mejorar los resultados de las recomendaciones grupales cuando se dan casos de *cold-start*. Utilizamos una base de casos CB en la que cada caso guarda la información relativa a un grupo y una recomendación pasada. Cuando un grupo activo G_a solicita una recomendación y uno o más de sus miembros está en *cold-start*, buscamos un caso $c \in CB$ de otro grupo similar, G_r, que contenga información sobre un recomendación anterior en la que los usuarios no estuvieran en *cold-start* y jugaran roles similares a los que juegan los usuarios del grupo activo. Básicamente lo que hacemos es: darle al usuario $u \in G_a$ que está en *cold-start* todos los ratings que él no tiene pero que su usuario más similar ($v \in G_r$) sí. Para ello, para cada grupo que tiene un usuario en *cold-start* encontramos el grupo más similar en nuestra base de casos CB. Identificamos al grupo más similar como aquel que tiene de media los usuarios más similares. A continuación, para cada usuario en *cold-start* en el grupo activo G_a buscamos el usuario más similar en el grupo recuperado G_r. Definimos la similitud entre usuarios como la media de cuan similares son estudiando sus valoraciones comunes, su edad, su género, su personalidad y confianza con otros miembros del grupo.

Los resultados de los experimentos que hemos realizado (y que se encuentran totalmente descritos en las aportaciones presentadas en el Capítulo 16, (Quijano-Sánchez et al., 2012b, 2013a)) muestran que nuestro método funciona mejor que cualquier otro método que no utilize información relativa al grupo (como es la personalidad y la confianza entre usuarios) o que no incluya ninguna consideración hacia el problema del *cold-start*. Es por esto que consideremos que nuestro método mejora la calidad de los recomendadores grupales.

4.8. Propuesta de un método de recomendación social basado en CBR

En las secciones anteriores (4.2-4.7) hemos presentado diferentes métodos (*DBR*, *IBR*, *Coaliciones*, *Memoria*, *ModelosDistribuidos* y *Cold-start*), que pertenecen a nuestro *MRS*, cuyo propósito es mejorar los resultados de los sistemas de recomendación grupales con la inclusión de factores sociales. La dinámica de estos métodos propuestos es combinar diferentes factores sociales utilizando las ecuaciones que hemos diseñado (por ejemplo las Ecuaciones 4.3,4.5 o 4.6). Como hemos visto en las secciones anteriores y en los artículos citados (Quijano-Sánchez et al., 2013c, 2011d, 2014a), estos *métodos sociales* que hemos diseñado si que mejoran los resultados de otros métodos de recomendación grupal que no utilizan factores sociales. Sin embargo, al tratarse de ecuaciones genéricas podría ocurrir que no fueran igual de eficientes para grupos con distintas características (grupos grandes o pequeños, grupos cuya personalidad global es principalmente fuerte, grupos con conexiones muy fuertes de amistad, etc). Por este motivo, en la aportación presentada en el Capítulo 17, (Quijano-Sánchez et al., 2012a) hemos extendido nuestro estudio sobre las recomendaciones a grupos de usuarios con un método que utiliza una base de casos de anteriores recomendaciones a grupos, y que nos permitirá reproducir patrones de comportamiento entre grupos similares.

Como hemos visto anteriormente (Sección 4.1), los sistemas de recomendación grupal generalmente agregan las diferentes predicciones que se han hecho para cada uno de los miembros del grupo (Jameson y Smyth, 2007). Es decir, utilizan un recomendador individual que predice las valoraciones que darían los usuarios a una colección de productos; luego, el recomendador grupal agrega estas predicciones. El método que aquí presentamos utiliza el mismo enfoque, agrega las preferencias de los miembros del grupo, sin embargo utiliza técnicas de razonamiento basado en casos (CBR) para la agregación, evitando así utilizar un mismo método de recomendación social para cualquier caracterización grupal.

Para ello, nuestro sistema almacena una base de casos de anteriores recomendaciones a grupos, CB. Cada caso c contiene los miembros del grupo al que representa; la lista de posibles productos a consumir; el producto que el grupo eligió tras la recomendación anterior; y las valoraciones que cada miembro del grupo dio al producto elegido tras realizar la actividad grupal. Para poder hacer una recomendación a un grupo activo G_a nuestro sistema utiliza una combinación de dos sistemas de recomendación colaborativos: *basados en usuarios* y *basados en productos* (Schafer et al., 2007a) (descritos en el Capítulo 2, Sección 2.1.1.1).

Primeramente, utilizamos el recomendador colaborativo *basados en usuarios* (Zhao y Shang, 2010) para realizar predicciones sobre las valoraciones que cada miembro del grupo le daría a cada posible producto a recomendar. Luego, recuperamos casos anteriores, es decir, anteriores recomendaciones grupales de grupos parecidos al grupo activo G_a. Para la recuperación de casos, el sistema usa una medida de similitud del tipo *usuario-usuario* (Resnick et al., 1994) y sigue el mismo proceso de las recomendaciones *basadas en productos* (Sarwar et al., 2001), esto es, asigna a cada miembro u del grupo activo G_a el miembro más parecido v del grupo G_r recuperado del caso c. La función de similitud compara por cada par de usuarios de grupos distintos, la edad, género, personalidad, valoraciones a productos y grados de confianza con los demás miembros de su grupo.

Una vez que el sistema recupera los k casos $c_1, .., c_k \in CB$ más similares al grupo G_a, reutiliza las contribuciones que cada miembro del grupo hizo a la hora de elegir un producto conjuntamente y se las transfiere a su usuario correspondiente en el grupo G_a. Es decir, predice los ratings que cada miembro u del grupo G_a le daría a los productos a recomendar utilizando las calificaciones que dió su usuario más similar v en su grupo correspondiente en el caso c. Esto se hace con técnicas de similitud de tipo *producto-producto* (Wang et al., 2006). La idea detrás de este proceso es que los casos recuperados actúen como modelos implícitos en los procesos de toma de decisiones. Finalmente, el sistema recomienda los productos que hayan obtenido las predicciones más altas.

La ventaja de este enfoque basado en casos en contraposición con los sistemas de agregación de preferencias (DBR, IBR, $Coaliciones$, etc) es que éste no requiere que nos ajustemos a un modelo de comportamiento social, expresado como un conjunto de ecuaciones que pueden ser o no válidas para todos los grupos posibles. Sino que, nuestro sistema de agregación con CBR realiza una predicción de valoraciones de productos como una generalización local de los comportamientos capturados por los casos vecinos en la base de casos.

4.9. Evaluación de nuestro MRS utilizando las diferentes técnicas de agregación existentes

En la introducción de este capítulo hemos explicado como nuestro *Modelo de Recomendación Social* (MRS) hace uso de un recomendador individual para obtener la estimación de los ratings de los miembros del grupo ($\hat{r}_{u,i}$), modifica estos ratings estimados mediante factores sociales ($FuncionSocial(\hat{r}_{u,i}, p_u, t_{u,v}, fs)$) y finalmente utiliza una función de agregación para combinar estos ratings (véase Ecuación 4.2). Como existen varias funciones de agregación posibles a utilizar (Masthoff, 2004) hemos realizado experimentos con varias de ellas y hemos descubierto que la función *Satisfacción media* es una de las que mejor funciona con nuestro MRS.

En las aportaciones presentadas en los Capítulos 14, (Quijano-Sánchez

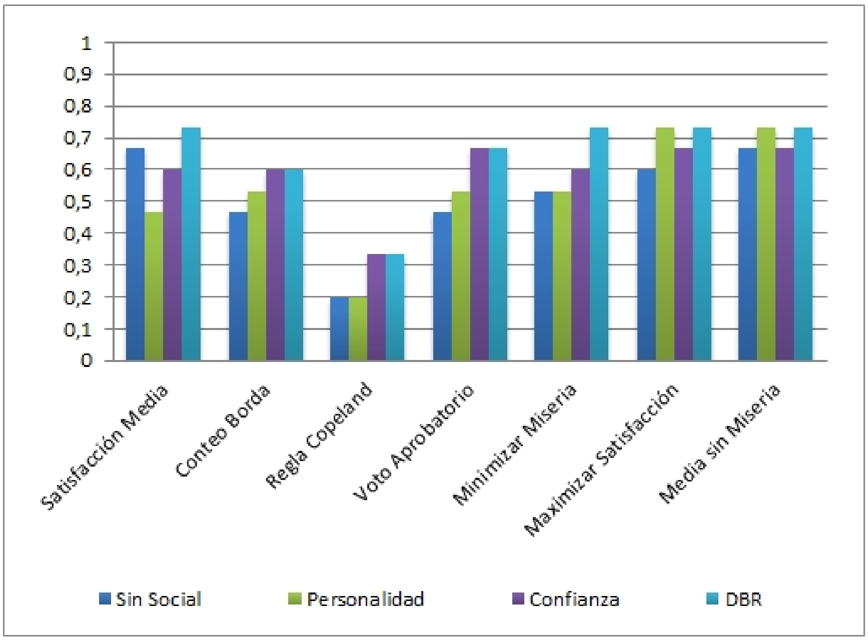


Figura 4.3: Resultados de la comparación de nuestro MRS (social), un método que no utiliza factores sociales (sin social), un método que utiliza solo el factor de personalidad (personalidad) y un método que utiliza solo el factor de confianza (confianza) utilizando diferentes funciones de agregación.

et al., 2011a) y 22, (Quijano-Sánchez et al., 2014a) hemos realizado experimentos tanto con usuarios reales como sintéticos en recomendadores que utilizan diferentes funciones de agregación estudiadas. El objetivo es estudiar cual era la función de agregación que mejor se adaptaba a los diferentes tipos de recomendación grupal ahí implementados. Para ello hemos realizado recomendaciones que utilizaban nuestro MRS, recomendaciones que sólo utilizaban el factor de personalidad, recomendaciones que sólo utilizaban el factor de confianza y recomendaciones que no utilizaban ningún factor social. Los resultados (véase Figura 4.3) muestran que ningún otro método supera a nuestro MRS (en este caso implementado mediante el método DBR) que, como recordamos utiliza principalmente dos factores sociales (*personalidad* y *confianza*) y que una de las funciones de agregación que mejor se comportan para este método es *Satisfacción media* (en inglés, *Average Satisfaction*). En Quijano-Sánchez et al. (2011a, 2014a) se encuentran todos los resultados obtenidos, conclusiones y detalles del experimento. En estos experimentos también hemos estudiado cual es la función de agregación que mejor se adapta a cada configuración grupal en función del tamaño. Para ello, hemos realizado recomendaciones a grupos de diferentes tamaños (véase Figura 4.4) esta vez sólo con nuestro MRS. Los resultados muestran que mientras que algunas funciones de agregación como *Satisfacción media* se comportan

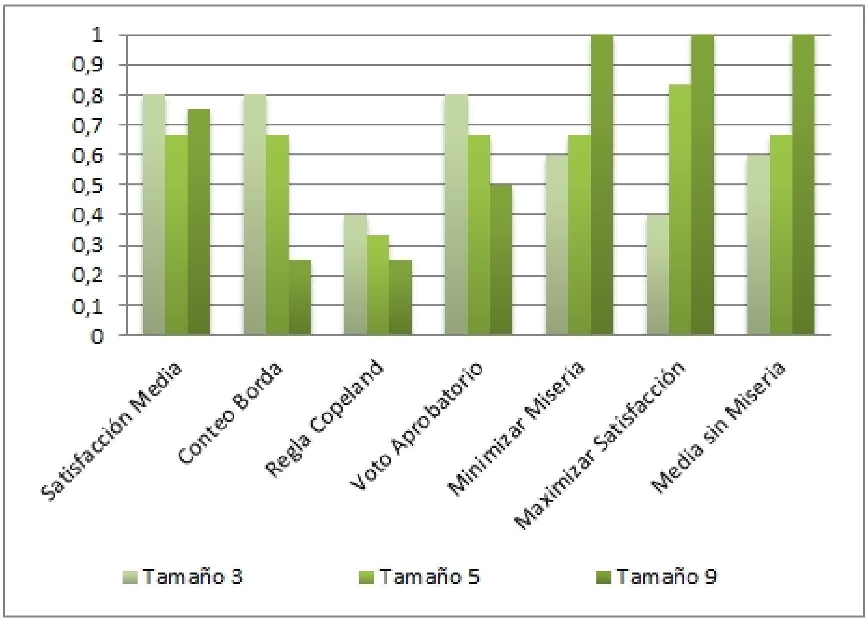


Figura 4.4: Resultados de la comparación de nuestro *MRS* para grupos de tamaño 3, 5 y 9.

mejor para grupos más pequeños, otras como *Minimizar la miseria* o *Maximizar la satisfacción* se comportan de manera contraria y funcionan mejor para grupos más grandes. Estos y otros resultados se pueden encontrar en (Quijano-Sánchez et al., 2011a, 2014a).

4.10. Evaluación de los métodos propuestos

Para comprobar el funcionamiento de los métodos propuestos a lo largo de este capítulo y observar si efectivamente suponen una mejora respecto a los recomendadores para grupos previamente existentes, hemos evaluado nuestros métodos usando el dominio de las películas. Hemos elegido este dominio porque es una área con diversas fuentes de datos (datasets) disponibles y por tanto obtener información acerca de películas es en una tarea relativamente fácil. Por otro lado, también ha influido a la hora de elegir un dominio de validación, el conocimiento general que cualquier usuario tiene sobre el mundo del cine. Este conocimiento le permite conocer el dominio sin necesidad de ser un experto, con lo que el uso del sistema y la evaluación de resultados resulta más sencilla.

A continuación, explicamos brevemente cómo ha sido la configuración de nuestros experimentos, que como se detallan en las aportaciones presentadas en los Capítulos 10, (Quijano-Sánchez et al., 2010), 14, (Quijano-Sánchez et al., 2011a) y 22, (Quijano-Sánchez et al., 2014a) entre otros artículos, ha consistido en la obtención de diferentes fuentes de datos y la simulación de nuestros *Métodos Sociales de Recomendación Grupal* (Secciones de la 4.2 a la 4.8).

Inicialmente, hemos realizado experimentos en los que utilizamos datos sintéticos. El motivo de utilizar inicialmente estos datos generados y no datos obtenidos de usuarios reales (como hemos hecho más tarde) es que queremos explorar casos extremos que puedan darse en situaciones de conflicto a la hora de recomendar a grupos de personas. Para ello necesitamos tener control de la distribución de los datos, cosa que no ocurre cuando se usan con datos reales.

Después de estudiar y analizar los resultados obtenidos en los experimentos con datos sintéticos, hemos continuado realizando experimentos siguiendo dos objetivos: (1) verificar que los resultados de los experimentos anteriores son fiables, es decir que la simulación de datos sintéticos es realista (este hecho se ha demostrado en varios artículos ((Quijano-Sánchez et al., 2010) y Quijano-Sánchez et al. (2014a) entre otros); (2) realizar un investigación más a fondo sobre cómo podemos mejorar las recomendaciones grupales por medio de la extracción de los factores sociales directamente de las redes sociales. Para ello debíamos realizar experimentos con datos de usuarios reales[7]. Para obtener los datos que nuestro experimentos necesitan hemos creado eventos en la Red Social Facebook[8] y le hemos pedido a distintos grupos de usuarios que se unan a ellos[9]. A través de estos eventos podemos extraer automáticamente los datos necesarios para realizar el cálculo de la confianza entre usuarios ($t_{u,v}$) y obtener el resto de datos necesarios por medio de 2 cuestionarios (que hemos presentado a los participantes de los eventos para que los rellenen)[10].

El primer cuestionario se utiliza para obtener el factor de personalidad (p_u). Para ello (como vimos en la Sección 3.2), llevamos a cabo el test de personalidad TKI (Thomas y Kilmann, 1974). El segundo cuestionario nos proporciona las preferencias individuales de cada usuario en el dominio de las

[7]Para facilitar la reproducibilidad de los resultados obtenidos hemos anonimizado los datos extraídos en nuestros experimentos y los hemos incluido para su descarga en `http://sourceforge.net/projects/jcolibri-cbr/files/misc/`.

[8]http://www.facebook.com

[9]Para esta evaluación algorítmica de los métodos pertenecientes a nuestro *MRS* utilizamos dos cuestionarios al margen de las redes sociales para obtener las preferencias y la personalidad de los usuarios, realizamos una evaluación presencial y únicamente utilizamos las redes sociales como medio para calcular el factor de confianza. Sin embargo, como veremos en el Capítulo 6, a lo largo del proceso de investigación realizado en esta Tesis Doctoral, mejoramos el proceso de evaluación y desarrollamos un prototipo de una aplicación en la red social Facebook, *HappyMovie*. Esta aplicación nos sirve como medio para extraer toda la información necesaria, haciendo el proceso más dinámico, como medio para presentar los resultados (recomendaciones), dándole visibilidad a nuestro trabajo, y como medio para obtener feedback, facilitando el proceso de evaluación.

[10]Los cuestionarios (en Español) están disponibles en `http://www.lara.warhalla.com/`.

películas ($r_{u,i}$). Los usuarios tiene que evaluar más de 40 películas heterogéneas seleccionadas de entre la base de datos de MovieLens (Bobadilla et al., 2009) (valorándolas entre 0 y 5). En nuestros experimentos han participado en torno a 58 usuarios.

Lo siguiente que necesitamos es una función de evaluación para medir la precisión del recomendador grupal y comparar los resultados con las decisiones que los usuarios habrían tomado si se hubiese tratado de una situación real. Para poder evaluar los resultados, les hemos pedido a los grupos de usuarios formados en los eventos de Facebook que se junten para debatir qué 3 películas (de entre una lista de 15 que simula una cartelera de cine) querrían ver en grupo, simulando que van a ir al cine juntos. Hemos conseguido reclutar a 15 grupos de 9, 5 y 3 miembros (4, 6 y 5 grupos respectivamente). Las 3 películas que cada grupo eligió se guardan como el conjunto de *favoritas reales del grupo*, *rgf*. De este modo, para evaluar la eficiencia de nuestros recomendadores podemos comparar la colección de las 3 mejores películas que nos ofrece el recomendador –el conjunto *gf* – con las preferencias reales del grupo – *rgf* –.

Finalmente, nuestros experimentos se han dividido en las siguientes partes: (1) hemos generado aleatoriamente grupos de usuarios con diferentes tipos de personalidades y de topología social para el caso de los experimentos con datos sintéticos y/o hemos utilizado los datos extraídos para los experimentos con datos reales; (2) hemos desarrollado un recomendador individual que sigue un enfoque colaborativo (Capítulo 2, Sección 2.1.1.1); (3) hemos creado distintos sistemas de recomendación para grupos que utilizaban el recomendador individual pero que implementaban diferentes enfoques: primeramente hemos desarrollado un recomendador estándar, también denominado *base* o *Sin Social* (explicado en la Sección 4.9, Ecuación 4.1), que sólo realiza agregación de preferencias. Seguidamente hemos creado otros dos recomendadores, uno que utilizaba únicamente el dato sobre la personalidad de cada individuo (como se explica en la Sección 3.2, Ecuación 3.1), y otro que sólo utilizaba la confianza entre usuarios (Secciones 3.3 y 4.2). Por último hemos implementado todos los métodos propuestos a lo largo de este capítulo, es decir nuestros *recomendadores sociales* que integran nuestro *MRS*, teniendo en cuenta nuestros factores sociales de *personalidad*, *confianza*, *homofilia*, *persuasividad* y *justicia*: el *DBR* (Sección 4.2, Ecuación 4.3), el *IBR* (Sección 4.3, Ecuación 4.5), el basado en *Coaliciones* (Sección 4.4), el basado en *ModelosDistribuidos* (Sección 4.5), los basados en *memoria* (Sección 4.6, Ecuación 4.6), los basados en el problema del *cold-start* (Sección 4.7) y los basados en CBR (Sección 4.8); (4) comparamos los resultados obtenidos con los diferentes recomendadores (*gf*) con los datos que reflejan las decisiones reales de cada grupo (*rgf*).

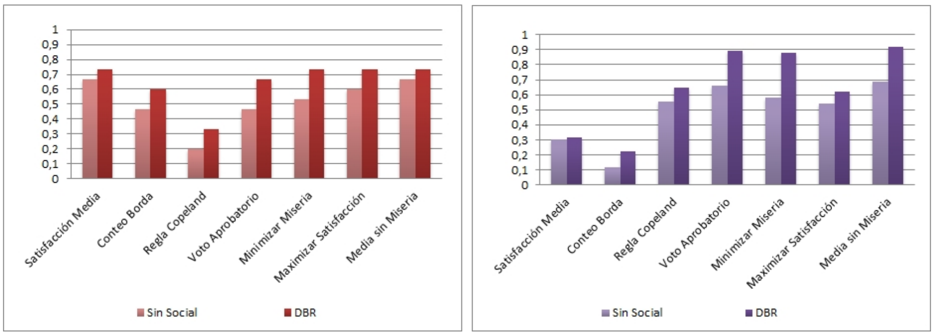


Figura 4.5: Resultados de la comparación de uno de nuestros *Métodos Sociales* (*DBR*) con un *Método Standard* (*SinSocial*) para las diferentes funciones de agregación estudiadas. Usuarios reales (izq) y usuarios sintéticos (dcha).

4.11. Conclusiones

En este capítulo hemos presentado nuestro *MRS*, cuyo enfoque general consiste en modelar sistemas sociales de recomendación grupal captando los comportamientos sociales existentes en un grupo dentro de las técnicas de recomendación grupal. Para poder captar los diferentes aspectos sociales dentro de las dinámicas grupales el modelo utiliza diferentes factores sociales. Hasta ahora, hemos analizado e incluido los 5 factores sociales presentados en el capítulo anterior: *personalidad*, *confianza*, *homofilia*, *persuasividad* y *justicia*. Sin embargo, nuestro modelo, representado por la Ecuación 4.2 permite la inclusión de más factores sociales, hecho que dejamos como tarea para un trabajo futuro. Esta ecuación, Ecuación 4.2, define una *FuncionSocial* que combina los factores sociales estudiados con técnicas de recomendación grupal. En este capítulo hemos presentado 7 métodos que implementan esta *FuncionSocial*: *DBR* (Sección 4.2), *IBR* (Sección 4.3), *Coaliciones* (Sección 4.4), *ModelosDistribuidos* (Sección 4.5), *Memoria* (Sección 4.6), *Cold-start* (Sección 4.7) y *CBR* (Sección 4.8). Además, hemos realizado varios experimentos para poder validar estos métodos y nuestro *MRS*. Los resultados de este proceso de evaluación nos han permitido evaluar el *Impacto de los factores y organizaciones sociales en procesos de recomendación grupal* y confirmar tanto la hipótesis general de esta Tesis: ***"La satisfacción real de un grupo de personas respecto a una recomendación grupal no se puede estimar fielmente utilizando una agregación simple de las preferencias individuales de cada uno de sus miembros. La consideración de las personas como entidades sociales que se relacionan permite mejorar la estimación de su satisfacción individual respecto al resultado de la recomendación y, por lo tanto, mejorar la satisfacción global del grupo"***, como la hipótesis de este capítulo: ***"H2: Es posible desarrollar sistemas de recomendación grupal que mo-***

delen el comportamiento social que tienen los grupos de personas mediante la inclusión de factores sociales". Este hecho, la mejora en los resultados cuando se utilizan factores sociales, se puede observar tanto en las Figuras 4.1, 4.2 y 4.3 anteriormente expuestas en este capítulo, como en los artículos (Quijano-Sánchez et al., 2010, 2011a,d,c, 2012b,a; Recio-García et al., 2013; Quijano-Sánchez et al., 2013c, 2014a) (que aquí solo hemos resumido), así como en la Figura 4.5[11].

[11] Esta Figura pertenece a los experimentos llevados a cabo en 22, (Quijano-Sánchez et al., 2014a), donde comparamos tanto para usuarios reales (izquierda) como para usuarios sintéticos (derecha) el comportamiento de nuestro método social *DBR* (explicado en la Sección 4.2) en comparación con un recomendador estándar que no utiliza datos sociales (denominado en la gráfica *Sin Social*) para las distintas funciones de agregación estudiadas. La función de evaluación utilizada es la denominada *acierto@3* (explicada en la Sección 4.2 y en el Capítulo 2, Sección 2.1.2.2).

Capítulo 5

Arquitectura genérica y metodología de desarrollo para la instaciación del modelo

5.1. Introducción

En anteriores capítulos hemos explicado cómo las redes sociales almacenan información personal de sus usuarios y cómo a través de ellas, los usuarios interactúan con otros usuarios y el sistema (comentan en los muros, realizan peticiones de amistad, valoran artículos, etc.). Estas fuentes de conocimiento se pueden usar para mejorar las técnicas de recomendación y desarrollar nuevas estrategias que se centren en recomendaciones sociales. En los Capítulos 3 y 4 hemos presentado nuestras ideas de cómo mejorar las recomendaciones grupales añadiendo factores sociales como la personalidad de cada individuo del grupo, la confianza existente entre miembros de un grupo o la justicia a largo plazo. Además hemos probado que esta inclusion de información social en los procesos de recomendación grupal mejora la precisión de las recomendaciones (como se vió en la Sección 4.10).

En este capítulo planteamos una nueva hipótesis: ***"H3: Es posible generalizar nuestro*** MRS ***de forma que sea aplicable a diferentes dominios y de forma que otros desarrolladores de sistemas de recomendación sean capaces de reutilizarlo"***. Para validarla trazamos dos nuevos objetivos:

(1) Abstraer nuestro *Modelo de Recomendación Social* (MRS), para ello generalizaremos los métodos de recomendación social indicando los pasos que se deben seguir a la hora de realizar una recomendación social ya sea grupal o individual (nótese que en este objetivo no nos centramos únicamente en recomendaciones grupales como hemos hecho anteriormente sino que generalizaremos nuestro método a cualquier tipo de recomendación so-

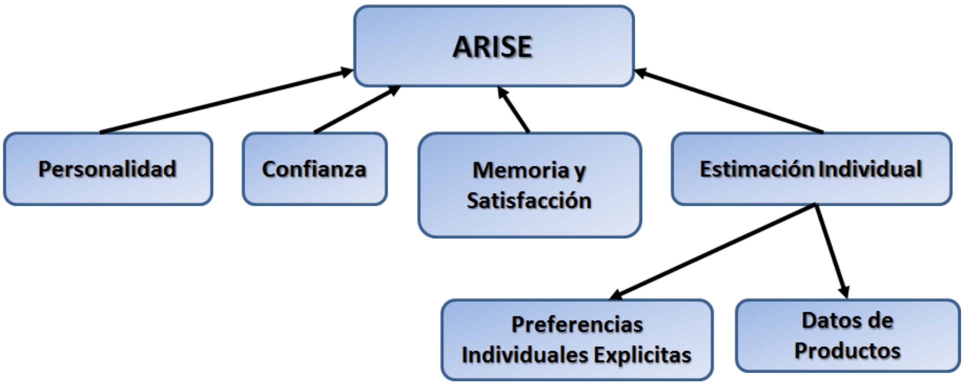


Figura 5.1: Visión general de ARISE.

cial). El resultado de este objetivo es una recopilación de nuestras técnicas y algoritmos en una arquitectura genérica llamada ARISE.

(2) Crear, compartir y validar una colección de plantillas que faciliten el trabajo de los desarrolladores y les permita crear recomendadores sociales de un modo casi automático.

5.2. Arquitectura genérica: ARISE

La abstracción de nuestro MRS en una arquitectura genérica reutilizable que sirva para la instanciación del MRS propuesto en recomendadores concretos de diferentes dominios es lo que llamamos ARISE[1]. El factor clave y común en todos los tipos de recomendadores que se pueden construir en diferentes dominios siguiendo esta arquitectura genérica es la inclusión de factores sociales. Estos factores sociales, que en nuestro MRS son principalmente la personalidad y la confianza (véase Capítulo 3), definen a cada persona como un componente potencialmente influenciable de una comunidad social o grupo determinada por su entorno, en la mayoría de los casos determinada por las redes sociales a las que pertenece. En nuestro MRS, hemos simulado el comportamiento social basándonos en la idea de que las relaciones entre individuos y su entorno (o red) de personas influyen directamente en sus decisiones (Christakis y Fowler, 2011).

En las aportaciones presentadas en los Capítulos 21, (Quijano-Sánchez et al., 2013b, 2014c) y 22, (Quijano-Sánchez et al., 2014a) presentamos la arquitectura de ARISE (Figura 5.1). La arquitectura se divide en 7 módulos diferentes: personalidad, confianza, memoria y satisfacción, estimación de preferencias individuales, preferencias individuales explícitas, datos de productos y el propio modulo ARISE. A continuación resumimos la funcionalidad de cada módulo (explicado en detalle en los artículos (Quijano-Sánchez et

[1]en inglés, *Architecture for Recommenders Including Social Elements.*

al., 2013b, 2014c,a)):

Módulo de personalidad: Este módulo es el encargado de obtener un valor que represente la personalidad de cada usuario. Este valor de la personalidad, p_u, está delimitado por un rango de (0,1], donde 0 refleja una personalidad cooperativa y 1 refleja una muy egoísta.

Módulo de confianza: Este módulo es el encargado de obtener un valor que represente el nivel de confianza, $t_{u,v}$, entre cada usuario u y v que pertenezcan a un entorno social común o grupo. Nótese que $t_{u,v} \in (0,1]$, donde 0 refleja una muy baja confianza entre usuarios y 1 una muy alta.

Módulo de memoria y satisfacción: Este módulo guarda todas las recomendaciones que se han realizado para cada usuario y/o grupo. Esto nos permite tener un sistema que evita la repetición de recomendaciones y que también asegura un cierto nivel de justicia a la larga. Creemos que este es un paso necesario cuando se proporciona una colección a lo largo del tiempo de recomendaciones "justas". De este modo, si un usuario acepta una recomendación en la que no estaba muy interesado, la próxima vez que utilize el sistema éste lo recordara y le dará al usuario algún tipo de prioridad en el proceso de recomendación.

Nótese que los módulos de personalidad, confianza y memoria y satisfacción son los encargados de calcular los factores sociales de *personalidad*, *confianza* y *justicia* vistos en los capítulos anteriores y utilizados en nuestro MRS. Y que los otros dos factores sociales: *homofilia* y *persuasividad*, estudiados en esta Tesis y incluidos nuestro MRS (Secciones 4.4 y 4.5), se computan por medio de los anteriores (Recio-García et al., 2010; Quijano-Sánchez et al., 2011d; Recio-García et al., 2013).

Módulo de estimación de preferencias individuales: Este módulo es el encargado de calcular las predicciones individuales, $\hat{r}_{u,i}$, para cada usuario u y cada producto i en el catálogo de productos a recomendar P. Las predicciones individuales, o recomendaciones, suponen un bloque básico en la arquitectura ya que nuestro método de recomendación consiste en predecir las valoraciones que cada usuario le daría a cada artículo en el catálogo y luego, en el caso de que la arquitectura se use para sistemas de recomendación grupal, estas valoraciones estimadas se agregan para obtener la predicción global para el grupo.

Módulo de preferencias individuales explícitas: Este módulo obtiene información sobre las preferencias del usuario ($r_{u,i}$), paso necesario para poder predecir valoraciones de productos nuevos. Esta tarea consiste comúnmente en obtener valoraciones de un catálogo de productos.

Módulo de datos de productos: Este módulo obtiene el catálogo de productos a recomendar, P.

Módulo ARISE: Este módulo sólo se necesita cuando se utiliza la arquitectura para la construcción de sistemas sociales de recomendación grupal. Su tarea es combinar toda la información recuperada por el resto de módulos y proporcionar la recomendación grupal. Las estrategias de combinación usadas en este módulo representan a los métodos de recomendación social presentados en capítulo anterior.

5.3. Metodología de desarrollo para facilitar la instanciación de sistemas de recomendación social: *Plantillas de Diseño de Recomendadores Sociales*

Después de diseñar los modulos de la arquitectura genérica ARISE, nuestro siguiente objetivo era el diseño de una herramienta que facilite el trabajo de otros desarrolladores de sistemas de recomendación. Para ello, hemos propuesto una herramienta basada en el diseño de plantillas de jCOLIBRI (Recio-García et al., 2014). En la aportación presentada en el Capítulo 21, (Quijano-Sánchez et al., 2013b, 2014c), hemos creado una colección de *plantillas sociales* que representan un paso intermedio entre ARISE y cualquier aplicación social que se pueda construir siguiendo su estructura. Nuestra propuesta tiene un enfoque CBR (véase Capítulo 2, Sección 2.1.1.2 y Figura 2.1), donde los casos son sistemas previamente diseñados. Cuando un diseñador quiere crear una nueva aplicación de recomendación social sólo tiene que recuperar un sistema similar que ya haya sido creado (nuestras plantillas). Lo reutiliza, y lo revisa para comprobar que todo lo que necesita está en las plantillas pudiendo añadir nuevas facetas.

Nuestras plantillas están formadas por *tareas* que representan los diferentes pasos que los desarrolladores deben tomar cuando diseñan un nuevo sistema de recomendación social. Estas *tareas* ayudan a los desarrolladores y facilitan su trabajo en el diseño de un nuevo sistema. Además, cada *tarea* tiene asignados diferentes *métodos* que la resuelven y representan una implementación concreta de realizarla. Estos *métodos* facilitan y aceleran el trabajo de implementación de los desarrolladores[2]. En la Figura 5.2 se muestran nuestras *plantillas sociales genéricas*, decimos genéricas porque de ellas se pueden obtener diferentes implementaciones que representan una instaciación final del modelo que proponemos. Estas plantillas están compuestas por *tareas genéricas* y *tareas simples*. Las *tareas genéricas* encapsulan secuencias de *tareas simples*. Dependiendo de la descomposición de cada *tarea genérica* en secuencias de *tareas simples*, obtenemos distintas *plantillas finales* que representan una instaciación concreta de nuestro modelo. En la Figura 5.2 podemos ver como hay dos plantillas distintas, una llamada *plantilla preciclo* y la otra llamada *plantilla ciclo*, esta división se ha realizado siguiendo

[2]Estos conceptos, referentes a *tareas* y *métodos*, están basados en los métodos de *Modelado del nivel de conocimiento* (del inglés *Knowledge level modelling*) (Aamodt y Plaza, 1994).

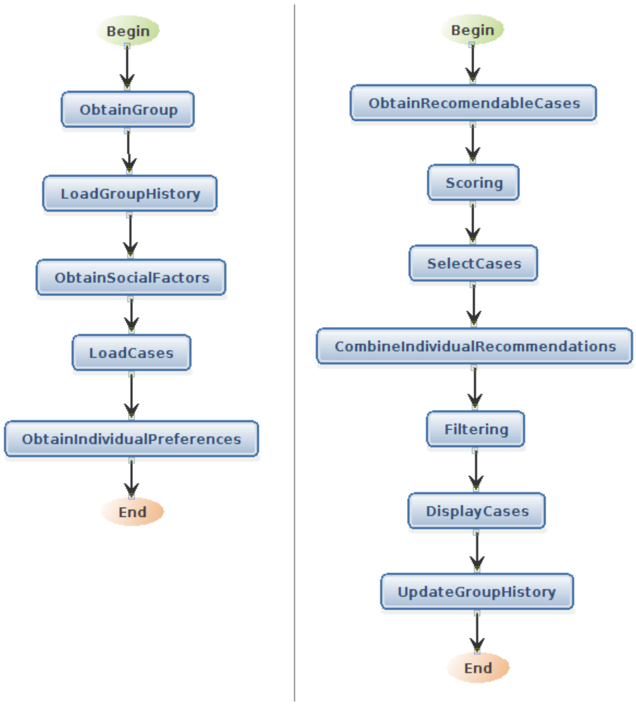


Figura 5.2: *Plantillas sociales*: *plantilla pre-ciclo* (izquierda) y *plantilla ciclo* (derecha).

la arquitectura de jCOLIBRI (Díaz-Agudo et al., 2007), donde el *pre-ciclo* carga los recursos y el *ciclo* ejecuta el ciclo CBR.

Si observamos la Figura 5.2 y la Figura 5.1, podemos ver que algunas de las diferente *tareas* de las plantillas corresponden directamente con los módulos de ARISE. En las plantillas, cada *tarea* se puede implementar a través de diferentes *métodos*. La mayor parte de ellos están ya implementados y por tanto son reutilizables, facilitando así el trabajo de otros desarrolladores que deseen construir un sistema de recomendación mediante el uso nuestras *plantillas sociales*.

A continuación, exponemos brevemente cada una de las *tareas* en las que se dividen nuestras plantillas y algunos posibles *métodos* que se pueden utilizar para instanciar dichas tareas. Nótese que estos se encuentran totalmente detallados en (Quijano-Sánchez et al., 2013b, 2014c). Primeramente nos centraremos en describir la Figura 5.2 parte izquierda, que muestra lo que denominamos la *plantilla pre-ciclo*, formada por las siguientes *tareas*:

ObtainGoup(=ObtenerGrupo) Consiste en obtener el *id* para cada usuario $u \in G_a$, siendo $G_a = \{u : 1 \ldots n\}$ el grupo de usuarios activo y $|G_a| > 1$. G_a (definido anteriormente en el Capítulo 4, Sección 4.1 para los sistemas sociales de recomendación grupal) representa a un grupo de personas que planean realizar una actividad conjunta. Como ARISE y las

Plantillas de Diseño de Recomendadores Sociales han sido diseñadas como herramientas no sólo para el diseño y desarrollo de sistemas sociales de recomendación grupal pero también para sistemas de recomendación individual, es necesario introducir una definición de G_a también para sistemas sociales de recomendación individual. Esta definición se entiende como las personas que pertenecen al círculo de confianza dentro del entorno social del usuario que está recibiendo la recomendación. En ambos casos, el grupo se define dentro del marco de las redes sociales. Algunos de los diferentes *métodos* que se pueden utilizar para obtener el G_a son:

- A través de la creación de un evento que permita organizar actividades conjuntas (para sistemas sociales de recomendación grupal).
- Calculando el grupo de amigos más cercanos dentro de la red social (para sistemas sociales de recomendación individual). Para ello, se utilizará el método que calcula la confianza (como se verá en la *tarea ObtenerFactorDeConfianza*) con todos los amigos del usuario en Facebook que también usen el sistema que se está implementando.

LoadGroupHistory(CargarHistorialDelGrupo) Corresponde con el módulo de *Memoria y Satisfacción* en ARISE. Supongamos una base de casos CB donde cada caso $c \in CB$ guarda un evento de una recomendación pasada. Esta *tarea* consiste en recuperar el caso c que corresponde con el usuario u o el grupo de usuarios G_a. Nótese que esta *tarea* es opcional y que puede ser saltada en caso de que los desarrolladores del nuevo sistema no deseen construir un sistema con memoria de recomendaciones pasadas.

ObtainSocialFactors(ObtenerFactoresSociales) Consiste en una *tarea genérica* que encapsula las siguientes *subtareas*:

- **ObtainUsersPersonality(ObtenerPersonalidadDeUsuarios)**: Corresponde con el módulo de *Personalidad* en ARISE. Consiste en obtener la personalidad de cada usuario u. Como recordamos, la idea de incluir la personalidad de cada usuario, denotada p_u, fue introducida en el Capítulo 3, Sección 3.2 y aplicada en los diferentes métodos del MRS. Para obtener este factor, los usuarios deben completar un test de personalidad al registrarse en el sistema de recomendación. Esta *tarea* se puede completar con uno de los siguientes *métodos*:
    - El Thomas-Killmann Conflict Mode Instrument (TKI) (Thomas y Kilmann, 1974) que plantea 30 situaciones donde el usuario tiene que expresar como reaccionaría. (Por ejemplo, este *método* fue utilizado en en (Quijano-Sánchez et al., 2010))
    - Metáfora TKI de películas alternativa al test, que consiste en mostrar dos personajes de películas conocidos con personalidades opuestas dentro de cinco posibles categorías (aspectos de la

personalidad). Un personaje representa las características de una categoría mientras que el otro representa las opuestas. El usuario debe elegir con cual de los personajes se siente mas identificada/o. Para ello moverá una flecha indicando su grado de semejanza con el personaje. (Este método se utiliza en *HappyMovie* (Quijano-Sánchez et al., 2014b) y en *HappyShopping* (Quijano-Sánchez et al., 2013b) como veremos en el próximo capítulo, Secciones 6.2 y 6.6).

- Cualquier otro test de personalidad del que se pueda extraer el factor p_u como una variable numérica de rango $(0, 1]$, donde el 0 representa una personalidad cooperativa y el 1 una personalidad egoísta.

- **ObtainTrustFactors(ObtenerFactoresDeConfianza)**: Corresponde con el módulo de *Trust* en ARISE. Consiste en obtener la fuerza del vínculo, nuestro factor de confianza $t_{u,v}$, entre los usuarios u y v ($u \neq v \in G_a$). Este factor, que mide la cercanía entre dos usuarios, se puede estimar con la distancia dentro de la red social, el número de amigos en común o la duración de la amistad entre otros aspectos. Para poder extraer esta información directamente de las redes sociales (sin tener que realizar cuestionarios que puedan resultar tediosos a los usuarios) se pueden utilizar redes sociales como Facebook, Tuenti o Google+, es decir *Redes sociales Humanas*. Por ejemplo, como mencionamos en el Capítulo 3, Sección 3.3, cuando presentamos la idea de introducir nuestro factor de confianza, en (Gilbert y Karahalios, 2009) identificaron 74 variables de Facebook como posibles predictores de la fuerza del vínculo entre dos personas. Por otra parte, en (Quijano-Sánchez et al., 2014b) presentamos un método (que se encuentra en nuestras plantillas como *método implementado*) para calcular $t_{u,v}$ mediante la extracción automática de las variables e información necesarias directamente de Facebook.

LoadCases(CargarCasos) Consiste en obtener productos i del catálogo del dominio $D = \{i : 1 \ldots m\}$.

ObtainIndividualPreferences(ObtenerPreferenciasIndividuales) Corresponde con el módulo de *preferencias individuales explícitas* en ARISE. Se puede basar en una combinación de datos implícitos, esto es, de acuerdo a los patrones de uso de los usuarios (véase Ardissono et al. (2004); Zimmerman et al. (2004)) o, en datos explícitos, donde el usuario brevemente, y mediante el uso del sistema, especifica sus preferencias al sistema (véase Billsus y Pazzani (1999); Mccarthy et al. (2004); Quijano-Sánchez et al. (2010)). Por ejemplo, un sistema que vende libros puede que recomiende nuevos libros para que el usuario los adquiera basándose en libros que haya mirado o comprado en el pasado (estos serían ratings implícitos), o en valoraciones que activamente ha indicado el usuario (estos serían ratings explícitos). El *método implementado* que proporcionamos para esta *tarea* consiste en obtener los ratings $r_{u,i}$ que cada usuario u en G_a le asigna a los productos i en D. Estos ratings están en una escala Likert, siendo $1 =$ terrible y $5 =$ excelente.

Continuamos las explicaciones de las plantillas con la otra plantilla creada, la *plantilla ciclo*, que se muestra en la Figura 5.2 parte derecha. Esta plantilla se diseñó principalmente para su uso en la creación de sistemas sociales de recomendación grupal, sin embargo, se puede utilizar también para la implementación de sistemas sociales de recomendación individual dejando las ultimas cuatro *tareas* sin implementar. La *plantilla ciclo* está formada por las siguientes *tareas*:

ObtainRecommendableCases(ObtenerCasosARecomendar) Corresponde con el módulo de *datos de productos* en ARISE. Consiste en obtener todos los productos candidatos a recomendar i del catálogo de recomendación $P = \{i : 1 \ldots p\}$. Por ejemplo, para *HappyMovie*, como veremos en el próximo capítulo, hemos construido un *Rastreador Web* que analiza una web de ocio[3] y recupera todas las películas y sesiones que se proyectan en los cines de España. En esta plantilla, proporcionamos este *Rastreador Web* como *método* que implementa esta *tarea* ya que se puede fácilmente adaptar a otros dominios relacionados con el ocio que ofrece este web, como restaurantes, teatros, conciertos o museos por ejemplo[4].

Scoring(Clasificación) Corresponde con el módulo de *estimación de preferencias individuales* en ARISE. Consiste en obtener ratings predichos $\hat{r}_{u,i}$ para cada usuario activo $u \in G_a$ y cada producto candidato $i \in P$. Algunos de los diferentes *métodos* que se pueden utilizar para implementar esta *tarea*, son los métodos tradicionales de recomendación individual vistos en el Capítulo 2, Sección 2.1.1, o el método de recomendación social basado en influencia visto en el capítulo anterior, Sección 4.3:

- *Recomendadores colaborativos* (Ekstrand et al., 2011; Koren y Bell, 2011; Herlocker et al., 2002).
- *Recomendadores basados en contenido* (Lops et al., 2011).
- *Recomendadores híbridos* (Burke, 2002).
- *Pedir a otros usuarios en G_a que proporcionen un rating estimado para i* (Costello et al., 2006), este método se apoya en feedback explícito de diferentes características en productos por parte de los usuarios.

[3] http://www.guiadelocio.com/

[4] Somos conscientes de que esta opción está limitada a las ofertas de ocio en España, pero creemos que seria fácilmente adaptable a otras webs de ocio de otros países. Por ello, hemos incluido el *Rastreador Web* como un *método* posible que implemente esta *tarea*.

- *Recomendadores basados en influencia* (Quijano-Sánchez et al., 2013c, 2010), que modifican las predicciones no sociales $\hat{r}_{u,i}$ obtenidas con alguno de los *métodos* anteriores con los factores de *personalidad* y *confianza* calculados en *tareas* anteriores. Este método, detallado en el capítulo anterior: IBR (Sección 4.3, ecuación 4.5), se utiliza en el caso de uso de estas plantillas, en las recomendaciones de *HappyShopping*, como veremos en el próximo capítulo, Sección 6.6.

SelectCases(CasosSeleccionados) Consiste en seleccionar para cada usuario activo $u \in G_a$ los k productos en P cuyos ratings predichos $\hat{r}_{u,i}$ sean los más altos. Por ejemplo, en *HappyShopping* (Sección 6.6), utilizamos $k = 4$. Nótese que las próximas tres *tareas* son específicas para sistemas sociales de recomendación grupal, por tanto, el *método* que implementa esta *tarea* necesitará tener una opción de *mostrar casos* si se está implementado un sistema de recomendación individual.

CombineIndividualRecommendations (CombinarLasRecomendacionesIndividuales) Corresponde con el módulo ARISE en ARISE. Consiste en obtener una predicción para el grupo, $\hat{r}_{G_a,i}$, mediante la agregación los ratings estimados para cada miembro del grupo, $\hat{r}_{u,i}$ para cada $u \in G_a$ y $i \in P$ (véase la Ecuación 4.2 del capítulo anterior). Los *métodos* a utilizar para implementar esta *tarea* son los presentados en el capítulo anterior, el DBR (Sección 4.2, Ecuación 4.3), el IBR (Sección 4.3, Ecuación 4.5) o los basados en memoria (Sección 4.6) entre otros.

Filtering(Filtrado) Consiste en seleccionar los k' productos de P que tengan los ratings grupales predichos más altos. Por ejemplo, en *HappyMovie* como veremos en el próximo capítulo, utilizamos $k' = 3$.

DisplayCases(MostrarCasos) Consiste en mostrar a cada usuario u recibiendo la recomendación los productos k' propuestos por el recomendador grupal.

UpdateGroupHistory(ActualizarHistorialDelGrupo) Corresponde con el módulo de *Memoria y Satisfacción* en ARISE. Consiste en en revisar el caso c que corresponde con el usuario activo u (en los sistemas recomendadores individuales) o con el grupo activo G_a (en los sistemas recomendadores grupales) con la nueva recomendación y reteniéndolo en la base de casos CB para recomendaciones futuras. Nótese que esta *tarea* es opcional y que puede ser saltada en caso de que los desarrolladores del nuevo sistema no deseen construir un sistema con memoria de recomendaciones pasadas.

5.4. Conclusiones

En este capítulo hemos validado nuestra tercera hipótesis: ***"H3: Es posible generalizar nuestro MRS de forma que sea aplicable a diferentes dominios y de forma que otros desarrolladores de sistemas***

de recomendación sean capaces reutilizarlo". Para ello, hemos diseñado nuestra arquitectura genérica ARISE. ARISE es una organización teórica de los componentes que se necesitan para construir recomendadores sociales de acuerdo a nuestro MRS. El factor clave y común en todos los tipos de recomendadores que se pueden construir en distintos dominios siguiendo esta arquitectura genérica es la inclusión de factores sociales. En el próximo capítulo presentaremos dos casos de uso basados en la arquitectura ARISE. Estos casos de uso se han construido en diferentes dominios, películas y ropa, y tienen diferentes objetivos, uno de ellos es una aplicación social de recomendación individual (*HappyShopping*, Sección 6.6) mientras que el otro caso de uso es una aplicación social de recomendación grupal (*HappyMovie*, Sección 6.2). Los diferentes objetivos que presentan estos dos casos de uso y los diferentes dominios en los que se basan nos permitirán concluir que ARISE es efectivamente una arquitectura genérica válida para construir recomendadores social en distintos dominios.

Después de diseñar ARISE, nuestro siguiente objetivo para validar esta tercera hipótesis ha sido diseñar una herramienta que facilitase el trabajo de otros desarrolladores de sistemas recomendadores. Hemos creado una colección de *Plantillas de Diseño de Recomendadores Sociales* que son una metodología de diseño software y representan un paso intermedio entre ARISE y cualquier aplicación social que se pueda construir siguiendo su estructura. Para ello, proponemos un enfoque CBR, donde cuando un desarrollador quiera construir una nueva aplicación de recomendación social sólo ha de recuperar un sistema similar previamente diseñado (nuestras plantillas). El desarrollador podrá reutilizar este sistema previamente diseñado y revisar que todos los requisitos del sistema se encuentren en estas plantillas (i.e. casos). En caso de que se necesite adaptar el sistema automáticamente generado (tras la selección de los *métodos* proporcionados ya implementados) para poder cubrir toda la funcionalidad del nuevo sistema, las plantillas pueden usarse como sistema base donde añadir nuevos *métodos*. En el próximo capítulo, presentaremos nuestro caso de uso *HappyShopping* (Sección 6.6), y detallaremos un experimento donde se le pidió a 3 desarrolladores que usaran nuestras *Plantillas de Diseño de Recomendadores Sociales* y desarrollaran una nueva aplicación social de recomendación. Más tarde, reportarán si preferían tener las plantillas para ayudarles, en cuyo caso podríamos concluir que nuestras *Plantillas de Diseño de Recomendadores Sociales* son efectivamente útiles para la comunidad de recomendadores, o, si en cambio no consideraban que nuestras *Plantillas de Diseño de Recomendadores Sociales* facilitaran y aceleraran su trabajo.

Capítulo 6

Pruebas de concepto en una red social

6.1. Introducción

Como comentamos en anteriores capítulos los factores sociales (como son la *confianza* o la *personalidad*) son difíciles de estimar sobre todo si se pretende diseñar un sistema de recomendación dinámico y poco intrusivo. Por ejemplo en otros trabajos (Golbeck, 2006b; Avesani et al., 2005) que hacen uso de factores sociales, de la confianza entre usuarios en este caso, se les pide a los usuarios que indiquen explícitamente su confianza con otros usuarios. Abusar de este proceso de extracción por medio de cuestionarios puede resultar tedioso y generar rechazo entre los usuarios. Hoy en día la web colaborativa proporciona una herramienta muy útil para solucionar este inconveniente: las redes sociales.

Las redes sociales permiten a los usuarios interactuar y desarrollar sus relaciones sociales a través de Internet. Por tanto, algunos trabajos han señalado que los elementos sociales pueden ser inferidos de ellas (Mislove et al., 2010; Konstas et al., 2009). Por ejemplo, podríamos estimar la intensidad del vínculo entro dos usuarios midiendo el número de mensajes intercambiados o de amigos en común. Además, las redes sociales se pueden utilizar como entorno experimental donde construir numerosos sistemas de recomendación, gracias a su diseño que facilita la interacción e intercambio de información con los usuarios, por su alcance, hoy en día casi todo el mundo utiliza alguna clase de red social (como se vió en el Capítulo 2, Sección 2.2.1) y por su dinámica de funcionamiento, por ejemplo la creación de eventos para realizar actividades en grupo en Facebook, resulta adecuada para la integración en ellos de recomendaciones grupales.

En el capítulo anterior presentamos una plataforma genérica que permite la reutilización de nuestro *Modelo de Recomendación Social* (*MRS*), ARISE (Architecture for Recommendations Including Social Elements) y un proceso de desarrollo software basado en plantillas que conceptualizan el comportamiento del *MRS* para poder realizar esa reutilización. En este capítulo nuestro objetivo es validar ARISE, las *Plantillas de Diseño de Recomendadores Sociales* y la hipótesis: ***"H4: Es posible validar y evaluar nuestra arquitectura genérica*** ARISE ***por medio de distintas aplicaciones concretas en diferentes dominios"***. Para ello, hemos construido dos casos de uso en la red social Facebook[1]: (1) *HappyMovie*, que es una aplicación social grupal que sigue nuestra arquitectura ARISE e implementa nuestro *MRS*. La construcción de esta aplicación nos proporciona un medio para testear la viabilidad de las extracción semi-automática de factores sociales que propusimos en el capitulo 3, y también como medio para testear los algoritmos sociales propuestos en el Capítulo 4 y las reacciones de los usuarios frente a ellos; (2) *HappyShopping*, que nos sirve para demostrar la validez de nuestro modelo en otros dominios y la utilidad para la comunidad de recomendadores de ARISE, y *Plantillas de Diseño de Recomendadores Sociales* propuestas en el capítulo anterior.

6.2. Prueba de concepto en el dominio de las películas: *HappyMovie*

HappyMovie[2] es una instanciación particular de nuestra arquitectura genérica ARISE, aplicada al dominio de las recomendaciones grupales de películas. Nos sirve también de caso de uso y nos proporciona un entorno de experimentación donde podemos evaluar nuestro *MRS* con usuarios reales.

Para facilitar la obtención del conocimiento social de los usuarios planteamos *HappyMovie* como una aplicación en una red social desde la que nos podemos beneficiar de la información almacenada en ella. Con este paso, movemos nuestras teorías para realizar recomendaciones sociales para grupos probadas en entornos simulados (Quijano-Sánchez et al., 2009, 2010, 2013c), a la instanciación de nuestro modelo en un escenario real: la red social Facebook. Existen varios motivos por los que tomamos esta decisión. En primer lugar, Facebook es una red social que se utiliza entre otras muchas cosas para crear eventos y organizar actividades en grupo, como nuestro objetivo es facilitar la toma de decisiones en grupo nuestro sistema es idóneo para facilitar la organización de este tipo de eventos. En segundo lugar, la actividad social de cada usuario en la red social queda guardada y es fácilmente extraíble, hecho que nuestra aplicación puede utilizar y obtener así información sobre la confianza entre usuarios de forma automática y no intrusiva (evitando así molestar a los usuarios con numerosos cuestionarios, hecho que puede generar rechazo y enlentece el proceso de recomendación). Finalmente,

[1]Facebook tiene mas de 1.19 billón de usuarios activos.
[2]https://happymovie.fdi.ucm.es

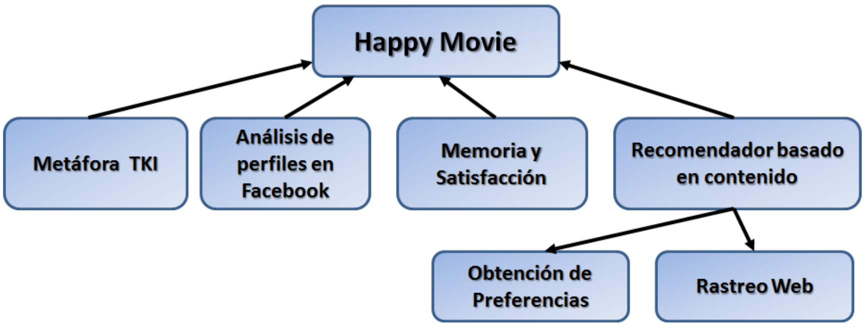


Figura 6.1: Arquitectura de *HappyMovie*

Facebook proporciona la posibilidad de realizar tests, encuestas y juegos sencillos de forma interactiva y dinámica por lo que es un entorno perfecto para obtener la personalidad de los usuarios, que es el último elemento requerido por nuestro modelo. Nótese que aunque consideramos que Facebook es la red más apropiada con la que trabajar, otras redes sociales como Tuenti o Google+, que también están clasificadas como *Redes sociales Humanas* –de las que podemos estimar la fuerza del vínculo entre usuarios–, podrían ser también adecuadas para el desarrollo de este tipo de aplicación.

En las aportaciones presentadas en los Capítulos 11, (Quijano-Sánchez et al., 2011e), 15, (Quijano-Sánchez et al., 2011b) y 23, (Quijano-Sánchez et al., 2014b) explicamos diferentes facetas de *HappyMovie*, que como hemos dicho es una es una instanciación particular de la arquitectura ARISE. La arquitectura de módulos de *HappyMovie* se muestra en la Figura 6.1. En las siguientes secciones describimos cada uno de estos módulos como una instantiation concreta (orientada a una aplicación para Facebook que proporciona una recomendación grupal a un grupo de personas que desean ir juntos al cine) de su correspondiente módulo de alto-nivel en ARISE. Nótese que ARISE fue descrita en la Sección 5.2 y su arquitectura quedó ilustrada en la Figura 5.1.

6.3. Módulos de *HappyMovie*

Como hemos descrito en los anteriores capítulos nuestro *MRS* integra factores sociales para mejorar la estimación y modelado de los procesos de toma de decisiones que siguen los grupos de personas cuando debaten sobre una actividad en común.

Nuestro objetivo con *HappyMovie* es dar un paso más en el diseño de sistemas recomendadores e introducirlos en la web social (definida en la Sección 2.2.1) –concretamente Facebook– donde las relaciones entre usuarios son fácilmente inferibles y por tanto utilizables para mejorar las recomendaciones grupales. Como hemos dicho anteriormente, con este entorno podemos obtener gran parte de la información necesaria para poder aplicar nuestro *MRS* directamente de la información almacenada en una red social. Como ya comentamos, anteriormente (Golbeck, 2006b; Bischoff, 2010) la adquisición de este tipo de información social se realizaba por medio de varios cuestionarios. La integración del sistema en una red social facilita este proceso de extracción y además nos proporciona la posibilidad de obtener el feedback necesario para evaluar y mejorar nuestra propuesta.

En las siguientes subsecciones detallamos los módulos de *HappyMovie*.

6.3.1. Metáfora TKI

Para realizar el cómputo de la personalidad, nuestra aplicación requiere que nuestros usuarios contesten un test de personalidad. Este test es una adaptación del test Thomas-Kilmann Conflict Mode Instrument (TKI) (Thomas y Kilmann, 1974), que como mencionamos en el Capítulo 3, es uno de los instrumentos líderes en la evaluación de la reacción de las personas a la hora de enfrentarse a situaciones conflictivas. Como mencionamos en la Sección 3.2, consiste en 30 situaciones diferentes con dos posibles respuestas. Dependiendo de estas respuestas, al usuario se le asigna una puntuación en cada uno de los 5 tipos de personalidad (ver Figura 3.1) que se organizan en dos dimensiones: *autoritarismo* y *cooperacionismo*. Sin embargo, cuando le preguntamos a nuestros usuarios en Quijano-Sánchez et al. (2011b) qué opinaban de este test, lo calificaron de tedioso y largo. Por ello, para hacer la aplicación más amena estudiamos la posibilidad de usar una metáfora de películas como alternativa al test original. En *HappyMovie* hemos desarrollado esta metáfora alternativa que ameniza esta actividad.

Nuestra metáfora interactiva consiste en mostrar dos personajes de películas con personalidades opuestas representando a cada uno de los cinco tipos de personalidad en la resolución de conflictos. Un personaje representa las características de ese tipo de personalidad, mientras que el otro representa justamente las opuestas. El usuario tiene que mover un flecha para reflejar cuanto se parece a un personaje u otro. En (Quijano-Sánchez et al., 2011b) concluimos que era viable sustituir el test TKI por la metáfora de películas ya que este último test proporciona una estimación bastante precisa del factor de personalidad que se obtenía utilizando el test TKI original (se demostró que el error medio obtenido no era estadísticamente significativo). Este test alternativo mejora significativamente la usabilidad y el interés mostrado hacia la aplicación. En la Figura 6.2 podemos ver una captura de pantalla del test de personalidad de *HappyMovie*.

6.3.2. Análisis de perfiles en Facebook

Este módulo es el encargado de obtener la confianza inter-personal o lazos sociales entre usuarios. Este factor se puede estimar utilizando dife-

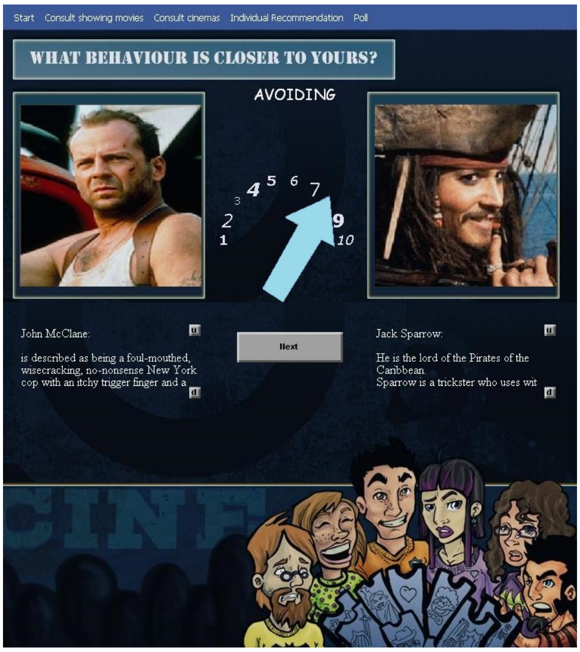


Figura 6.2: Test de personalidad en *HappyMovie*.

rentes enfoques, implicando la mayoría de ellos una extracción manual o largos cuestionarios (Golbeck, 2006a), tarea que nuestros usuarios encontraban tediosa y de la que se resentían. Por tanto, nuestra propuesta se centró en la extracción de este factor directamente de las redes sociales. En esta sección detallamos como calcular la confianza entre dos usuarios de forma automática gracias a la inclusión de la aplicación en una red social. Este proceso consiste en calcular la confianza inter-personal mediante el análisis de los perfiles de usuario y sus interacciones en la red social. En Facebook, sus usuarios publican grandes cantidades de información personal, hecho que utilizamos para calcular la confianza con otros usuarios como son: los gustos e intereses, fotos, juegos, etc.

El uso de la confianza y la utilización de información social extraída de las redes sociales para el desarrollo de sistemas de recomendación no es una novedad (Golbeck, 2006b; Avesani et al., 2005). Es por esto que primero revisamos otros trabajos de expertos en la extracción de información social (Gilbert y Karahalios, 2009; Golbeck, 2006a) para identificar las variables que debíamos analizar. Sin embargo, a la hora de pasar de la teoría a la práctica es importante tener en cuenta que estas variables no son fáciles de cuantificar y que dependemos de la API de la red social. En *HappyMovie*, como se detalla en Quijano-Sánchez et al. (2014b), calculamos la confianza entre dos usuarios u y v $\{t_{u,v} : u, v \epsilon U, u \neq v\}$ como la media ponderada de las siguientes variables: t_1) **Intimidad**, que representa cuanto interactúan los usuarios fuera de la red social. En nuestro enfoque estimamos este valor con el porcentaje de fotos en la red social en las que aparecen juntos; t_2) **Intensidad**, que representa cuanto interactúan los usuarios dentro de la red social. En nuestro enfoque estimamos este valor calculando el número de interacciones dentro de la red social entre los usuarios; t_3) **Duración**, que representa hace cuanto se conocen los usuarios. En nuestro enfoque estimamos este valor con una variable estructural que calcula el número de amigos en común; y t_4) **Servicios Recíprocos**, que representa cuan similares los perfiles de los usuarios en la red social son, en términos de intereses comunes (películas, música, etc). En nuestro enfoque estimamos este valor con el porcentaje de información común publicada en la red social.

El cómputo de la confianza se realiza siempre que un usuario se una a un evento con todos los usuarios que también participan en ese evento. Estos valores no se guardan sino que se calculan constantemente ya que los perfiles de Facebook cambian continuamente al igual que la confianza entre personas.

6.3.3. Memoria y Satisfacción

En *HappyMovie* se guardan todas las recomendaciones que se realizan a cada usuario y cada grupo. Esta función nos permite evitar la repetición de recomendaciones y proporcionar un nivel equitativo de satisfacción dentro de un grupo. Con frecuencia podemos esperar que un grupo reutilize la aplicación varias veces y que por tanto obtenga una amplia colección de recomendaciones. Sin embargo, nuestro MRS tiende a favorecer siempre a los mismos usuarios (ya sea porque tienen personalidades fuertes o porque tienen relaciones más fuertes con el resto del grupo). Consecuentemente, podríamos dar con una situación donde algunos usuarios se sintiesen discriminados e insatisfechos con el sistema al tener sus opiniones menos en cuenta por el bien del grupo. Para poder evitar este tipo de situación donde hay una gran divergencia entre los niveles de satisfacción del grupo, debemos tener en cuenta la satisfacción de los usuarios con las recomendaciones anteriores, ya que sería recomendable para futuras recomendaciones favorecer aquellos usuarios que están menos satisfechos con la recomendación para igualar los niveles de satisfacción. Para resolver este problema *HappyMovie* utiliza nuestra propuesta de recomendación basada en memoria como se detalló en la Sección 4.6.

6.3.4. Recomendador basado en contenido

Para poder predecir el rating que cada usuario le daría a cada película hemos utilizado un recomendador *basado en contenido* (Lops et al., 2011; Pazzani y Billsus, 2007). Hemos elegido este enfoque en lugar de un enfoque *colaborativo* (Ekstrand et al., 2011), ya que las películas a recomendar son las que están en cartelera en el momento y por tanto es difícil tener ratings de ellas. Por esta razón no tenemos suficiente información para operar con sistemas de recomendación colaborativos. Este módulo proporciona al sistema una colección $\{\hat{r}_{u,i} : u \epsilon G_a, i \epsilon P\}$ por cada usuario u en el grupo activo G_a que representa los ratings predichos para cada uno de los productos del catálogo de recomendación P.

6.3.5. Obtención de preferencias

Este modulo implementa un test permite a los usuarios reflejar sus preferencias cinematográficas. Los ratings que aquí se obtienen los utilizará el recomendador individual para estimar las películas que le debe recomendar a cada usuario de acuerdo con su gusto en actores, género, etc. Por ejemplo, digamos que un usuario vota con 3 estrellas una película determinada, como podemos ver por ejemplo en la Figura 6.3, dada esta votación podemos considerar que a esta persona le gusta este tipo de películas, por lo que luego, el recomendador individual analizará las características de la película e intentará encontrar una similar para recomendarle al usuario.

Para completar este test les pedimos a los usuarios que voten al menos 40 películas a través de una escala Likert. Los usuarios pueden regresar a este test y modificar o aumentar sus ratings siempre que quieran. Cuanto más películas voten más preciso será su perfil de usuario y por tanto el recomendador individual conseguirá mejores recomendaciones. Este test devuelve a la aplicación un set de ratings reales $r_{u,i}$ para cada usuario u en el grupo activo G_a y cada producto i en el la colección del test del dominio de las películas D.

6.3.6. Rastreo web

Hemos construido un rastreador web que busca y recupera de la página web *La Guía del Ocio*[3] las películas y las sesiones que se proyectan en el momento en los cines de España. Este rastreador es el encargado de obtener la ficha técnica de las películas en cartelera. Cada característica de la película es un campo de comparación para el recomendador individual. Por ejemplo en nuestro dominio en concreto comparamos el reparto, directores, género que tienen en común las distintas películas. Esta colección de películas, junto con toda su información descriptiva, es el catálogo de películas P que contiene los productos i a ser recomendados.

6.3.7. *HappyMovie*

Este módulo se encarga de combinar toda la información recuperada por el resto de módulos y proporcionar la recomendación grupal. Hemos implementado todos los métodos de recomendación basados en factores sociales

[3]http://www.guiadelocio.com/

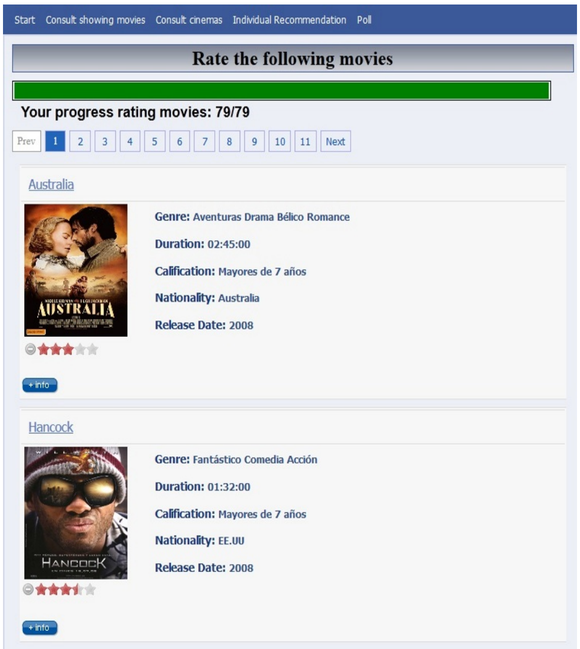


Figura 6.3: Test de preferencias en *HappyMovie*.

presentados en el Capítulo 4 junto con las diferentes funciones de agregación también revisadas en ese capítulo. Esto nos permite utilizar *HappyMovie* como herramienta para testear cualquiera de los métodos propuestos en nuestro *MRS*.

A continuación describimos en líneas generales el funcionamiento de *HappyMovie*.

6.4. Descripción funcional de *HappyMovie*

La página principal de *HappyMovie*(Figura 6.4) ofrece tres actividades diferentes: *realizar el test de preferencias*(test web), *realizar el test de personalidad* y *crear un nuevo evento.* Es necesario completar los test de personalidad y preferencias antes de poder crear nuevos eventos. Complementariamente, los usuarios pueden recibir invitaciones para unirse a eventos ya creados. El usuario tendrá en esta página principal una lista con acceso a todos los eventos en los que participa. Una vez que se ha contestado a los dos test obligatorios los usuarios podrán disfrutar de toda la funcionalidad que *HappyMovie* ofrece (puesto que sus perfiles ya han sido completados y el sistema no necesita más información). Entre las diferentes opciones que pueden elegir está la creación de nuevos eventos, invitar a amigos a eventos ya creados o seguir completando su perfil contestando más preguntas del test de preferencias, entre otras muchas posibilidades:

Figura 6.4: Página principal de *HappyMovie*.

- **Aceptar invitaciones**: Los usuarios pueden aceptar o rechazar las invitaciones pendientes.
- **Crear nuevos eventos**: Los usuarios pueden crear nuevos eventos indicando cuándo y dónde tendrán lugar y la fecha limite para unirse al evento. (Figura 6.5). Como hemos dicho, los nuevos eventos se listarán en la página principal.
- **Eventos**: Esta página (Figura 6.6) muestra toda la información relativa al evento: asistentes, fecha y lugar de celebración, fecha límite para unirse, muro de comentarios, etc. Pero, su función principal es mostrar las tres mejores recomendaciones que el sistema ha encontrado para los asistentes en ese momento. Esta recomendación es tentativa, y se irá actualizando cada vez que la cartelera de la ciudad indicada cambie o cada vez que la configuración del grupo se modifique (ya sea porque se unan nuevos usuarios al evento o se borren).

Cuando se crea un evento todos los asistentes puede invitar a sus amigos a través de una lista que se le proporciona con todos sus contactos de Facebook, también pueden borrarse del evento en cualquier momento. Sin embargo, cuando llega se cierra el plazo de invitaciones estas dos opciones desaparecen y el grupo queda cerrado. En ese momento se mostrará la recomendación final. En este punto los usuarios pueden votar estas tres películas propuestas para poder llegar a un acuerdo en común siendo la más votada la elegida. Está acción nos permite obtener el nivel de satisfacción de cada usuario con las recomendaciones.

A continuación detallamos funcionalidades extra que presenta *HappyMovie*, éstas, son accesibles en todo momento a través de la barra superior que

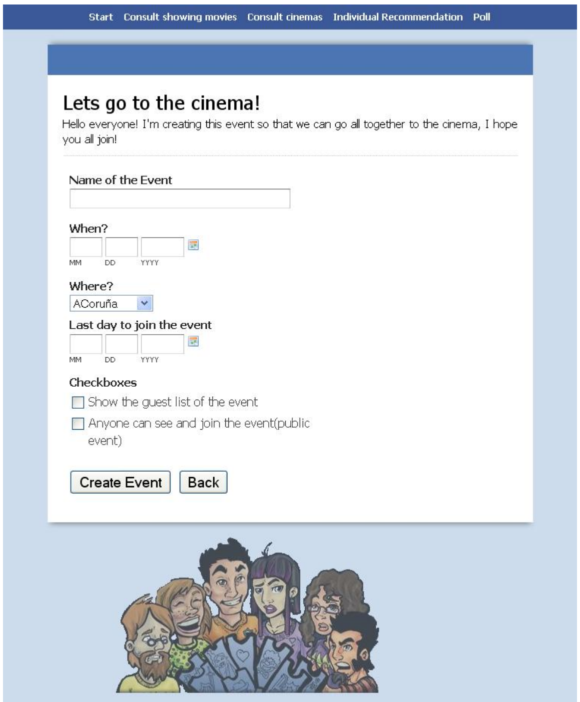


Figura 6.5: Página de creación de eventos en *HappyMovie*.

se encuentra en cada una de las páginas de *HappyMovie* (como podemos ver por ejemplo en la Figura 6.4).

Recomendación Individual: Presenta una lista con las 5 mejores películas que el recomendador individual ha encontrado para una ciudad seleccionada y la cartelera actual.

Consultar películas: Muestra todas las películas que se están emitiendo en una ciudad seleccionada. Resaltaremos que cada vez que se muestra una película en *HappyMovie* ya sea las que se recomiendan, las que están en el test de preferencias, etc. los usuarios siempre pueden ver su título, el poster y algunos datos extras. Adicionalmente, siempre aparecerá un botón de “mas info” que conduce a una página donde se detallan toda la ficha técnica de la película (sinopsis, reparto, nacionalidad, etc) como mostramos en la Figura 6.7. Además, también habrá siempre un botón de “Cines en los que se proyecta esta película” que conduce a una página donde los usuarios pueden ver todos los cines de la ciudad seleccionada donde se proyecta dicha película y qué sesiones tiene.

Consultar cines: Muestra todos los cines de la ciudad seleccionada junto con su localización geográfica y cartelera. Esta página presenta todos los detalles y localización de cada cine a través de *Google Maps*.

Encuesta: Esta página contiene diferentes cuestionarios sobre *HappyMovie*, lo que nos permite obtener feedback de los usuarios para poder conocer

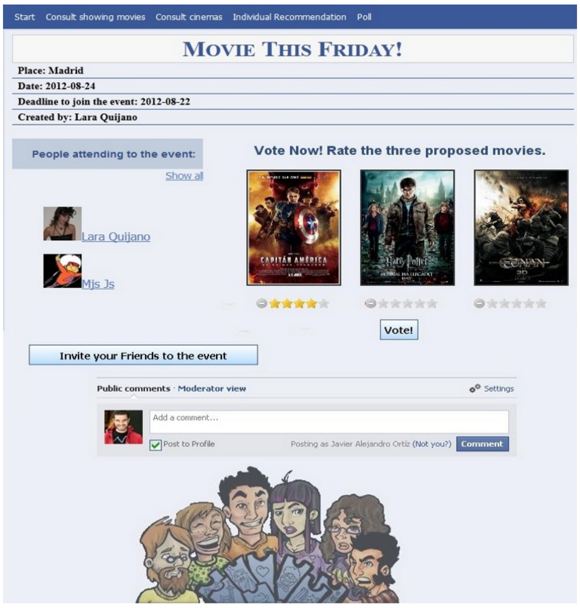


Figura 6.6: Página de eventos en *HappyMovie*.

la satisfacción de nuestros usuarios con las recomendaciones y el sistema en general y así poder mejorarlo. En la siguiente sección detallaremos algunos de los cuestionarios que utilizamos en nuestros experimentos para probar *HappyMovie*.

6.5. Evaluación experimental

Para verificar la eficiencia de nuestro *MRS* y la usabilidad de *HappyMovie*, en la aportación presentada en el Capítulo 23, (Quijano-Sánchez et al., 2014b) hemos llevado a cabo varios experimentos donde 60 usuarios (25 mujeres y 35 hombres) han testeado la funcionalidad de la aplicación. Estos experimentos, que a continuación resumimos pero que se encuentran totalmente detallados en (Quijano-Sánchez et al., 2014b), han consistido principalmente en la realización de una evaluación funcional de la aplicación seguida de un cuestionario donde los usuarios han indicado su percepción de la usabilidad de la aplicación y de las recomendaciones recibidas. Además, les pedimos a estos mismos usuarios que más tarde contestaran un último cuestionario indicando cuales en su opinión eran los aspectos fundamentales que una aplicación que recomienda productos a grupos de usuarios debe cumplir. El objetivo de este último experimento ha sido el estudio de qué deberían intentar mejorar la próxima generación de sistemas recomendadores grupales y cómo obtener feedback de calidad de los usuarios. Los resultados de este estudio centrando en el feedback de los usuarios se pueden encontrar

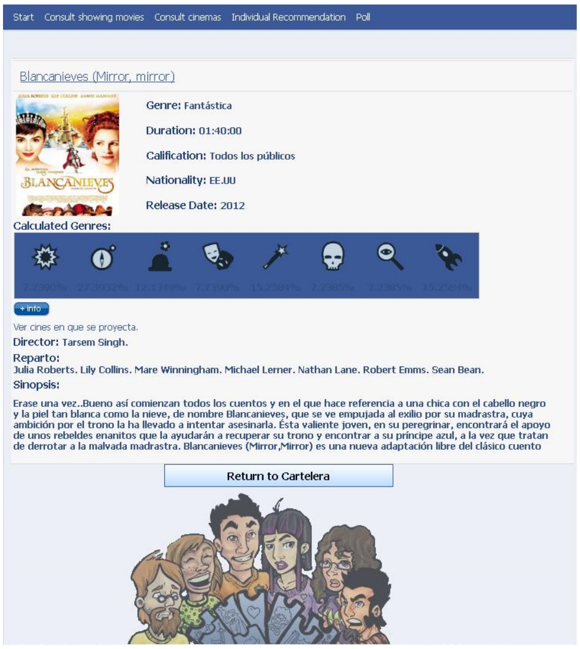


Figura 6.7: Películas en *HappyMovie*.

en la aportación presentada en el Capítulo 20, (Quijano-Sánchez y Bridge, 2013).

Volviendo al cuestionario inicial, en (Quijano-Sánchez et al., 2014b) les pedimos a los usuarios que completasen un test de funcionalidad de *HappyMovie* y que contestasen algunas preguntas. Concretamente les pedimos que siguieran los siguientes pasos:

Paso 1. Contestar el test de personalidad a través de la metáfora de películas (Figura 6.2).
Paso 2. Contestar el test de preferencias (Figura 6.3).
Paso 3. Consultar las películas recomendadas que se presentan en la opción "Recomendación Individual".
Paso 4. Agruparse en grupos de 3 y crear un evento para ir al cine juntos.
Paso 5. Consultar las 3 películas que el *Recomendador Social* ha calculado como las mejores para el grupo.

A continuación, se les pidió a los usuarios de cada grupo que argumentasen y decidiesen si les gustaban y seguirían las recomendaciones grupales ofrecidas por el sistema. Finalmente, se les pidió que contestasen individualmente las siguientes preguntas (con un sistema de 5 estrellas del tipo Likert):

Q1. Utilidad (u): "*Encuentro útil la aplicación (siendo 0 nada útil y 5 muy útil)*".

Q2. Proceso de decisión (dP): "*Es útil porque acelera el proceso de toma de decisiones en grupo (siendo 0 muy poco y 5 mucho)*".

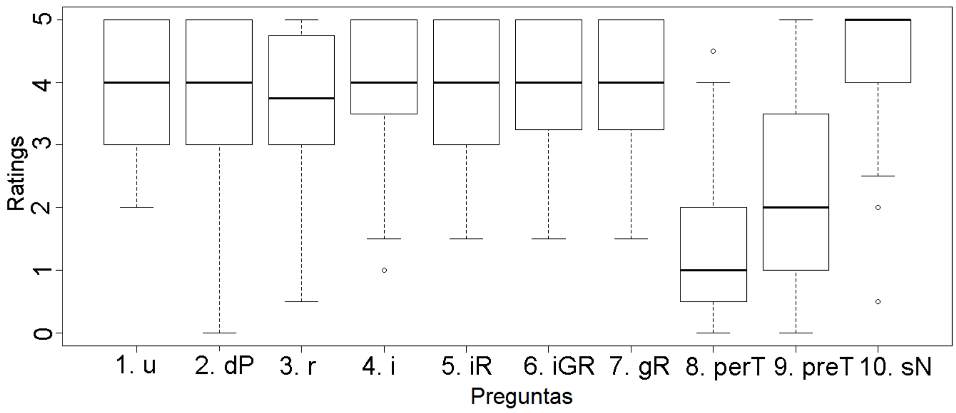


Figura 6.8: Resultados del cuestionario de *HappyMovie*.

Q3. Reusabilidad (r): "*Tengo intención de utilizar la aplicación cuando vaya a ir al cine con mis amigos (siendo 0 muy poco y 5 mucho)*".

Q4. Usabilidad (i): "*La aplicación es intuitiva y fácil de usar (siendo 0 nada intuitiva y 5 muy intuitiva)*".

Q5. Recomendación Individual (iR): "*Me gusta la recomendación individual del sistema (siendo 0 apenas y 5 mucho)*".

Q6. Recomendación Grupal de forma Individual (iGR): "*Individualmente me gusta la recomendación grupal del sistema (siendo 0 apenas y 5 mucho)*".

Q7. Recomendación Grupal (gR): "*Como grupo nos gusta la recomendación grupal del sistema (siendo 0 apenas y 5 mucho)*".

Q8. Test de Personalidad (perT): "*Contestar el test de personalidad ha sido fácil (siendo 0 muy fácil y 5 nada fácil)*".

Q9. Test de Preferencias (preT): "*Contestar el test de preferencias ha sido fácil (siendo 0 muy fácil y 5 nada fácil)*".

Q10. Red Social (sN): "*Considero positivo tener la aplicación dentro de una red social (siendo 0 nada positivo y 5 muy positivo)*".

La Figura 6.8 muestra los resultados del cuestionario que como podemos ver son bastante buenos. Principalmente son buenos porque en general a los usuarios les gusta la aplicación (como vemos en las respuestas 1 u, 2 dP y 4 i), reflejan que tienen intención de usarla más veces (respuesta 3 r) y lo más importante es que piensan que los recomendadores, tanto el individual como el grupal, son el punto fuerte (pues como se ve en las respuestas 5 iR, 6 iGR y 7 gR tienen la media mas alta y la menor desviación) y por tanto son la mejor cualidad de la aplicación. Además, queda reflejado con las respuestas 8 perT y 9 preT que los usuarios no se resienten de los tests de la aplicación, hecho que era uno de nuestros objetivos principales pues queríamos conseguir una aplicación dinámica y fácil de usar para poder incrementar así las

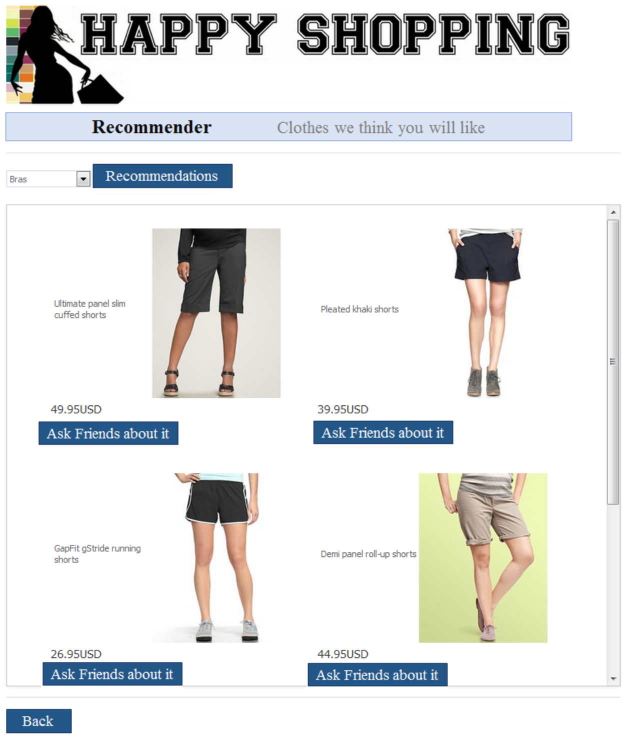


Figura 6.9: Página con recomendaciones en *HappyShopping*.

posibilidades de que los usuarios utilizaran *HappyMovie* con frecuencia.

6.6. Prueba de concepto en el dominio de la ropa: *HappyShopping*

Como comentamos en la introducción de este capítulo, otro de nuestros objetivos ha sido probar la aplicabilidad de ARISE (detallada en la Sección 5.2) en diferentes dominios y la usabilidad de nuestras *Plantillas de Diseño de Recomendadores Sociales* (detalladas en la Sección 5.3). Para ello, en la aportación presentada en el Capítulo 21, (Quijano-Sánchez et al., 2013b, 2014c) realizamos un experimento donde le pedimos a 3 desarrolladores externos que utilizaran nuestras *Plantillas de Diseño de Recomendadores Sociales* y construyesen una nueva aplicación de recomendación social basada en ARISE. Sin embargo, en vez de pedirles que la construyesen en el dominio de las películas (dominio en el que ya habíamos testeado nuestro *MRS* con la aplicación *HappyMovie*) la diseñamos para un nuevo dominio, la ropa. El resultado de este experimento ha sido *HappyShopping*, donde la Figura 6.9 muestra un ejemplo de las recomendaciones que esta aplicación presenta.

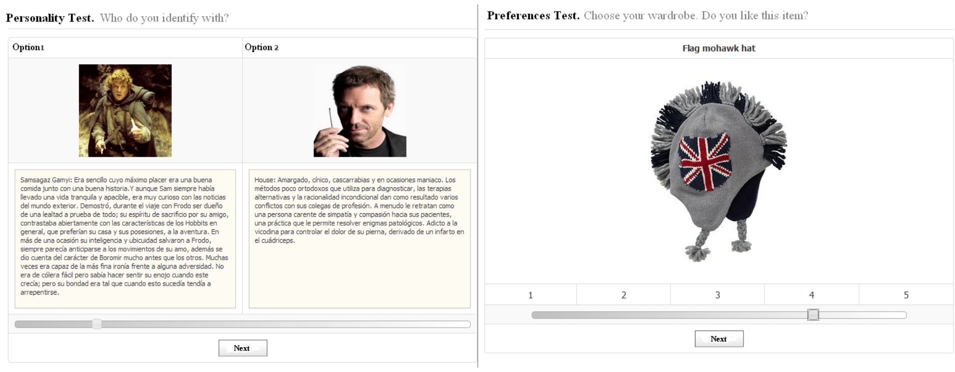


Figura 6.10: Test de personalidad (izquierda) y test de preferencias (derecha) en *HappyShopping*.

HappyShopping es una aplicación de Facebook que recomienda prendas de ropa a un usuario. Es importante destacar aquí la diferencia con *HappyMovie*, que aunque también es una aplicación de Facebook, implementa nuestro modelo para el caso concreto de sistemas grupales de recomendación social. Esta diferencia, el tener dos aplicaciones sociales diferentes siendo una un recomendador individual (*HappyShopping*) y la otra un recomendador grupal (*HappyMovie*), nos permite validar ARISE como una arquitectura social válida que engloba nuestro *MRS*.

Además, *HappyShopping* se basa en el hecho de que antes y/o después de comprar un producto, las personas generalmente escuchan las opiniones de sus amigos. Y que estos amigos, a los que entendemos de confianza, llegan a influenciar la decisión del usuario (este es el enfoque *IBR* visto en la Sección 4.3). Este enfoque es una instanciación concreta del módulo de *estimación de preferencias individuales* de ARISE (véase la Figura 5.1) y la tarea *Scoring* de las *Plantillas de Diseño de Recomendadores Sociales* (véase la Figura 5.2). Primeramente, al igual que en *HappyMovie* y siguiendo nuestro *MRS*, el proceso de recomendación que sigue *HappyShopping* requiere que primero se conteste un test de personalidad. Este test (mostrado en la Figura 6.10 izquierda) sigue el mismo enfoque que el módulo de la *Metáfora TKI* de *HappyMovie* (Sección 6.3.1) e implementa el test TKI que, tal y como se explica en la Sección 3.2 nos permite identificar personalidades asertivas y cooperativas. Nótese que aquí utilizamos esta definición de la personalidad para medir el grado en el que una persona influye o es influenciada por otros. Seguidamente, y de nuevo siguiendo la arquitectura ARISE (el módulo de *preferencias individuales explícitas* esta vez) y siguiendo las *tareas* de las *Plantillas de Diseño de Recomendadores Sociales*, los usuarios deben de especificar qué productos son de su interés. Los productos que el usuario marque de su agradado pasaran a formar parte de su "armario" (véase la Figura 6.10 derecha). Finalmente, la aplicación modela el impacto que las opiniones de

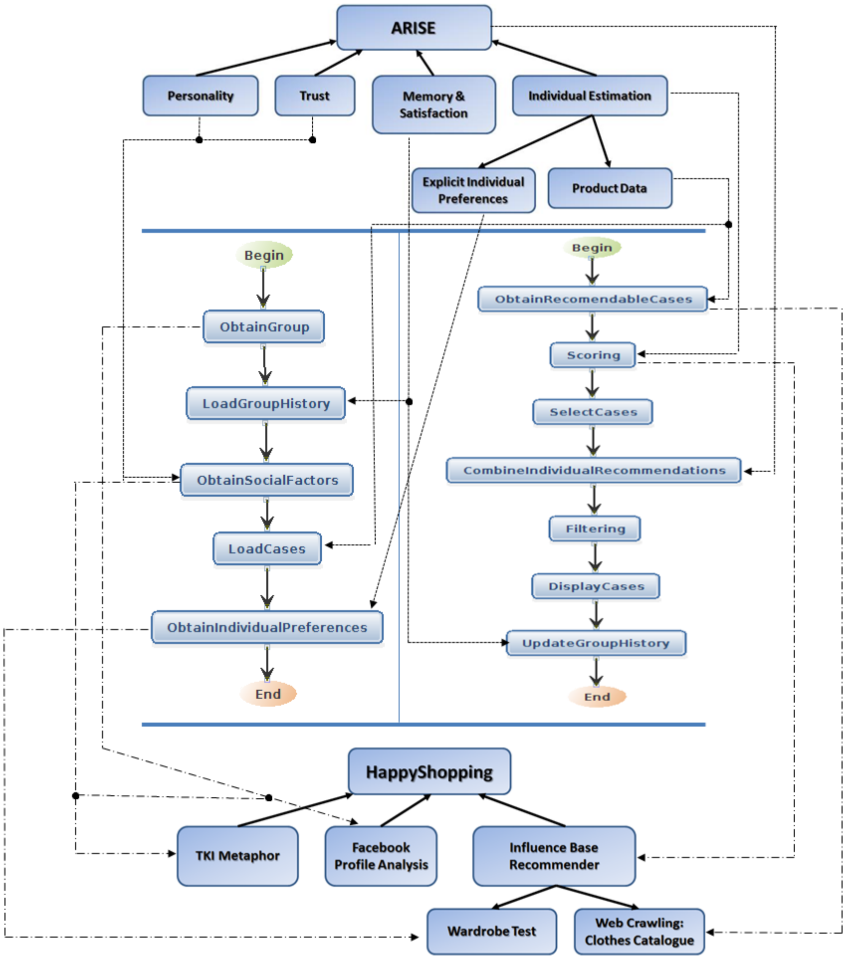


Figura 6.11: Relación entre la arquitectura ARISE, las *Plantillas de Diseño de Recomendadores Sociales* y su instanciación en *HappyShopping*.

las personas más cercanas al usuario en su entorno social y la influencia que éstas pueden tener en su decisión final[4]. Además, valora el grado en el que el usuario puede ser influenciado de acuerdo con su personalidad. Este proceso se realiza a través de la función de recomendación basada en *IBR* explicada en la Sección 4.3 e implementando en la tarea *Scoring* de nuestras *Plantillas de Diseño de Recomendadores Sociales*.

Para poder comprender mejor cómo cada módulo de ARISE define una *tarea* de las plantillas y cómo *HappyShopping* implementa estas *tareas*, incluimos la Figura 6.11. Arriba en la figura podemos ver los módulos de ARISE, cada línea que sale de un módulo señala a la *tarea* concreta en las *Plantillas*

[4]Nótese que esta cercanía se refiere a nuestro factor de confianza y que se implementa del mismo modo que el módulo de *Análisis de perfiles en Facebook* en *HappyMovie* (Sección 6.3.2)

de Diseño de Recomendadores Sociales que la define y cada línea que sale de una *tarea* en las plantillas señala al módulo concreto en la estructura de *HappyShopping* que lo implementa.

Tras finalizar el proceso de construcción de *HappyShopping* hicimos una evaluación informal sobre el coste de desarrollo. Para ello, le preguntamos a sus 3 desarrolladores sobre el esfuerzo y la viabilidad de basarse en la arquitectura de ARISE para el diseño y utilizar las plantillas para la implementación. Las respuestas al cuestionario que realizamos se encuentran en la aportación presentada en el Capítulo 21, (Quijano-Sánchez et al., 2013b, 2014c). En resumen, contestaron que tanto ARISE como las *Plantillas de Diseño de Recomendadores Sociales* les habían facilitado el trabajo y que habían supuesto un gran ahorro de tiempo en implementación y diseño, por lo que preferían tenerlas para asistirles en cualquier nuevo diseño. Para medir el esfuerzo, les preguntamos que cuánto tiempo les había costado implementar una versión inicial de la aplicación, contestaron que 5 semanas la versión inicial y 10 semanas el desarrollo de la versión final de *HappyShopping*. Si comparamos estos resultados con el tiempo que nos costó a nosotros desarrollar e implementar *HappyMovie*,que fue mas de 5 meses, podemos concluir que el uso de las *Plantillas de Diseño de Recomendadores Sociales* y ARISE ha sido un éxito.

6.7. Conclusiones

En este capítulo hemos presentado dos casos de uso: *HappyMovie* y *HappyShopping*. Con ellos hemos podido validar nuestra arquitectura genérica ARISE, al ser capaces de desarrollar dos aplicaciones distintas basándonos en su estructura. Las aplicaciones son distintas en dos aspectos: (1) una se trata de un recomendador social grupal, *HappyMovie*, y la otra de un recomendador grupal individual, *HappyShopping*. Este hecho nos permite concluir que ARISE es una arquitectura válida para los dos tipos de recomendadores existentes, los grupales y los individuales. (2) las aplicaciones han sido desarrolladas en dos dominios distintos, las películas, *HappyMovie*, y la ropa, *HappyShopping*. Este hecho nos permite concluir que ARISE es una arquitectura válida para diferentes dominios. Si bien se han desarrollado los dos casos de uso en dos dominios diferentes, las películas y la ropa, ambos proporcionan productos fácilmente catalogables y valorables. Por ello, creemos que ARISE es una arquitectura válida para muchos otros dominios, como por ejemplo viajes o libros, siempre que el dominio elegido tenga productos con estas características: catalogable y valorable. Estas dos propiedades coinciden con los objetivos de los dos módulos de ARISE que son dependientes del dominio: *Módulo de datos de productos* y *Módulo de estimación de preferencias individuales* (véase Sección 5.2). En cuanto al alcance de la aplicabilidad de ARISE en diferentes redes sociales, en ambos casos de uso nos hemos limitado a realizar pruebas en Facebook, principalmente por motivos de implementación. Además, como dijimos anteriormente Facebook proporciona una plataforma ideal para desarrollar este tipo de aplicaciones que permiten interactuar a varios usuarios creando eventos o completando tests interactivos, para darlas visibilidad por la importancia y la cantidad de usuarios que la utilizan, y para obtener el factor de *confianza* por la información alojada en ella. Como vimos en el Capítulo 2, Sección 2.2.1, Facebook es lo que se denomina una *red social Humana*. Es por ello, que aunque no lo hemos demostrado mediante experimentos y dejamos para un trabajo futuro su verificación, que creemos que la reproductividad de nuestro *MRS* en otras redes sociales queda limitado a redes sociales de este tipo en contraposición a las *redes sociales de Contenidos* (como Twitter), pues como vimos en la Sección 2.2.1 las primeras son de un carácter más íntimo por lo que es posible inferir modelos de confianza y/o contagio emocional.

En este capítulo también hemos podido validar nuestras *Plantillas de Diseño de Recomendadores Sociales*. Para ello, como vimos en la sección anterior y en los artículos (Quijano-Sánchez et al., 2013b, 2014c), realizamos una evaluación informal junto a 3 desarrolladores que utilizaron nuestras plantillas para construir *HappyShopping*. Estos desarrolladores pudieron re-utilizar los *métodos* previamente diseñados (pertenecientes al proceso de desarrollo de las plantillas explicado en el capítulo anterior) y evitarse así empezar el proceso de desarrollo de cero. Como trabajo futuro nos gustaría implementar más *métodos* dentro de las plantillas para que representen a sistemas más heterogéneos y darles más visibilidad, para que más desarrolladores pueden utilizarlas y beneficiarse de ellas.

Finalmente, y de forma más general hemos podido validar también nuestro *MRS*, tras la evaluación positiva que una muestra significativa de usuarios dió al pedirles que realizaran una evaluación experimental de *HappyMovie*, visto en la Sección 6.5 y detallado en (Quijano-Sánchez et al., 2014b).

Estas conclusiones nos permiten poder validar la última hipótesis planteada en esta Tesis: ***"H4: Es posible validar y evaluar nuestra arquitectura genérica*** ARISE ***por medio de distintas aplicaciones concretas en diferentes dominios"***.

A continuación, en el último capítulo de esta Tesis concluiremos el trabajo aquí presentado y presentaremos algunas líneas de trabajo futuro.

Capítulo 7

Conclusiones y trabajo futuro

Me alegro de estar contigo, Samsagaz Gamyi. Aquí, al final de todas las cosas.

Frodo Bolsón

El objetivo perseguido en esta Tesis Doctoral ha sido la mejora de los sistemas de recomendación grupales a través de la inclusión de factores sociales. Para ello, al principio de esta Tesis se formuló la hipótesis de que contrariamente a las técnicas de agregación simples que se vienen aplicando en sistemas de recomendación grupal (Lieberman et al., 1999; Crossen et al., 2002): ***"La satisfacción real de un grupo de personas respecto a una recomendación grupal no se puede estimar fielmente utilizando una agregación simple de las preferencias individuales de cada uno de sus miembros. La consideración de las personas como entidades sociales que se relacionan permite mejorar la estimación de su satisfacción individual respecto al resultado de la recomendación y, por lo tanto, mejorar la satisfacción global del grupo"***. Para probar dicha hipótesis hemos estudiado el "*Impacto de los factores y organizaciones sociales en los procesos de recomendación para grupos*" tras lo que hemos podido concluir que efectivamente la inclusión de factores sociales en los procesos de recomendación de grupos mejora el rendimiento de las técnicas existentes de recomendación grupal. Estas conclusiones están respaldadas por los artículos presentados como principales aportaciones de esta Tesis –Capítulos 8 a 23–.

En este capítulo se resumen los objetivos y aportaciones principales realizadas a lo largo de este trabajo, los cuales nos han permitido poder demostrar la hipótesis de esta Tesis. Finalizamos este documento con algunas líneas de trabajo futuro.

7.1. Conclusiones

Tras la introducción del planteamiento de trabajo realizado en el Capítulo 1 se han verificado las hipótesis planteadas en cada capítulo completando los siguientes objetivos:

- **Hipótesis 1 (H1)**: ***"Existe la necesidad de mejorar los sistemas de recomendación grupal por medio de un modelado más detallado de los procesos de toma de decisiones, posiblemente mediante la inclusión de factores sociales".***
  Objetivo 1 (O1)→ Estudiar la obtención y el uso de los factores sociales en los procesos de recomendación grupal para facilitar la toma de decisiones en grupo: En los Capítulos 2 y 3 hemos estudiado el uso de los factores sociales en los procesos de recomendación grupales para facilitar la toma de decisiones en grupo. El resultado de este estudio se resume en las siguientes aportaciones:

  - **Aportación 1→ Estudio de sistemas recomendadores existentes, incluyendo diferentes técnicas de recomendación individual y grupal**: En el Capítulo 2, hemos estudiado los sistemas recomendadores más representativos. Este estudio ha abarcado las diferentes técnicas de recomendación individual y grupal conocidas hasta el momento y se ha realizado un amplio repaso de los principales sistemas de recomendación grupal.
  - **Aportación 2→ Estudio de factores sociales en los sistemas de recomendación y evaluación de las redes sociales y la información que se puede extraer de ellas**: En el Capítulo 2, hemos estudiado los factores sociales que otros investigadores han incluido hasta el momento en los sistemas de recomendación y la importancia de las redes sociales en los últimos años, los modelos de confianza que se pueden extraer de ellas y diferentes recomendadores que se han diseñado utilizando información alojada en estas redes.
  - **Aportación 3→ Identificación y estudio del comportamiento, respecto a la resolución de conflictos, de las personas en un grupo en función de su personalidad**: En el Capítulo 3 hemos resumido nuestro estudio del comportamiento grupal (respecto a la resolución de conflictos) de las personas en función de su *personalidad*. Durante este estudio se ha propuesto incluir en los procesos de recomendación grupal un factor que simule el comportamiento y las reacciones en situaciones conflictivas de los diferentes miembros del grupo (englobados en un factor que representa la personalidad). Este factor es clave a la hora de personalizar las recomendaciones a las características de cada

grupo. Además, se ha demostrado que el factor de personalidad ayuda a mejorar la simulación que los sistemas de recomendación grupal hacen sobre las argumentaciones y decisiones que tienen lugar en un grupo a la hora de elegir un producto que consumir juntos. Los artículos que cubren esta aportación son los incluidos en los Capítulos 7, (Quijano-Sánchez et al., 2009) y 10, (Quijano-Sánchez et al., 2010).

- **Aportación 4→ Identificación de los factores sociales que influyen en la confianza entre personas y cómo obtenerlos a través de las redes sociales**: En el Capítulo 3 hemos resumido nuestro estudio sobre los factores sociales que influyen en la *confianza* entre personas. Durante este proceso se han estudiado las variables claves para predecir la *fuerza del vínculo* entre usuarios (a.k.a. nuestro factor de confianza), cómo estimar este factor mediante la extracción automática de información almacenada en las redes sociales y cómo utilizarlo en los procesos de recomendación grupal. Los artículos que cubren esta aportación son los incluidos en los Capítulos 10, (Quijano-Sánchez et al., 2010), 18, (Quijano-Sánchez et al., 2013c) y 23 (Quijano-Sánchez et al., 2014b).
- **Aportación 5→ Identificación de factores sociales adicionales que influyen en los procesos de toma de decisiones en grupo**: En el Capítulo 3 hemos presentado también otros factores sociales, la *homofilia*, la *persuasividad* y la *justicia*, que puede influir en los procesos de toma de decisiones en grupo y que por tanto es interesante tener en cuenta a la hora de modelar los procesos de toma de decisiones en grupo. Los artículos que cubren esta aportación son los incluidos en los Capítulos 12, (Quijano-Sánchez et al., 2011d), 19, (Recio-García et al., 2013) y 13, (Quijano-Sánchez et al., 2011c).

- **Hipótesis 2 (H2)**: ***"Existe la necesidad de mejorar los sistemas de recomendación grupal por medio de una modelación mejor de los procesos de toma de decisiones, posiblemente mediante la inclusión de factores sociales".***
  Objetivo 2 (O2)→ Desarrollar nuestro MRS mediante la inclusión de los factores sociales identificados en el objetivo anterior: Tras demostrar la utilidad de incluir factores sociales que nos ayuden a modelar los procesos de toma de decisiones en situaciones conflictivas (Quijano-Sánchez et al., 2009, 2010, 2013c) hemos desarrollado nuestro *Modelo de Recomendación Social* (MRS). En el Capítulo 4 hemos resumido nuestro desarrollo del MRS y los diferentes métodos de recomendación basados en factores sociales que éste incluye. El resultado de este desarrollo se resume en las siguientes aportaciones:

- **Aportación 6→ Propuesta de un método de recomendación basado en delegación, *DBR* (Delegation-Based Recommendations)**: En este método se propone una forma de combinar los factores sociales de *personalidad* y *confianza* de forma que las recomendaciones para cada miembro del grupo estén basadas en las preferencias del resto de componentes del grupo. Los experimentos realizados con este método y resumidos en el Capítulo 4 demuestran que es, de entre todas las técnicas de recomendación social que planteamos, la que mejores resultados obtiene. Los artículos que cubren esta aportación son los incluidos en los Capítulos 12, (Quijano-Sánchez et al., 2011d), 13, (Quijano-Sánchez et al., 2011c) y 18, (Quijano-Sánchez et al., 2013c).
- **Aportación 7→ Propuesta de un método de recomendación basado en influencia, *IBR* (Influence-Based Recommendations)**: En este método se propone una forma de combinar los factores sociales de *personalidad* y *confianza* de forma que las recomendaciones para cada miembro del grupo se vean modificadas en función de la influencia que cada uno del resto de componentes tiene sobre él. El artículo que cubre esta aportación es el incluido en el Capítulo 18, (Quijano-Sánchez et al., 2013c).
- **Aportación 8→ Propuesta de un método de recomendación basado en *coaliciones***: En este método se propone el uso de los factores sociales principales en nuestro MRS, la *personalidad* y la *confianza*, además de uno adicional, la *homofilia*. El método basado en *coaliciones* estudia la identificación de líderes dentro de un grupo y cómo estos líderes pueden intentar formar coaliciones que apoyen sus preferencias. El artículo que cubre esta aportación es el incluido en el Capítulo 12, (Quijano-Sánchez et al., 2011d).
- **Aportación 9→ Propuesta de un método de recomendación basado en modelos distribuidos y argumentación**: En este método estudiamos cómo modelar mediante argumentaciones dinámicas los procesos de toma de decisiones en grupo. Para ello, se propone el uso de sistemas multiagentes con topología de red social, donde cada agente representa a un miembro del grupo. Además, en este método se añade el factor social de *persuasividad* junto a los dos factores sociales principales de *personalidad* y *confianza*. Los artículos que cubren esta aportación son los incluidos en los Capítulos 9, (Recio-García et al., 2010) y 19, (Recio-García et al., 2013).
- **Aportación 10→ Propuesta de un método de recomendación basado en memoria**: En este método se propone un forma de combinar los factores de *personalidad* y *confianza* junto

con el factor social de *justicia* mediante el uso de una memoria de recomendaciones pasadas que evite repetir recomendaciones y ofrezca un recomendador que asegure una satisfacción entre los miembros del grupo homogénea. Los artículos que cubren esta aportación son los incluidos en los Capítulos 13, (Quijano-Sánchez et al., 2011c), 18, (Quijano-Sánchez et al., 2013c) y 23, (Quijano-Sánchez et al., 2014b).

- **Aportación 11→ Propuesta de un método de recomendación para resolver el problema del *cold-start***: En este método se propone el uso de los factores sociales en los sistemas de recomendación grupal para definir medidas de similitud sociales entre usuarios y grupos. Estas medidas se utilizan luego para poder asignar a usuarios que tienen pocas valoraciones –están en *cold-start* y por tanto resulta muy difícil realizar buenas estimaciones para ellos– ratings de los usuarios más similares del grupo más similar. Los artículos que cubren esta aportación son los incluidos en el Capítulo 16, (Quijano-Sánchez et al., 2012b, 2013a).
- **Aportación 12→ Propuesta de un método de recomendación social basado en CBR (Cased-Based Reasoning)**: En este método se propone el uso de factores sociales junto con técnicas CBR para poder definir medidas de similitud sociales entre usuarios y grupos y simular así el comportamiento de los usuarios más similares en los grupos más similares, prediciendo así cómo actuaran los miembros del grupo a recomendar. Este método se presenta como alternativa a los anteriores métodos que presentan ecuaciones predefinidas, contemplando así la posibilidad de que cada ecuación diseñada no sea siempre la idónea para cada tipo de grupo existente. El artículo que cubre esta aportación es el incluido en el Capítulo 17, (Quijano-Sánchez et al., 2012a).
- **Aportación 13→ Evaluación de nuestro *MRS* utilizando las diferentes técnicas de agregación existentes**: Durante este proceso se han estudiado técnicas existentes de agregación simple y se han implementado todas ellas para nuestro *MRS*. A continuación, se ha experimentado con usuarios tanto sintéticos como reales para averiguar cuál es es la técnica de agregación más adecuada en función de diferentes configuraciones grupales (grupos grandes o pequeños) y configuraciones de los recomendadores (sin factores sociales, sólo con el factor de la *personalidad*, sólo con el factor de la *confianza* o con estos dos factores sociales). Los resultados de estos experimentos han demostrado que, en general, la técnica de agregación que mejores resultados tiene en nuestro *MRS* es la *Satisfacción media* y que aquellos recomendadores que incluyen los factores sociales de *personalidad* y *confianza* son

los que mejores resultados obtienen. Los artículos que cubren esta aportación son los incluidos en los Capítulos 14, (Quijano-Sánchez et al., 2011a) y 22, (Quijano-Sánchez et al., 2014a).

- **Aportación 14→ Evaluación de los métodos propuestos**: Durante este proceso se ha demostrado que nuestro MRS mejora el rendimiento de los sistemas de recomendación que no utilizan los factores sociales que nosotros incluimos (Ecuación 4.2). Se han realizado experimentos tanto con usuarios reales como sintéticos, donde se ha probado la eficiencia y precisión de nuestro MRS en el dominio de las películas. Además, se ha probado la viabilidad de utilizar datos sintéticos pues sus resultados son equivalentes a los obtenidos con datos reales. El artículo que cubre esta aportación es el incluido en el Capítulo 22, (Quijano-Sánchez et al., 2014a).

- **Hipótesis 3 (H3)**: ***"Es posible generalizar nuestro*** MRS ***de forma que sea aplicable a diferentes dominios y de forma que otros desarrolladores de sistemas de recomendación sean capaces de reutilizarlo".***
Objetivo 3 (O3)→ Proporcionar una arquitectura genérica y una metodología de desarrollo que permita la instanciación de nuestro *MRS*: Tras presentar nuestro MRS y verificar su relevancia con respecto a la mejora de rendimiento que presenta en las técnicas de recomendación grupal, hemos querido proporcionar una arquitectura genérica y una metodología de desarrollo que permita la instanciación del modelo propuesto. Este desarrollo, que ha sido resumido en el Capítulo 5, ha dado lugar a las siguientes aportaciones:

  - **Aportación 15→ Propuesta de una arquitectura genérica reutilizable,** ARISE: Durante este proceso se ha abstraído nuestro MRS y se ha modulado y organizado de forma que pueda ser reutilizable en otros dominios aparte del ya probado dominio de las películas. El resultado de este proceso es nuestra arquitectura genérica ARISE. Los artículos que cubren esta aportación son los incluidos en los Capítulos 21, (Quijano-Sánchez et al., 2013b, 2014c) y 22, (Quijano-Sánchez et al., 2014a).
  - **Aportación 16→ Instanciación semi-automática de la arquitectura** ARISE **por medio de *Plantillas de Diseño de Recomendadores Sociales***: Durante este proceso se han diseñado una serie de plantillas, denominadas como *Plantillas de Diseño de Recomendadores Sociales*, que representan un paso intermedio entre ARISE y cualquier aplicación social que se pueda construir siguiendo su estructura. Los artículos que cubren esta aportación son los incluidos en el Capítulo 21, (Quijano-Sánchez et al., 2013b, 2014c).

- **Hipótesis 4 (H4)**: ***"Es posible validar y evaluar nuestra arquitectura genérica*** ARISE ***por medio de distintas aplicaciones concretas en diferentes dominios".***
Objetivo 4 (O4)→ Desarrollo de una aplicación para validar nuestro *MRS* **en una red social**: Para validar las aportaciones presentadas en el objetivo anterior se han desarrollado dos casos de uso diferentes uno en el dominio de las recomendaciones de películas para grupos y otro en el dominio de las recomendaciones de ropa para individuos en entornos sociales. Ambos, se han desarrollado en la red social Facebook por motivos prácticos. Estos casos de uso se han resumido en el Capítulo 6 y han dado lugar a las siguientes aportaciones:

    - **Aportación 17→ Desarrollo de una aplicación en la red social Facebook que implementa** ARISE **y recoge las técnicas de recomendación basadas en factores sociales propuestas en el** *MRS*: ***HappyMovie***: Con esta aplicación se ha demostrado la importancia de la utilización de los factores sociales en los procesos de recomendación grupal y la eficiencia de nuestro *MRS*. Con la aplicación como herramienta, hemos podido realizar una evaluación con usuarios reales y validar así las recomendaciones obtenidas y la facilidad de uso de *HappyMovie*. Los artículos que cubren esta aportación son los incluidos en los Capítulos 11, (Quijano-Sánchez et al., 2011e), 15, (Quijano-Sánchez et al., 2011b), 20, (Quijano-Sánchez y Bridge, 2013) y 23, (Quijano-Sánchez et al., 2014b).
    - **Aportación 18→ Desarrollo de una aplicación en la red social Facebook, *HappyShopping*, que demuestra que la arquitectura** ARISE **es viable para otros dominios y que las *Plantillas de Diseño de Recomendadores Sociales* propuestas facilitan el desarrollo de nuevas aplicaciones sociales**: Con esta aplicación se ha demostrado que nuestro *MRS* que inicialmente sólo se había probado en el dominio de las películas es válido para otros dominio. Además, durante el proceso de construcción de esta aplicación se han utilizado nuestras *Plantillas de Diseño de Recomendadores Sociales*, descritas en (Quijano-Sánchez et al., 2013b, 2014c), y la arquitectura ARISE (Quijano-Sánchez et al., 2014a), demostrando así su viabilidad y utilidad. Los artículos que cubren esta aportación son los incluidos en el Capítulo 21, (Quijano-Sánchez et al., 2013b, 2014c).

En resumen, los resultados que hemos presentado a lo largo de esta Tesis Doctoral y que a su vez se encuentran recogidos en los artículos publicados que se presentan como núcleo central de este documento (parte III), sostienen que:

1. Nuestro *MRS* es efectivo y mejora el rendimiento de los métodos de recomendación grupal basados en agregación simple de preferencias.

2. Nuestro trabajo supone un avance sobre el estado del arte.

3. Los métodos que hemos propuesto a lo largo del trabajo realizado son novedosos.

4. Los resultados de los experimentos realizados tanto con usuarios reales como sintéticos son significativos.

5. Los resultados de nuestra investigación han sido presentados a la comunidad científica.

6. El resultado de nuestro trabajo propone una línea de investigación que está siendo seguida por otros investigadores (Gaillard et al., 2014; Kompan y Bieliková, 2014; Leonard, 2014; Christensen y Schiaffino, 2014)(con más de 80 citas, eliminando autocitas, a nuestros trabajos).

A continuación y para concluir, se presentan algunas líneas futuras de investigación.

7.2. Trabajo futuro

Tras finalizar este trabajo de Tesis, proponemos 3 líneas de trabajo futuro que creemos que merece la pena investigar.

7.2.1. Recomendadores sociales grupales adaptativos

A lo largo de los experimentos realizados en esta Tesis Doctoral (Quijano-Sánchez et al., 2010, 2013c, 2014a), hemos podido comprobar que no todas las composiciones grupales son iguales y que no todas las técnicas de recomendación funcionan igual para grupos de diferentes tamaños, distribución de personalidades o de confianza, etc. Por ello, sería muy interesante diseñar recomendadores cuyas técnicas de recomendación se adapten a la configuración del grupo pudiendo así optimizar los resultados obtenidos. Como consecuencia de esto, una futura línea de trabajo sería: *El estudio de las estructuras y composiciones grupales para la utilización de recomendadores sociales grupales adaptativos*. Por recomendadores adaptativos entendemos aquellos que automáticamente elijan un enfoque de recomendación grupal ya sea alguno de nuestros métodos en el *MRS* (*DBR*, *IBR*, etc) o alguna de las técnicas de agregación aplicada dentro de esos métodos (*Satisfacción Media*, *Minimizar la miseria*, etc) en función de las características del grupo pudiendo así maximizar los resultados.

Dentro de esta misma línea de trabajo en recomendadores adaptativos, otro posible paso en la mejora de los sistemas de recomendación grupal sería el análisis del comportamiento grupal en función de la caracterización del grupo, por ejemplo, la distribución de las edades de los miembros del grupo. En el trabajo realizado en esta Tesis Doctoral hemos partido siempre de la premisa de que nuestros grupos de personas estaban realizando actividades junto con amigos. Una situación totalmente diferente serían las recomendaciones grupales a familias, puesto que la diferencia de edad (personas mayores, niños) varía considerablemente las actividades posibles y las prioridades a la hora de satisfacer a los diferentes miembros del grupo. En esta línea serían necesarios recomendadores adaptativos que automáticamente limitaran el conjunto de productos a recomendar a la edad apropiada y además estableciera diferentes pesos y prioridades en función del estudio de como grupos, por ejemplo con niños, suelen comportarse.

Relativo a establecer pesos a cada miembro del grupo, una línea que no se ha trabajado en este trabajo de Tesis, y que sería interesante a desarrollar, es la dependencia del contexto en los grupos. Véase, un grupo de amigos puede que no tenga el mismo comportamiento un día que otro. Ya sea porque un día sea el cumpleaños de alguno de los componentes del grupo en cuyo caso las preferencias de este miembro quizás tengan más peso, porque sea una fecha señalada, por ejemplo en Halloween se suelen ver películas de miedo en cuyo caso este tipo de género tendría más peso, o por el estado emocional de los propios miembros del grupo, pudiendo indicar que no están de buen humor y les apetece ver una comedia.

7.2.2. Recomendadores sociales grupales basados en casos

En el Capítulo 4, Sección 4.8 hemos explicado nuestro estudio de cómo utilizar recomendaciones pasadas para recrear los comportamientos de cada usuario en un grupo, de esta forma evitamos utilizar métodos prefijados que pueden no ser siempre igual de productivos en todas las posibles configuraciones grupales. Una posible línea de investigación futura es continuar con esta idea y estudiar las estructuras grupales (el grafo que forman) y en función de éstas, desarrollar técnicas basadas en casos previos. Esta línea de trabajo futura se definiría como: *El análisis y almacenamiento de comportamientos y recomendaciones grupales para su posterior reutilización en sistemas de recomendación grupal basados en CBR.* Siguiendo esta línea de trabajo, en (Quijano-Sánchez y Bridge, 2013) planteamos como trabajo futuro la necesidad de construir una base de casos (de grupos y las actividades conjuntas que realizan) con casos más completos y detallados, para su futura reutilización[1]. Una estructura de casos más rica podría permitir-

[1] Nótese que hasta ahora no existe ninguna base de casos en la que los casos sean grupos de personas que hayan realizado actividades conjuntamente.

nos capturar múltiples aspectos de los procesos de toma de decisiones. Por ejemplo, la parte descriptiva del caso podría contener por ejemplo algunos o todos de los siguientes datos: (a) información sobre cada miembro del grupo, información demográfica, información acerca de la personalidad, de sus gustos (por ejemplo por medio de ratings); (b) información sobre las relaciones inter-personales entre los miembros del grupo; (c) productos candidatos, esto es, aquellos de entre los cuales el recomendador realiza las recomendaciones; (d) ratings predichos para cada miembro del grupo y cada producto considerado; (e) predicciones sobre otras dimensiones (la experiencia grupal o la satisfacción individual). Por otra parte, la parte de la solución del caso podría contener por ejemplo el producto o productos que se han recomendado, pero podría contener aun más información, como por ejemplo el ranking que el recomendador predijo para cada producto candidato. Además, es bien sabido que los grupos tienden a repetirse (con pequeñas variaciones) así como sus estructuras (como la de unos padres con sus hijos, o la de amigos de la universidad), por tanto una hipótesis basada en CBR (donde los problemas similares tienen soluciones similares) podría ser idónea en este caso.

7.2.3. Explicaciones sociales en sistemas de recomendación grupal

En el trabajo realizado en esta Tesis Doctoral hemos investigado diferentes técnicas de recomendación grupal que integran factores sociales (nuestro *MRS*). Además, se ha realizado un caso de uso de nuestro modelo por medio de una aplicación para ir al cine, *HappyMovie*, que está integrado en la red social Facebook. Como hemos visto a lo largo de este documento, el sistema *HappyMovie* trata de paliar ciertas limitaciones existentes en los sistemas de recomendación grupal, como son la obtención del perfil de los usuarios (que requiere el esfuerzo y el tiempo de los usuarios) o la accesibilidad y presentación de las propias recomendaciones grupales (mediante la introducción del sistema dentro de una red social de uso diario que ayude y facilite a los usuarios en situaciones de conflicto).

Una posible línea de investigación futura sería evaluar el impacto que tienen en los usuarios las explicaciones grupales en sistemas de recomendación social (esto es una propuesta novedosa que aún no se ha estudiado en el área de sistemas recomendadores). Para ello, se incluirían distintos tipos de explicaciones sociales al grupo que recibe la recomendación. Esta línea de trabajo futura se definiría como: *El estudio de explicaciones sociales en sistemas de recomendación grupal.* En estas explicaciones, se intentaría explicar por qué nuestro sistema ha calculado que el producto que se presenta es el mejor para el grupo en general. Más concretamente, se evaluaría la viabilidad y utilidad de incluir tanto explicaciones gráficas como textuales y el impacto que éstas tienen en la aceptación de las recomendaciones o en la confianza en el sistema.

Una vez un sistema realiza una recomendación es natural pensar que los componentes del grupo a recomendar deseen saber en cierto modo cómo se llegó a la recomendación, y en particular cuán de atractiva es dicha recomendación para cada uno de ellos como individuos. Por ello muchos sistemas recomendadores acompañan cada solución con una explicación de la recomendación. Un ejemplo de sistema que utiliza explicaciones para justificar las soluciones propuestas es Let's Browse (Lieberman et al., 1999). Las explicaciones en los sistemas de recomendación presentan múltiples variaciones pudiendo ir desde un simple índice de la confianza del sistema a una visualización compleja de los pros y contras de una solución.

Una aportación adicional al trabajo realizado durante esta Tesis doctoral sería incluir en nuestro MRS un modelo que explique a los usuarios por qué el sistema ha predicho que un determinado producto es la mejor opción para un grupo. Esta técnica es novedosa pues hasta el momento no se había hecho ninguna investigación en el área de explicaciones a grupos. Para ello, primeramente, se realizaría, un estudio del estado del arte relativo a los sistemas recomendadores que realizan explicaciones. Seguidamente, se propondrían diversas alternativas para realizar explicaciones a grupos de personas que reciben una recomendación teniendo en cuenta diferentes aspectos, como las predicciones sobre las preferencias individuales, los factores sociales que forman parte de nuestro MRS. Esta línea de investigación es novedosa en el ámbito de los recomendadores pues pretendemos no solo extender el ámbito de las explicaciones a los recomendadores grupales (pues hasta ahora, los trabajos relacionados en el tema sólo se han centrado en las explicaciones a individuos concretos y no grupos) sino también incluir a los métodos existentes de explicaciones en sistemas recomendadores los factores sociales que en esta Tesis se incluyen de forma innovadora. Este último factor supone un reto aun mayor pues las explicaciones de los factores sociales que se ven envueltos en los procesos de toma de decisiones (y que nuestro MRS utiliza) como pueden ser la *personalidad* de cada individuo o la *confianza* entre usuarios son un tema delicado, donde se juega con la sensibilidad de las personas y sus relaciones, es por ello que se ha de poner un especial interés en producir explicaciones no sólo eficientes y claras sino con tacto. Estas alternativas tendrían entre otros objetivos el incrementar la aceptación de la recomendación recibida así como incrementar la confianza de los usuarios en el sistema.

7.3. Conclusiones finales

A lo largo de este trabajo de Tesis hemos estudiado los sistemas de recomendación, más concretamente los sistemas de recomendación para grupos. Tras identificar algunas carencias que estos sistemas sufren a la hora de modelar el comportamiento social dentro de un grupo, hemos propuesto un modelo –MRS– basado en la inclusión de factores sociales. El modelo está formado por un conjunto de métodos, una arquitectura genérica que lo engloba, unas plantillas que permiten su reutilización y dos aplicaciones que lo instancian. Además, hemos podido demostrar que la utilización de nuestro modelo supone una mejora en los sistemas de recomendación grupal. Queda mucho trabajo por hacer, tanto en el area de los sistemas de recomendación para grupos como en el area de los sistemas de recomendación social, pues son dos líneas de investigación en auge. Sin embargo, esta Tesis Doctoral resuelve con éxito las hipótesis planteadas e identifica nuevas líneas de trabajo futuro. Con ella, cerramos esta etapa de nuestra investigación y comenzamos un nuevo camino.

Part III

Presented Papers

Next, the collection of published papers that reflect the results and investigation process of this PhD Thesis is presented.

Chapter 8

Social Based Recommendations to Groups

8.1 Citation

Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B. Social Based Recommendations to Groups. in: Procs. of the 14th UK Workshop on Case-Based Reasoning. CMS Press, University of Greenwich, pages: 46-57, 2009. ISBN 978-1-904521-64-8. UK-CBR is ranked as CORE C in the CORE ranking.

8.2 Contributions covered by this paper

In this paper we have studied group behaviour in decision making processes. This has led us to the necessity of including social factors such as the personality of each group member and the extraction of the group topology (by analysing the social networks they belong to) to improve group recommenders systems that deal with conflict solving problems.

Social Based Recommendations to Groups

Lara Quijano-Sánchez, Juan A. Recio-García, and Belén Díaz-Agudo

Department of Software Engineering and Artificial Intelligence,
Universidad Complutense de Madrid, Spain
lara.quijano@fdi.ucm.es, jarecio@fdi.ucm.es, belen@sip.ucm.es

Abstract. In this paper we describe some new ideas to improve recommendations to groups. We use distributed CBR systems, where each node collaborates, arguments and counterarguments its local results with other nodes to improve the performance of the system's global response. We have performed a case of study based on the movie recommendation domain with heterogeneous groups according to the group personality composition and a group topology based on a social network.

Keywords: Group Recommendation, Personality, Group Topology, Social Network.

1 Introduction

Within in the CBR community there have been a number of sophisticated techniques for the elicitation of good queries from the user and efficiently assessing similarity between the current query and the available cases. It is straightforward to establish a correspondence between retrieval in CBR systems and the process of obtaining goals and preferences of a user in order to rank products, services or information sources in recommender systems. Actually, CBR has played a key role in the development of several classes of recommender system [1,2].

Although most of the most popular recommender systems are focused on recommending items for individual users, the need of systems capable of performing recommendations for groups of people is getting more interest as there are many activities that are carried out in groups, e.g., going to the cinema with friends, watching TV at home or listening music in the car. Existing works on group recommender systems [3] are typically based on the aggregation of the preferences of the members of the group where every person in the group is considered as equal to the others.

A group recommender usually manages several subsets of preferences -one per person- that have to be managed independently and combined to create a global recommendation suitable for everyone in the group. According to this, it is natural to think in a distributed architecture where the preferences of every user are managed independently by an autonomous agent. This agent represents the user inside the global recommender system and is in charge of promoting his preferences when making a recommendation process for the whole group. To allow this kind of distributed CBR systems, we have developed a set of extensions for the jCOLIBRI CBR framework. These extensions (ALADIN, SALADIN and D^{2}ISCO[18]) enable the development of distributed CBR systems that includes the deliberative process required to build recommenders for groups.

Our recent work [4] involves the improvement of current group recommendation techniques by introducing a novel factor: the personality of every individual. Intuitively, when a group of 2 or more friends choose a movie there are some members that are only happy if they impose their opinion, whereas other individuals don't care letting other people decide. Therefore, we have used a personality test to obtain different profiles that interact in different ways when joining a decision making process.

Besides personality, there are other factors to consider regarding the structure of the group. In this paper we describe a preliminary work about making recommendations for groups of people connected through social network structures. We propose a distributed CBR architecture where each person in the group is represented by an agent and the final recommendation is influenced by the personality of each member of the group and the way they are interconnected through their social relations, basically friendship, defined in the social network. Therefore our method proposes making recommendations to groups using existing techniques of collaborative filtering [5], taking into account the group personality composition [4] and the social connections between the individuals of the group. We are testing our method in the movie recommendation domain.

The paper runs as follows: Section 2 describes existing techniques to obtain recommendations for groups and gives some details of our "personality aware" recommendation process. Section 3 outlines our extension for building distributed CBR systems in jCOLIBRI, whereas Section 4 describes our proposal for including social network's topologies in the decision process. Section 5 depicts our preliminary results and Section 6 concludes and presents the main lines of future work.

2 Recommendations to Groups

Recommender systems have traditionally recommended items to individual users, but there has recently been a body of work about recommenders that extend their recommendations to groups of users [3]. When moving from individuals to groups many new issues arises. For example, acquiring the preferences of the group, helping the group to decide the more convenient option, or explaining the recommendation to the group. Depending on the size and homogeneity of the group the recommender system has to choose the option that satisfies the biggest number of people taking into account the individual user preferences. As stated in [3] the main approaches to generate a preference aggregation based on the individual user preferences are (a) merging the recommendations made for individuals, (b) aggregation of ratings for individuals and (c) constructing a group preference model.

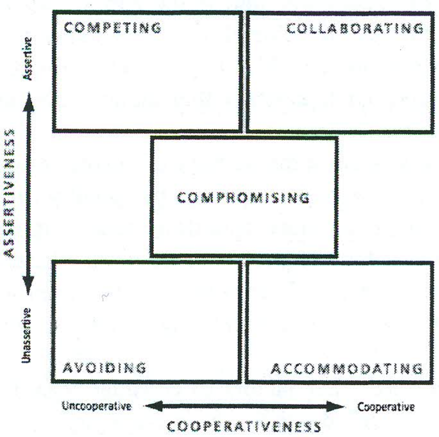


Fig. 1. TKI personality modes

The most common employed approaches in group recommenders are (b) and (c). They are used in many different fields like selecting the background music of a fitness center [6] recommendation of video clip sequences [7], movies [8] among others.

The work described in [8] recommends movies for groups based on the inferred ratings by MovieLens and using the "least misery" strategy to generate the preference aggregation. This strategy supposes that a small group of people will be as happy as its least happy member. The work in [9] criticizes the aggregation strategies like the one employed in PolyLens because they claim that this strategies combine the ratings always in the same way without considering how the members in the group interact with each other. To alleviate this issue, they propose the use of genetic algorithms to learn about the interactions among the users that make up a group from the known group ratings. Although the results seem to be significant, they suppose that the groups are fixed and they have previously rated some items in group.

Most of the previous works in group recommendation consider the preferences of every member of the group with the same degree of importance and try to satisfy the preferences of every group member. However, groups of people can have very different characteristics like size and can be made of people with similar or antagonistic personal preferences. It is a fact that when we face a situation in which the concerns of people appear to be incompatible *conflict* arises. The existing recommender systems for groups typically solve the conflict trying to maximize the preferences of the biggest number of group members. However, the general satisfaction of the group is not always the aggregation of the satisfaction of its members as different people have different expectations and behaviour in conflict situations that should be taken into account. In [4] we have presented a method for recommendation to groups where we distinguish between different types of individuals in a group. Our research characterizes people using the Thomas-Kilmann Conflict Mode Instrument (TKI) [17] that describes a person behavior in conflict situations along two basic dimensions: *assertiveness* and *cooperativeness*. These two dimensions of behavior can be used to define five personality modes of dealing with conflicts: competing, collaborating, avoiding, accommodating and compromising (see Figure 1). Our method named Personality Aware Recommendation to Groups [4] takes into account these five personality modes. We study how the group personality composition influences the recommendation accuracy for the group, and how it is improved for certain types of groups compared with different simple group recommendation algorithms. We have experimentally evaluated the behavior of this method using the MovieLens data set with groups of users of different sizes and degrees of homogeneity. The novelty of our approach lies in the use of the member personalities to choose the most interesting movie that would better satisfy the whole group, using the type of personality trait of dealing with conflicts of each member to weight the influence of his/her ratings during the recommendation process.

Our current work described in this paper proposes the use of real social networks topologies to reflect the real interactions of the users. Our proposal relates also with other works that have employed more social issues in order to include the group member interactions to perform the recommendations. For example, the work described in [7] uses individual satisfaction and the transmission of emotions in order to recommend a sequence of video clips for a group. They consider that a member changes the selection of her best clip according to the clip selected during the previous selection step. This change can be reflected in the recommendation algorithm as an individual satisfaction function that computes the individual affective state. This state influences on the affective state of the other members, getting carried by others emotional state should be taken into account during the group recommendation. Other interesting results from this work are that the most common strategies employed by individuals in real group recommendation are non-dominant strategies such as the average, least misery and average without misery. Additionally, they point out the existence of a tendency where the social status influences on the selection.

3 Distributed CBR Systems in jCOLIBRI

jCOLIBRI is currently a reference platform in the CBR community for building CBR systems that includes facilities to design different types of CBR applications [10,11,12].

However its underlying architecture has taken the conventional so called *single agent, single case base* problem solving approach where one, usually well-maintained, case base functions as the central knowledge resource. Research efforts in the area of distributed CBR concentrate on the distribution of resources within CBR architectures and study how it is beneficial in a variety of application contexts. In contrast to single-agent CBR systems, multi-agent systems distribute the case base itself and/or some aspects of the reasoning among several agents. In [13] the research efforts in the area of distributed CBR are categorized using two criteria: (1) how knowledge is organized/managed within the system (i.e. single vs. multiple case bases), and (2) how knowledge is processed by the system (i.e. single vs. multiple processing agents).

Much of the work in distributed CBR assumes multi-case base architectures involving multiple processing agents that differ in their problem solving experiences [14]. One key point in distributed CBR systems is the "ensemble effect" described in [15] they state that a collection of agents with uncorrelated case bases improves the accuracy of any individual. With this idea we have extended jCOLIBRI to support distributed CBR applications, as it will improve the performance of standard (single-agent, single-case base) CBR applications.

The extension to support distributed CBR applications in jCOLIBRI is called ALADIN (Abstract Layer for Distributed Infrastructures). This layer defines the main components of every distributed CBR system: agents, directory, messages, etc. and could be implemented using different alternatives: JADE[1], sockets, shared memory, ... It was defined after reviewing the existing literature on distributed CBR and tries to fit the IEEE FIPA standards for multiagent systems. Because ALADIN is only composed of interfaces that define the behavior of the system, we have developed an implementation of this abstract layer using standard network sockets. This extension in called SALADIN (Sockets implementation of ALADIN) and provides a fully functional muli-agent environment for building distributed CBR systems. Both extensions are available in the jCOLIBRI contributions web page[2].

Once the building blocks were defined, in [18] we have presented our work for including deliberative capabilities in the distributed CBR systems. This approach - named D^2ISCO[3] is based on the AMAL protocol that is proposed in [15] as a case-based approach to groups of agents that coordinate, collaborate, and communicate in order to improve their collective and individual decisions by learning from communication and by the argumentation over debated outcomes [16]. This protocol consists on a series of rounds. In the initial round, each agent states which is its individual local solution for the problem. Then, at each round an agent can try to rebut the solution or prediction made by any of the other agents giving a counterexample. When an agent receives a counterargument or counterexample, it informs the other agents if it accepts the counterargument (and changes its solution) or not. Moreover, agents have also the opportunity to answer to counterarguments by trying to generate a counterargument to the counterargument.

The work presented in this paper tries to reuse the collaborative and deliberative capabilities of D^2ISCO by including two novel factors: the personality of every individual (detailed in following section) and their interconnections (explained in Section 5).

---

[1] JADE is a framework for building multi-agent systems following the FIPA specifications. It is available at: http://jade.tilab.com/

[2] *jCOLIBRI contributions:* http://gaia.fdi.ucm.es/projects/jcolibri/jcolibri2/contributions.html

[3] Deliberative, DIStributed and COlaborative extension for jCOLIBRI

4 Including Social Networks in the Recommendation Process

The novelty of our approach is to include the social network connections into our recommendation model. So our main goal of our ongoing work described in this paper is to improve the recommendations by to taking into account both, the network topology and the group personality composition. In order to achieve this, we are using the SALADIN platform for organizing the multi-agent system with a social network topology, where each agent represents a member of the group. Every agent will discuss his best interest with the other agents he is connected to, and the final recommendation will be, as we have been explaining, depending on the personality, this means that the preferences of each individual will have different weights depending on the way in which that particular member of the group would react in a conflict situation.

To carry out this project we would like to improve the techniques that have been typically used when recommending items to groups [3]. One of the main approaches to generate a recommendation is based on aggregation of ratings for individuals. However, as we have explained in Section 2, this aggregation strategy has been criticized by some authors because the ratings are combined always in the same way, without considering how the members in the group interact with each other. We will like to fix that by including the group member interactions, that is, to perform the recommendation using personal information about the conflict mode behavior of every group member and how they will interact with each other.

In this way we consider the recommendation process in a group taking into account a network topology based on a social network. Social networks have been one of the most important topics in the last few years, with nets like Facebook[4], Twitter[5], MySpace[6]...etc, and they represent a more realistic form of the structure and relations between group members. Therefore we think that these recommendations, that we obtain with this new method are more accurate and realistic than the ones we would get if we'd applied the topology "all connected", which is the default topology used in typical group recommendation approaches. Figure 2 shows the difference between the two different types of network topologies mentioned, where we can see how the agents are connected.

In our particular case of study we are trying to imitate the real process of recommendations to groups when deciding which movie they will be watching. We will test how the group personality and its structure influence the recommendation accuracy for the group. That is, taking into account the closeness of each individual member of the group, who will listen to who, who is the center of the discussion, among others questions. And also considering their own personality, the way they're going to react if there is a proposal different than the one they wanted, if they are open minded and would not complain when a movie they're less keen on is the chosen one, and so on.

[4] http://www.facebook.com/

[5] http://www.twitter.com/

[6] http://www.myspace.com/

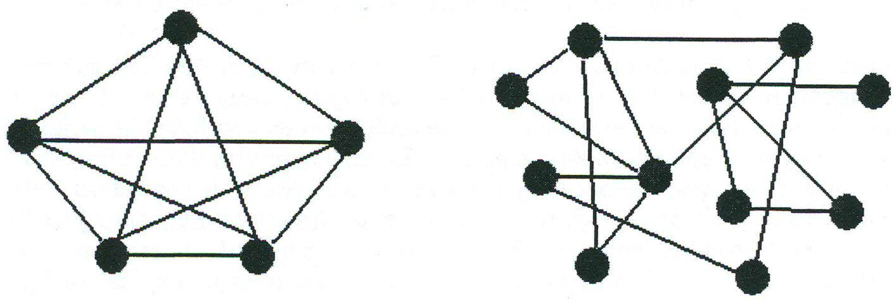


Fig. 2. Comparison between "all connected" networks (left) and social networks (right) topologies

These users interchange recommendations with other users that have a social link with them. To do this for groups is kind of a natural thing because people tend to do this kind of activities with their friends and family, i.e, following social links. Besides they usually go together in the same group.

The reason why this new organization of the structure of the group will affect and improve the result of the recommendation is mainly because with the social network topology we give a more realistic structure and organization of the agents, which is closer to how the argumentations would take place in a real group when they argue on which movie to watch. For a given a group of friends, we draw a network where each node represents a person, and each connection represents that that particular person has a relation with the one he is connected to. Each node discusses his/her opinions with all her neighbors in the network. When two nodes are not connected, it means that the people they are representing don't know each other or that they're not close, this structure is reflecting a face to face discussion where they would have never argue or have to come to a solution between each other. For example if someone brings along his/her boy/girlfriend and the rest of the group doesn't know him, this new member will never start arguing or try convincing someone he has never met before, therefore when we represent this relations we only connect him with the ones that do know him. On the other hand in the all connected network topology every agent will debate with all the rest, no matter if the person he represents knows the other one or not.

For the process of argumentation we are using a multiagent system, so each agent will represent one member of the group, and will argue for his best interest. For the implementation, we will use D^{2}ISCO [18] and jCOLIBRI [3,19], in this way we use the structure of the collaborative nodes for the CBR system that D^{2}ISCO uses, the network topology will be the social network type. The reasoning protocol will begin with an agent issuing a query to the agents that is linked to. Each agent will provide its individual local solution for the problem. At each round an agent can try to rebut the solution or the prediction made by any other of his neighbour agents. This different counterexamples will have different weights depending on the personality of the person that is being represented by that particular agent.

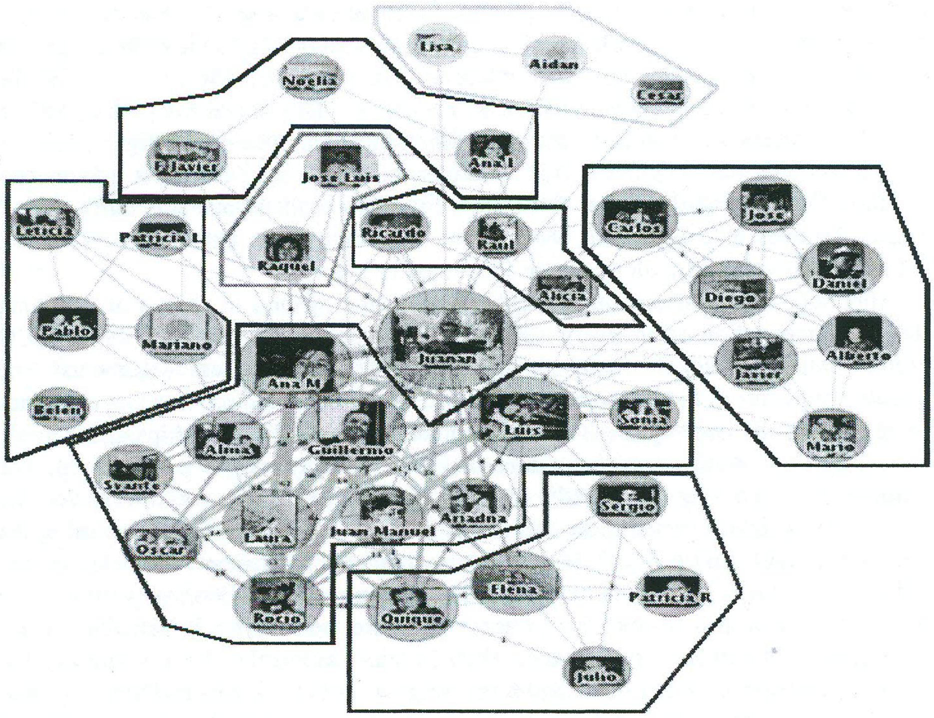


Fig. 3. Users organization in a real social network

Next we summarize our approach:

1. Given: a set of users $U=\{u_1,u_2,..., u_i,...,u_n\}$ (where u_q is the user that launches the query), the set of relationships of an user $rel(u_i)$, and the personality weight $per(u_i)$.
2. Split U into several subsets $S=\{S_1,S_2,...,S_j,...S_m\}$ where $S_j \subseteq U$.
3. For every $S_j \in S$

   compute I_j = obtainSubRecommendation(S_j, $rel(u_i \in S_j)$, $per(u_i \in S_j)$)
4. Being $I=\{I_1,I_2,...,I_j,...I_m\}$ the set of items recommended by each subset,

   For every $I_j \in I$

   $w_j = \sum rel(u_i \in S_j) \cdot r + \sum per(u_i \in S_j) \cdot p$
5. Return the item I_j with the maximum weight w_j

Note that this algorithm implies several processes that must be carefully addressed. Firstly we found the process of splitting the users according to the topology of the social network. In our preliminary experiments we are identifying the *k* users with more relationships as the "group leaders" and then assigning every remaining user to one of these leaders. Moreover, there is a second way to organize the users: some social networks allow to group users according to the different groups they have joined or by common interests. For example, Figure 3 shows the representation of a real social network in Facebook[7]. This social network allows to classify the users into groups (represented using different areas in the figure, that we have intentionally highlighted). This feature enables us to split the U set into subsets S_j that reflect the relationships of the users in a very realistic manner. Furthermore, Figure 3 serves to confirm our premise that real interactions between users do not follow an all-with-all topology as was illustrated in Figure 1.

After splitting the users, our algorithm applies the method in charge of obtaining the recommendation from every subset S_j. This method is based in the D^2ISCO argumentation protocol (detailed in Section 3) that uses arguments and counterarguments in the deliberation process. Finally the global process that combines the partial results (step 4), takes into account the number of relationships inside every subset S_j ($rel(u_i \in S_j)$) and their accumulated personality weight $per(u_i \in S_j)$. The number of relationships is directly obtained from the social network. However, the personality weight is much more difficult to measure. To obtain it, we are using the TKI tests detailed in Section 2. We are summarizing the 5 different modes in just one value. This value is high if the person has a competing and collaborative profile and is low if the person has an avoiding or accommodating personality. Informally, we are giving more importance to the subset that contains more related users and has the highest personality. Parameters r and p are weights for both factors (related users and personality). Parameter r defines how the number of relations of every subset determines the final recommendation. In addition, parameter p weights the impact of the accumulated personality in the recommendation. Both parameters will be measured in an experimental way. Although initial results cannot lead us to any conclusion, one of the goals of our experiments is to measure how their influence will affect the recommendations.

Next section describes some preliminary results.

5 Preliminary Results

From our ongoing work we have already obtained some preliminary results that we summarize in this section. However detailed results from a complete experimentation process will be performed as future work.

The social network topology gives a more realistic structure and organization of the agents, which is closer to how the argumentations would take place in a real group

[7] This figure has been obtained using the touch graph application in Facebook. http://apps.facebook.com/touchgraph/

when they all argue on which movie to watch. We have informally tested our approach in the movie recommendation case of study and we have seen that our premises and new ideas are effective and produce better results than the Personality Aware Recommendation to Groups method using the all-connected network structure. We have informally question our users and obtained slightly better satisfaction results in the group (around 10% improvement).

Besides known groups of people we also plan to make me tests using anonymous data from different users through an application for Facebook called MyPersonality[8]. It consists on personality self-ratings and friend ratings from users around the world. This includes demographic information such as age and the country they live in, personal information such as favorite films and also a list of their friends which provides an opportunity unique to social networks to examine and model network effects.

6 Conclusions and Future Work

In this paper we have introduced a novel method of making recommendations for groups based on existing techniques of collaborative filtering and taking into account the group personality composition and the structure of the group. Once shown that personality profiles can improve a recommendation for a group of people [4], we are going to extend this approach by reflecting in a more realistic way the relationships between the users involved in the recommendation.

To achieve this goal we are using the existing extensions of jCOLIBRI2 for building distributed and deliberative CBR systems. The SALADIN extension provides the infrastructure required to build the distributed system, whereas D^{2}ISCO [18] offers deliberation capabilities through a modified version of the AMAL protocol.

This ongoing work poses many open issues to be addressed:

1. Test the real improvement of this method using real data from social networks.
2. Study other types of personality and compare them with the TKI conflict modes. We have already done some tests studying how people's affective reactions to the cultural media (favorite songs, movies and classic art) may be an indicator of their own personality [20].
3. Explore techniques for grouping the users into the subsets involved in the proposed recommendation process.
4. Explore different methods to combine the items proposed by every subset of users taking into account the relationships and personality factors.

We think that creating a system with memory of the previous recommendations is a necessary step when providing a whole set of recommendations. We have been focusing on the situation in which a recommender will make recommendations to a

[8] http://mypersonality.org/research/interested-in-collaborating/

group just once. But often the group will expect the system to make a larger set of decisions. We could treat each decision separately, however to avoid future group dissatisfaction, ensuring fairness could be taken into account. For example, there might be a situation when a given recommendation is very attractive for the group as a whole but one group member was especially dissatisfied. It would be desirable that the future recommendations favor that particular group member, so that he/she reaches an appropriate level of satisfaction. We are studying how we could take into account the history of the results that have been obtained. This means that it would be possible to create a system with memory of the previous recommendations for the group, so that if one member had to accept a proposal he wasn't very interested in, now he will have some kind of preference for the next recommendation, meaning that the weight of his opinion will be higher this time. These weights would also be adapted depending on the personality of each member.

7 References

1. D. Bridge, M. H. Göker, L. McGinty, and B.Smyth. Case-based Recommender systems. Knowledge Engineering Review, 20(3):315–320, 2006.
2. Smyth, B.: Case-base recommendation. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321, Berlin Heidelberg New York, Springer-Verlag (2007) this volume.
3. A. Jameson and B. Smyth. Recommendation to groups. In P. Brusilovsky, A.Kobsa, and W. Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 596-627. Springer, 2007.
4. Juan A. Recio-García, Guillermo Jiménez-Díaz, Antonio A.Sánchez-Ruiz and Belén Díaz-Agudo, *Personality Aware Recommendations to Groups*, in: 3rd ACM Conference on Recommender Systems, New York, USA, ACM Press, 2009
5. J. A. Recio-García, B. Díaz-Agudo, and P. A.González-Calero. Prototyping Recommender Systems in jCOLIBRI. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages243-250, New York, NY, USA, 2008. ACM.
6. McCarthy, Joseph F. and Anagnost, Theodore D. *MusicFX: an arbiter of group preferences for computer supported collaborative workouts*, CSCW '98: Proceedings of the 1998 ACM conference on Computer supported cooperative work, pages 363-372. NY, USA, 1998.ACM.
7. Masthoff, Judith and Gatt, Albert. *In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems*. User Modeling and User-Adapted Interaction, volume 16, pages 281-319, number 3-4. Hingham, MA, USA, 2006.
8. O'Connor, Mark and Cosley,Dan and Konstan, Joseph A. and Riedl,John. *PolyLens: a recommender system for groups of users*. ECSCW'01: Proceedings of

the seventh conference on European Conference on Computer Supported Cooperative Work, pages 199-218. Norwell, MA, USA, 2001. Kluwer Academic Publishers.

9. Y.-L. Chen, L.-C. Cheng, and C.-N. Chuang. *A group recommendation system with consideration of interactions among group members.* Expert Syst.Appl., 34(3):2082-2090, 2008.
10. Juan A. Recio-García, Belén Díaz-Agudo, Marco A. Gómez-Martín and Nirmalie Wiratunga, *Extending jCOLIBRI for textual CBR*, in: Proceedings of Case-Based Reasoning Research and Development, 6th International Conference on Case-Based Reasoning, ICCBR 2005, Springer, 2005.
11. Belén Díaz-Agudo, Pedro A. González-Calero, Juan A. Recio-García and Antonio A. Sánchez-Ruiz, *Building CBR systems with jCOLIBRI* (2007), in: Special Issue on Experimental Software and Toolkits of the Journal Science of Computer Programming, 69:1-3
12. Juan A. Recio-García, Belén Díaz-Agudo, Pedro A. González-Calero and Antonio A. Sánchez-Ruiz, *Ontology based CBR with jCOLIBRI*, in: Applications and Innovations in Intelligent Systems XIV. Proceedings of AI-2006, the Twenty-sixth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Springer, 2006
13. Enric Plaza and Lorraine McGinty. *Distributed case-based reasoning.* Knowledge Eng. Review, volume 20, number 3, pages 261-265,2005.
14. McGinty, L., Smyth, B.: *Collaborative case-based reasoning: Applications in personalised route planning.* In: Procs. ICCBR 2001, Springer LNAI. (2001) 362-376.
15. Ontanon, S., Plaza, E.: *Arguments and counterexamples in case-based joint deliberation.* In: Argumentation in Multi-Agent Systems, ArgMAS 2006, Selected and Invited Papers. Volume 4766 of LNCS., Springer (2006) 36-53.
16. Ontanon, S., Plaza, E.: *An Argumentation-based Framework for Deliberation in Multi-Agent Systems.* Argumentation in Multi-Agent Systems. LNCS, volume 4946, pages 178-196, 2008.
17. K. Thomas and R. Kilmann. *Thomas-Kilmann Conflict Mode Instrument.* Tuxedo, N.Y., 1974.
18. Sergio González-Sanz, Juan A. Recio-García and Belén Díaz-Agudo, *DISCO: Distributed Deliberative CBR Systems with jColibri*, in: 1st International Conference on Computational Collective Intelligence, Louisiana, USA, 2009
19. J. Recio-Garcia. *jCOLIBRI: A multi-level platform for building and generating CBR systems.* Phd Thesis,Universidad Complutense de Madrid, Spain, 2008.
20. Michael A.Faber and John D.Mayer, *Resonance to archetypes in media: There's some accounting for taste.* Journal of Research in Personality, Volume 43, Issue 3, June 2009, Pages 307-322.

Chapter 9

Distributed Deliberative Recommender Systems

9.1 Citation

Recio-García, J. A., Díaz-Agudo, B., González-Sanz S. and Quijano-Sánchez, L. Distributed Deliberative Recommender Systems. Journal of Transactions on Computational Collective Intelligence, 1; pages: 121-142, Springer, 2010.

9.2 Contributions covered by this paper

In this paper we have studied a new technique of making social group recommendations by using a framework that implements deliberative and collaborative CBR systems by using CBR agents that collaborate, argument and counterargument their local results with other agents to improve the performance of the recommendation.

Distributed Deliberative Recommender Systems

Juan A. Recio-García, Belén Díaz-Agudo,
Sergio González-Sanz, and Lara Quijano Sanchez

Department of Software Engineering and Artificial Intelligence,
Universidad Complutense de Madrid, Spain
jareciog@fdi.ucm.es, belend@sip.ucm.es,
sergio.gonzalez@ciemat.es, lara.quijano@fdi.ucm.es

Abstract. Case-Based Reasoning (CBR) is one of most successful applied AI technologies of recent years. Although many CBR systems reason locally on a previous experience base to solve new problems, in this paper we focus on *distributed retrieval* processes working on a network of collaborating CBR systems. In such systems, each node in a network of CBR agents collaborates, arguments and counterarguments its local results with other nodes to improve the performance of the system's global response. We describe D²ISCO: a framework to design and implement deliberative and collaborative CBR systems that is integrated as a part of jCOLIBRI 2 an established framework in the CBR community. We apply D²ISCO to one particular simplified type of CBR systems: recommender systems. We perform a first case study for a collaborative music recommender system and present the results of an experiment of the accuracy of the system results using a fuzzy version of the argumentation system AMAL and a network topology based on a social network. Besides individual recommendation we also discuss how D²ISCO can be used to improve recommendations to *groups* and we present a second case of study based on the movie recommendation domain with heterogeneous groups according to the group personality composition and a group topology based on a social network.

1 Introduction

Case-Based Reasoning (CBR) is based on the intuition that situations tend to recur. It means that new problems are often similar to previously encountered problems and, therefore, that past solutions may be of use in the current situation [1].

Research efforts in the area of *distributed CBR* concentrate on the distribution of resources within CBR architectures and study how it is beneficial in a variety of application contexts. In contrast to single-agent CBR systems, multiagent systems distribute the case base itself and/or some aspects of the reasoning among several agents. In [2] the research efforts in the area of distributed CBR are categorized using two criteria: (1) how knowledge is organised/managed within the system (i.e. single vs. multiple case bases), and (2) how knowledge is processed by the system (i.e. single vs. multiple processing agents).

N.T. Nguyen and R. Kowalczyk (Eds.): Transactions on CCI I, LNCS 6220, pp. 121–142, 2010.

Much of the work in distributed CBR assumes multi-case base architectures involving multiple processing agents differing in their problem solving experiences [3]. The "ensemble effect" [4] shows that a collection of agents with uncorrelated case bases improves the accuracy of any individual. Multiple sources of experience exist when several CBR agents need to coordinate, collaborate, and communicate. Within this purpose AMAL has been proposed as a case-based approach to groups of agents that coordinate, collaborate, and communicate in order to improve their collective and individual decisions by learning from communication and by argumentation over debated outcomes [4].

Our current work, described in this paper, explains the modification of the AMAL approach to use fuzzy reasoning to combine recommendations that are retrieved from each agent's database using CBR to compute similarity. Recommendations are propagated through a network that is modelled after a social network, while agents model behaviours to exhibit personality traits that feed into a process of group recommendation.

The paper also has important technological contributions. jCOLIBRI 2 [5] is a well established framework in the CBR community that can be used to design different types of CBR systems [6]. However its underlying architecture has taken the conventional so called *single agent, single case base* problem solving approach where one, usually well-maintained, case base functions as the central knowledge resource. In this paper we propose two working extensions, ((S)ALADIN and D2ISCO[18]), to jCOLIBRI 2 to design deliberative and distributed multiagent CBR systems where the case base itself and/or some aspects of the reasoning process are distributed among several agents. Our work focuses on distributed *retrieval* processes working on a network of collaborating CBR systems. We deal with aspects such as the topology of the network, the definition of trust models for the different agents, voting and negotiation techniques between agents to reach a consensus in the final solution. In this paper we consider a simplified type of retrieval-only CBR systems: recommender systems. CBR has played a key role in the development of several classes of recommender systems [7] as it is straightforward to establish a correspondence between retrieval in CBR systems and the process of obtaining goals and preferences of a user in order to rank products, services or information sources in recommender systems.

In the network of agents every agent should be able to define the trustworthiness regarding the connected agents [8,9]. In the recommender systems arena, in order to provide meaningful results, trust, or reputation, must reflect user similarity to some extent; recommendations only make sense when obtained from like-minded people exhibiting similar taste. Our model is based on the social connections of the collaborative agents, including the level of trust of the agent they collaborate with. We use social trust as the basis for recommender systems [10] [11][12]. Social networks offer an opportunity to get information about the social environment of a given user and associate trustworthiness values to it. If a node receives a query and it cannot give a good answer to it, then it will ask for collaborations with other nodes it has relations or links with. The trust models evolve in time according to the real accuracy of the answers provided by a certain node.

We perform a case study for a collaborative music recommender system for individuals and present the results of an experiment of the accuracy of the system results using a fuzzy version of the argumentation system AMAL and a network topology based on a social network.

Although most of the popular recommender systems are focused on recommending items for individual users, the need of systems capable of performing recommendations for groups of people is getting more interest as there are many activities that are carried out in groups, e.g., going to the cinema with friends, watching TV at home or listening music in the car. Existing works on group recommender systems [13] are typically based on the aggregation of the preferences of the members of the group where every person in the group is considered as equal to the others. A group recommender usually manages several subsets of preferences -one per person- that have to be managed independently and combined to create a global recommendation suitable for everyone in the group. According to this, it is natural to think about a distributed architecture where the preferences of every user are managed independently by an autonomous agent. This agent represents the user inside the global recommender system and is in charge of promoting his preferences when making a recommendation process for the whole group. We present a case study of movie recommendation for groups.

The paper runs as follow. Section 2 describes our approach to design distributed CBR systems with deliberation capabilities, and the software support we provide for our approach. We focus on the modification of the AMAL protocol to use fuzzy reasoning to combine recommendations that are retrieved from each agent's database using CBR to compute similarity. A first case study in the music recommendation domain for individual users is presented in 3. In Section 4 we describe how to extend these deliberation capabilities for group recommendations and present a preliminary case study on the movie recommendation domain. Section 5 summarizes the main results and concludes the paper.

2 Distributed Reasoning for Collective Experiences

The development of distributed CBR systems with deliberation capabilities is not a simple task from the software engineering point of view. To alleviate this design cost we have identified a set of different features that characterize different architectures of distributed deliberative systems, namely:

- Number of agents. Distributed systems could range from a few agents to thousands of them, so the platform must be scalable.
- Size of the case base. From small to large sizes, it has a direct impact in the retrieval time.
- Overlapping of the case base of different agents. Some distributed systems contain agents with independent case bases where cases cannot be owned by several agents. On the other hand, some architectures allow the overlapping of individual case bases.

– Type of cases. Case bases can be homogeneous if cases have the same attributes or heterogeneous if their structure changes among agents. This feature has a repercussion in the deliberation protocol and the reasoning technique.
– Network topology. There are different ways to organize and link agents: all-with-all, ring, star, hierarchical, etc.
– Trust model. Depending on the nature of the distributed system, there are several options for building a trust model that will affect the deliberation process.
– Composition of the final result. Every agent can have a different impact in the final result depending on the number or quality of cases provided to the deliberation.
– Deliberation protocol. There are several deliberation protocols that are can be applied only depending on the nature of the agents.
– CBR reasoning. In this kind of systems every agent executes its own CBR reasoning process. This process could be different for every agent or they can execute just the same reasoning method.
– Query propagation. Queries can be propagated in different ways through the agents network.

We also provide software support and we have extended jCOLIBRI 2 [5], a general platform for the implementation of different kinds of CBR systems, to support designing different types of distributed deliberative systems characterized according to these features. The basic extension to support distributed CBR applications in jCOLIBRI 2 is called ALADIN (Abstract Layer for Distributed Infrastructures). This layer defines the main components of every distributed CBR system: agents, directory, messages, etc. and could be implemented using different alternatives: JADE [1], sockets, shared memory, ... It was defined after reviewing the existing literature on distributed CBR and tries to fit the IEEE FIPA standards for multiagent systems. Because ALADIN is only composed of interfaces that define the behavior of the system, we have developed an implementation of this abstract layer using standard network sockets. This extension is called SALADIN (Sockets implementation of ALADIN) and provides a fully functional multi-agent environment for building distributed CBR systems that could be particularized in many ways. In this paper we detail one of these particularizations.

D²ISCO[2] is built on top of SALADIN and it provides one particular choice of the deliberation functionality [14]. D²ISCO deliberation capabilities are rooted in the AMAL argumentation process proposed in [4,15]. These extensions and D²ISCO are available in the jCOLIBRI 2 contributions web page [3]. Next we will detail the deliberation protocol of D²ISCO.

[1] JADE is a framework for building multi-agent systems following the FIPA specifications. It is available at: `http://jade.tilab.com/`

[2] D²ISCO: **D**eliberative, **DIS**tributed & **CO**llaborative extension for j**CO**LIBRI

[3] jCOLIBRI contributions: `http://gaia.fdi.ucm.es/projects/jcolibri/jcolibri2/contributions.html`

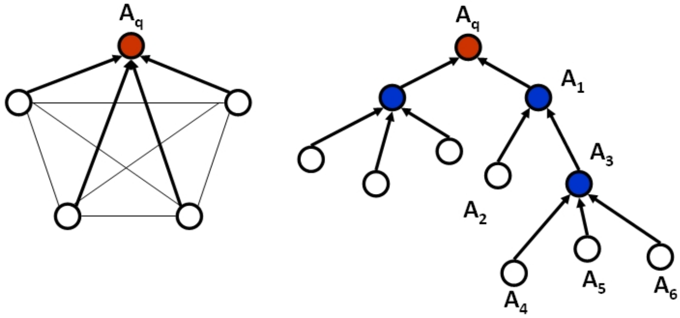


Fig. 1. Comparison between AMAL (left) and D²ISCO (right) topologies

2.1 D²ISCO Deliberation Protocol

D²ISCO deliberation capabilities are rooted in the AMAL argumentation process proposed in [4,15]. AMAL follows the same mechanism as human committees, first each individual member of a committee exposes his arguments and discuses those of the other members (joint deliberation), and if no consensus is reached, then a voting mechanism is required.

The interaction protocol of AMAL allows a group of agents $A_1, \ldots, A_n$ to deliberate about the correct solution of a problem Q by means of an argumentation process. Each of these agents uses a CBR system to find a solution for Q using their own case base and then they start a deliberation process to order the local solutions and find the best of them. If the argumentation process arrives to a consensual solution, the joint deliberation ends; otherwise a weighted vote is used to determine the joint solution [15]. Moreover, AMAL also allows the agents to learn from the counterexamples received from other agents. The reasoning protocol begins with an agent (A_q) issuing a query to the agents that is linked to $(A_1, A_2, ..., A_n)$. Each one of these agents retrieves k items from their own case base. Then, an argumentation process consisting of k cycles is performed to defend and discard the proposed items by means of counterexamples. When the process finishes A_q receives at most k trusted items.

The AMAL protocol consists on a series of rounds. In the initial round, each agent states which is its individual local solution for the problem Q. Then, at each round an agent can try to rebut the solution or prediction made by any of the other agents giving a counterexample. When an agent receives a counterargument or counterexample, it informs the other agents if it accepts the counterargument (and changes its solution) or not. Moreover, agents have also the opportunity to answer to counterarguments by trying to generate a counterargument to the counterargument.

Although the original AMAL design does not satisfy requirements for our application it is readily adaptable. Next we summarize some of the requirements we need to adjust from the original AMAL approach:

– Regarding the topology, AMAL proposes to link every agent with all the agents in the system. This N to N topology has repercussions in the efficiency of the argumentation process and it is not scalable to real size systems.
– In real scenarios, users –or agents– are organized and linked by means of topologies that are analogous to social networks. Therefore, the N to N topology does not reflect faithfully the relations among users.
– AMAL is based on Description Logic (DLs) case representation and reasoning what implies an additional knowledge representation effort.
– AMAL does not take into account the trust between agents in the argumentation process.

Our D²ISCO approach for building distributed and deliberative systems solves the requirements mismatch with the original AMAL protocol. Namely, the features of our framework are:

– The topology of the systems follows the structure of a social network. This enables to increase easily the number of agents in the system and to incorporate trust factors in the argumentation process that are obtained from this social network.
– Social networks offer an opportunity to get information about the social environment of a given user and associate trustworthiness values to it. If the social network connects users with similar tastes, that means that it reflects the preferences of the users and make it possible to add this information to the argumentation process. This way, social networks have two possible uses: 1) to obtain the trust among users, and 2) to compute the similarity between users according to its preferences.
– The argumentation process is directed by a lead node/agent A_q that issues the query and organizes the deliberation of its children nodes (A_c). This agent is in charge of accepting or rejecting the counterexamples presented by those children agents.
– Our argumentation and case retrieval process is hierarchical. When solving a problem Q, the agent that issues the query A_q becomes the root of the whole hierarchy of agents –defined by the structure of the social network. Then, the query is sent to the leaves of the tree and the retrieval follows an inverse direction. The leafs of the tree deliberate with their immediate parent A_p node that organizes the reasoning. When this intermediate deliberation finishes, A_p participates in the deliberation organized by its parent node but this time it takes the role of a children node A_c. This behavior is repeated until reaching the root A_q. It is important to note that in every intermediate deliberation A_p receives the cases retrieved by its children nodes A_c and incorporates them in its own case base. Figure 1 illustrate the difference between AMAL and D²ISCO topologies. The direction of the arrows represent the forwarding of cases. The left graph shows a typical AMAL net where every agent is linked with every agent. On the right we find the hierarchical topology of D²ISCO. Here we can note how the argumentation process begins with A_3 being the organizing agent (A_c) and $\{A_4, A_5, A_6\}$ its corresponding children (A_c). Afterward, A_3 takes part of the deliberation

conducted by A_1 where $A_c = \{A_2, A_3\}$. Finally, A_1 contributes in the final deliberation leaded by A_q.
- D²ISCO reasons with the case ratings and it does not requires expressive case representations based on DLs. To substitute the reasoning capabilities of DLs, our approach uses a fuzzy decision system. Moreover this fuzzy system takes into account the trust and similarity between users –obtained from the social network– and the similarity between cases.
- Because we are not using logical formulas to define the counterarguments, our deliberative process applies the concept of *defenses*. Defenses are complete cases that are highly rated by the agent involved in the deliberation and are offered to trust the arguments presented by the agent.

Once we have described the behavior of our approach, we can focus on its main improvement: the fuzzy decision system. Next section summarizes its main features and illustrates it by means of an example.

2.2 Fuzzy Decision System

A key feature of the distributed reasoning protocol described in the previous section is the decision system that accepts or rejects counterexamples and defenses. In the original AMAL protocol these arguments are generated using a description logic. Our proposal relies in a fuzzy reasoner [16] that allows extending the protocol to cases which attributes are not expressed using a description logic, and thus, it is not possible to generate arguments using logic induction. For instance, in the CBR recommender systems, most of the items' attributes are numeric values referring customers' opinions and it is not possible to generate a logic induction using these numeric values. However, these attributes contain important pieces of information that must be used in order to improve the results of the recommender system. The developed fuzzy reasoner allows using these numeric attributes to generate arguments for the reasoning protocol.

It is important to note that the agent that leads the argumentation (A_p) does not retrieve cases from its own case base, but plays an important role in the argumentation because defines the trust in the agents involved in the argumentation process. Trust is an important input for the fuzzy subsystems explained below. It is a numeric value between the leader of the argumentation process A_p an a agent taking part in the argumentation A_r. The value of the trust between agent A_p and agent A_r may vary in time according to the quality of the solutions given by A_r to the queries started by A_p. The value of trust from A_p to A_r can be different to the value of trust from A_r to A_p because the quality of the solutions of A_r can be higher or lower than the solutions given by A_p. A set of membership functions must be defined in order to allow the system using this value.

According to the terminology of recommender systems we define the measure of *goodness* as the *rating* of the item given by the corresponding agent/user. In our case, goodness is a numeric value which minimum and maximum values may vary according to the concrete problem solved by the distributed CBR system (for instance, in the following example the maximum value of goodness is 10 and the minimum is 0). A set of membership functions must be defined in order to allow the system using the goodness value. As the maximum and the minimum values, the membership functions must be fitted to the concrete problem that is being solved. In the following example, five different membership functions have been defined.

The fuzzy decision system is divided into five subsystems implemented using large fuzzy rule bases. Each one of these five subsystems is involved in one step of the argumentation process. The designed subsystems are:

1. **Case evaluation subsystem:** it generates a value V_t measuring the degree of trust for a certain case C_i solving a query Q. It uses as inputs: 1) the value of goodness of C_i in its local case base, 2) the similarity between C_i and the current query Q, and 3) the compatibility between agent A_i that returns C_i and the agent that initiates the query A_q.

   The value V_t is maximum when the value of goodness of C_i is maximum, C_i is equal to the query Q and A_i is similar to the agent that starts the query A_q. V_t will decreased if the value of goodness of C_i falls, C_i is not complete similar to the query Q or A_i is not complete similar to the agent A_q. The minimum value of V_t is reached when the value of goodness of C_i is minimum, C_i has nothing in common to the query Q and A_i and A_q are not compatible at all. It means that a case is a good solution to a query Q if the user's rating (goodness) is high, the case is similar to the query Q and the user is compatible to the user that makes the query.
2. **Counterexample evaluation subsystem:** a counterexample against the case C_i is a case C_e that is rather similar to C_i but it has a low value of goodness. The subsystem measures the trust of a counterexample (V_c). If an agent A_i presents a counterexample, it includes the value V_c that indicates its confidence on its argumentation. To obtain this value, the fuzzy subsystem uses: 1) the goodness of C_e in its local case base, 2) the similarity between C_e and Q, and 3) the compatibility between agent A_i and A_q.

   The value V_c is maximum if the value of goodness of C_e is minimum and the similarity between C_e and Q and between A_i and A_q is maximum. If the value of goodness of C_e rises, or the similarity between C_e and Q or between A_i and A_q falls, the value V_c will fall. The minimum value of V_c is reached when the value of goodness of C_e is maximum, and the similarity between C_e and Q and between A_i and A_q are minimum. It means that a case is a good counterexample if the user's rating (goodness) is low, the case is similar to the case being rebutted and the user is compatible with the user that makes the query.
3. **Counterexample acceptance subsystem:** it decides if a counterexample is accepted. A value (V_a) is computed when an agent A_i proposes a counterexample C_e to the conducting agent A_p. It is based on: 1) the confidence of the counterexample V_c, 2) A_p trust in A_i, and 3) A_p trust in the agent being rebutted. Finally, the counterexample is accepted if the *defuzzifyed* value of V_a is higher than a certain threshold α defined in the system ($V_a > \alpha$). The

α value is a numeric value that should be tuned for the different problems to be solve by the CBR system.

If there is a large value of V_c, a large value of trust in A_i and a low value of the trust in the agent being rebutted, then the value V_a will be large. If the confidence in the counterexample V_c or the trust in A_i falls, the value V_c will fall. V_c also will fall if the trust in the agent being rebutted rises. It means that is easier to accept a counterexample of an agent if its trust is high and the trust in the agent being rebutted is low. If the trust in of both agents is high, the decision will rely on the value V_c.

4. **Defense evaluation subsystem:** A defense against a counterexample C_e is a case C_d that is rather similar to C_e and it has a high value of goodness. The subsystem measures the trust of a defense (V_d). If an agent A_j presents a defense for one of its solutions to a query, it includes the value V_d that indicates its confidence on its argumentation. This subsystem is analogous to V_c and is based on: 1) the similarity between a case C_d and the counterexample that is being rebutted C_e and 2) the goodness of C_d in its local case base.

   The value V_d is high when there is a high value of goodness of C_d and a high similarity between C_d and C_e. The value V_d will fall if any of these values fall. It means that a case is a good defense if it is similar to the counterexample being rebutted and it has a good user's rating.
5. **Defense acceptance subsystem:** it is analogous to the counterexample acceptance subsystem and decides if a defense is accepted by the conducting agent A_p. A value (V_n) is generated when an agent A_j proposes a defense V_d for one of its solutions to a query and against a counterexample with trust V_c of an agent A_i. The subsystem bases its decisions on: 1) the trust of A_p in A_i, 2) the trust of A_p in A_j, 3) the trust value of the counterexample V_c and, 4) the trust value of the defense V_d. Finally, the defense is accepted if the defuzzifyed value of V_n is higher than a certain threshold β defined in the system ($V_n > \beta$). The β value is a numeric value that should be tuned for the different problems to be solve by the CBR system.

   The value V_n will be high if the trust in A_i is low, the trust in A_p is high, the trust is the counterexample V_c is low and the trust in the defense V_d is high. It means that a defense is easier accepted if the trust of the rebutting agent and its counterexample are low and the trust in the defending agent and its defense are high.

Since the fuzzy decision system needs to compare cases and agents, two fuzzy *similarity functions* has been implemented. They obtain a similarity value between cases or agents comparing their attributes one by one. It is also possible to compare a case to query because a query is expressed as a case with missing attributes.

Example. To illustrate the behavior of our deliberative recommender, let's use a real example described in Table 1. Here we will not use intermediate nodes for clarity reasons. Also every agent returns only one case. In the example A_q

Table 1. Argumentation Example

	Rating	Artist	Title	Year	Price	Style
Query (Q)	10	Mike Oldfield	*	*	*	*
Case-A_2 (C_2)	5,58	Mike Oldfield	The Millennium Bell	1999	11	House
Case-A_3 (C_3)	3,77	Mike Oldfield	Hergest Ridge	1974	26	Rock
Case-A_1 (C_1)	1,82	Mike Oldfield	Crises	1983	22	Rock
Round 1 - A_2 has the token						
A_2 counterexample for C_1	0,5	Mike Oldfield	Incantations	1978	27	Rock
Accepted because A_1 cannot generate a defense						
Round result:						
C_2	5,58	Mike Oldfield	The Millennium Bell	1999	11	House
C_3	3,77	Mike Oldfield	Hergest Ridge	1974	26	Rock
Round 2 - A_3 has the token						
A_3 counterexample for C_2	1,36	Deep Dish	George is on	2005	19	House
A_2 defense (C_e)	5,3	Deep Dish	George is on	2005	19	House
A_2 defense is accepted						
Round result:						
C_2	5,58	Mike Oldfield	The Millennium Bell	1999	11	House
C_3	3,77	Mike Oldfield	Hergest Ridge	1974	26	Rock
Round 3 - A_1 has the token						
A_1 counterExample for C_2	1,67	Mike Oldfield	Earth Moving	1989	23	Pop
Accepted because A_2 cannot generate a defense						
Round result:						
C_3	3,77	Mike Oldfield	Hergest Ridge	1974	26	Rock

sends the query to A_1, A_2 and A_3, and they answer returning a case in the order shown in the table (A_2,A_3,A_1). This order is used to decide how to propose the examples and counterexamples.

In the first round, A_2 begins presenting a counterexample to the case C_1 retrieved by A_1. A_q decides to accept the counterexample because A_1 cannot generate any defense by retrieving another counterexample from its case base. This way, C_1 is removed from the initial retrieved set. The following round begins with A_3 having the token. Here it presents a counterexample to the case C_2 retrieved by A_2, but this agent manages to find a defense D_2 that is accepted by A_q. Therefore, the round finishes without changes in the retrieved set. In the third round, A_1 has the token and presents a counterexample for C_2 that is accepted. As this case is removed, the only remaining case is C_3 that is the solution that A_q obtains for its query.

During each round of the argumentation, the fuzzy system participates several times. To illustrate its behavior let's detail the reasoning process of the first round. It begins when A_2 presents a counterexample C_e for C_1. The behavior of this subsystem is shown in the first row of Figure 2. On the left we find the inputs and outputs of the system and on the right we have included a representation of the fuzzy formula with ($rat(C_e) = 0.5$). The output value is $V_c = 9$ and it is the trust measure sent by A_2 to A_q. Then, A_q decides if the counterexample is accepted taking into account its trust in A_1 –the case being rebutted–, A_2 and V_c. This subsystem is shown in the second row of Figure 2 with V_c set to

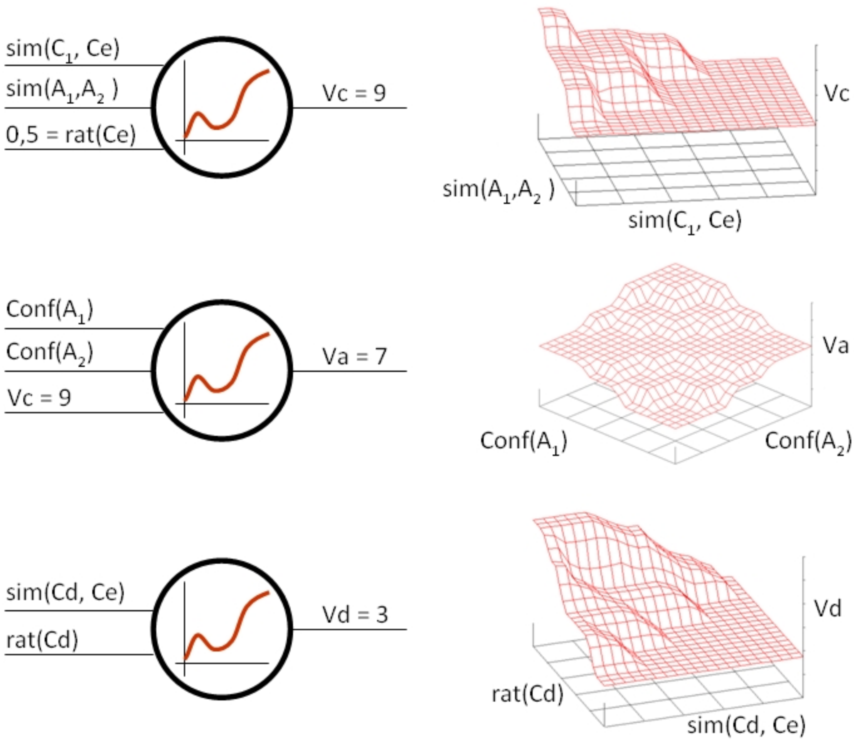


Fig. 2. Example of the behavior of the second, third and fourth fuzzy subsystem

9. Here, the result is $V_a = 7$ and it exceeds the threshold $\alpha > 6.5$ configured in the system. It means that the counterexample is accepted and sent to A_1. Next A_1 has the opportunity of presenting a defense. It looks in its case base but it cannot find any case which $V_d > \beta$ that has been set to $\beta = 5$. This step is represented in the third row of Figure 2. Therefore the first round finishes by removing C_1 from the retrieval set. If A_1 could find a defense, it will be sent –together with V_d– to A_q and a reasoning process similar to the one shown in the second row of Figure 2 would happen.

3 Case Study: Distributed and Collaborative Music Recommender System

In this section we describe a case study in the domain of music recommendation. Music is a classical example where several successful recommender applications have been proposed. The reason is that there are many users interested in finding and discovering new music that would fulfill their preferences. Moreover, the users of this kind of applications tend to interchange recommendations with other users that have similar preferences.

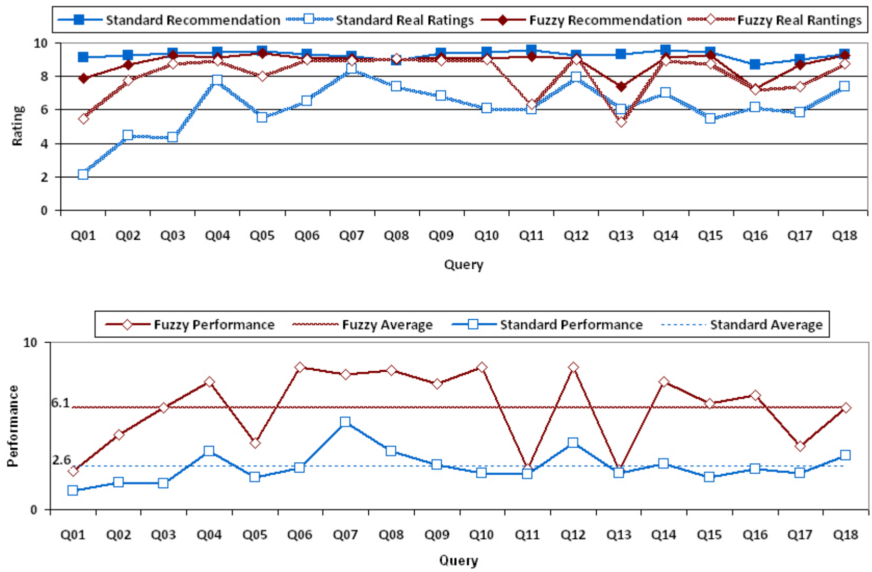


Fig. 3. Results using a social network built with Pearson's coefficients

Relationships between users that make up social networks, reflect the similarity in the preferences of the users and allow them to discover new interesting items. We measure the trust value between two users depending on their corresponding distance in the social network.

The experiments were designed to simulate a real scenario with the highest possible fidelity. As the case base has a catalog of songs, and each user may have a part of this catalog in its internal list of rated items. Every user interacts with its corresponding recommender agent. When a recommender agent receives a query from the user, it forwards the query to the other agents in the system. These agents will use their rated items to recommender songs that fulfill the preferences of the query. Agents are organized according to a social network that ideally reflects the similarity and confidence between users. Our premise is that a real social network will relate users with similar preferences, but this initial intuition was also tested in our experiments by simulating two different networks: a network that relates users with similar preferences and a random social network. To measure the benefits of the architecture of our recommender we used cross-validation over the rated items of a user, by comparing the recommendations performed by a single agent that manages the whole catalog –this is, the real ratings of the user– and the recommendations of our collaborative approach. We have simulated a real social network where we have randomly generated the ratings of the users to control the different factors involved in the experiment. The catalog of songs contains 300 real items. Then we used two numbers of users for every experiment: 10 and 20 users. A local product base of 50 songs was assigned to every user in a random way. It means that these local catalogs may overlap. The ratings of the songs were simulated by assigning preference profiles to the users. Then, ratings were generated using probabilistic distributions associated with every preference of the profile. For example, a user that prefers pop songs will rate higher this kind of items according to certain probability.

The most important component of the experiment is the social network that relates users according to their similarity and mutual confidence. Our initial premise was that a real network will reflect this feature, so we decided to generate the network by linking users with similar preferences. To perform this, we have used the Pearson's coefficient. This metric is a common way for measuring the similarity between users in recommender systems. So, we decided to compute this coefficient between every pair of users in the system and create a network link if a pair has a similarity above certain threshold. However, to test the influence of the topology of the network in the recommendation process we also generated another network with random links. Finally, we used these coefficients to define the confidence level between each pair of users/agents. Figure 3 shows some representative results of our case study. These graphs summarize the ratings obtained with our fuzzy approach compared to the standard AMAL protocol. Here we are raising 18 random queries into a network with 50 users linked by means of the Pearson's coefficients. To measure the performance of the system we used a cross-validation process and compared the ratings for the k best recommended items with the real ratings given by the user to these same items. Ideally, a perfect recommender system will return high rated items according to the preferences of the user. To illustrate this measure, the upper graph of Figure 3 shows the ratings of the recommended items together with the real ratings of the user for these items. Note, that the songs retrieved by both systems (fuzzy and standard) can be different for the same query and it implies that the lines showing the average of the real ratings given by the user for these retrieved sets are different too.

As the graph shows, the original AMAL protocol tends to offer high ratings because it simply chooses the best-rated items from every agent. However, this does not reflect the real ratings given to these items by the user, and therefore the two lines *Standard Recommendation* and *Standard Real Ratings* are quite distant. On the other hand, our fuzzy decision system approximates better the preferences of the user, and the differences with the real ratings are lower meanwhile returning high rated items. This is due to the improvement in the negotiation and decision system thanks to the fuzzy reasoning.

It is important to note that during the experiments we obtained no significant differences when modifying the k value. Moreover we found another noticeable feature: the similarity between the recommended items to the query and the similarity of the corresponding real user-rated items to the query was very close. This result allowed us to leave the similarity aside and concentrate our performance measures on the ratings of the items.

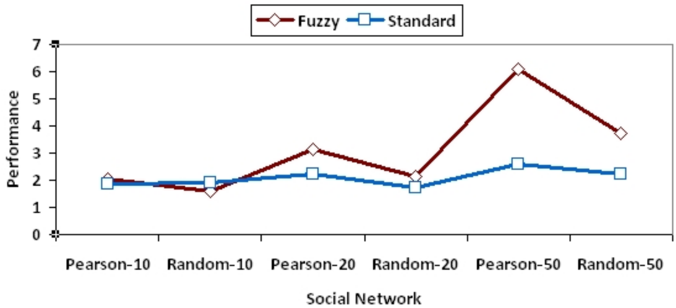


Fig. 4. Global results using different social networks

Because we are trying to maximize the ratings but minimizing their difference with the real ones, we have defined the following performance measure to test our architecture of recommender systems:

$$performance = \frac{rat_{rec}(r)}{1 + |rat_{rec}(r) - rat_{real}(r)|}$$

where r is the set of k items returned by the system, $rat_{rec}(r)$ is the average of the ratings returned by the recommender for r, and rat_{rec} is the average of the real ratings given to r by the user. Figure 3-bottom shows graphically this performance measure for both approaches (fuzzy and original AMAL). Here, a higher value means better performance, and as we can note, the fuzzy approach is always on top. We also include the average of both lines to measure the global performance of the recommendation. These results provide supporting evidence that the fuzzy approach (section 2.2) is significantly better than the original AMAL protocol. Moreover, we have included the performance of the system without any argumentation protocol by issuing the queries to every agent and selecting the top-rated item. Although these ratings are very high, they do not reflect the preferences of the user and the performance of this simple strategy is worse. This fact proves the "ensemble effect" described in [4]: a collection of agents with uncorrelated case bases improves the accuracy of any individual[4].

Finally, we have generated several social networks with different topologies to test the behavior of the architecture of our recommender. Results are shown in Figure 4, where we measure the performance for random and Pearson-based networks with 10, 20 and 50 nodes. Our reasoning process always reports better results with the networks generated using Pearson's coefficients. This fact confirms our premise that social networks will improve the recommendation process. Regarding the fuzzy improvement for the standard AMAL protocol, the Figure shows that the fuzzy approach is better for the networks with 20 and 50 nodes. In the case of the 10-node networks, the fuzzy system is not able to find the minimum similarity relationships between users to perform the decision process

[4] Under some restrictions like the aggregation function (e.g. majority voting) or the retrieval strategy (e.g. content-based k-NN).

correctly. However, this is a coherent and meaningless consequence due to the low number of users in the network.

4 From Individual to Group Recommendations

Although many popular recommender systems are focused on recommending items for individual users, the need of systems capable of performing recommendations for groups of people is getting more interest as there are many activities that are carried out in groups, e.g., going to the cinema with friends, watching TV at home or listening music in the car.

There has recently been a body of work about recommenders that extend their recommendations to groups of users [13]. When moving from individuals to groups many new issues arises. For example, acquiring the preferences of the group, helping the group to decide the more convenient option, or explaining the recommendation to the group. Depending on the size and homogeneity of the group the recommender system has to choose the option that satisfies the biggest number of people taking into account the individual user preferences. As stated in [13] the main approaches to generate a preference aggregation based on the individual user preferences are (a) merging the recommendations made for individuals, (b) aggregation of ratings for individuals and (c) constructing a group preference model.

A group recommender usually manages several subsets of preferences -one per person- that have to be managed independently and combined to create a global recommendation suitable for everyone in the group. According to this, it is natural to think in a distributed architecture like D^{2}ISCO where the preferences of every user are managed independently by an autonomous agent. This agent represents the user inside the global recommender system and is in charge of promoting his preferences when making a recommendation process for the whole group.

Although existing works on group recommender systems are typically based on the aggregation of the preferences of the members of the group where every person in the group is considered as equal to the others, our recent work [17] involves the improvement of current group recommendation techniques by introducing a novel factor: the personality of every individual. Intuitively, when a group of 2 or more friends choose a movie there are some members that are only happy if they impose their opinion, whereas other individuals don't care letting other people decide. Therefore, we have used a personality test to obtain different profiles that interact in different ways when joining a decision making process.

Besides personality, there is another important factor to consider: the structure of the group. We propose a distributed CBR architecture using D^{2}ISCO where each person in the group is represented by an agent and the final recommendation is influenced by the personality of each member of the group and the way they are interconnected through their social relations, basically friendship, defined in the social network. Therefore our method proposes making recommendations to groups using existing techniques of collaborative filtering [18], taking into account the group personality composition and the social connections between the individuals of the group.

Groups of people can have very different characteristics like size and can be made of people with similar or antagonistic personal preferences. It is a fact that when we face a situation in which the concerns of people appear to be incompatible conflict arises. The existing recommender systems for groups typically solve the conflict trying to maximize the preferences of the biggest number of group members. However, the general satisfaction of the group is not always the aggregation of the satisfaction of its members as different people have different expectations and behaviour in conflict situations that should be taken into account. In our approach we distinguish between different types of individuals in a group. This research characterizes people using the Thomas-Kilmann Conflict Mode Instrument (TKI) [19]. The TKI is a test designed to measure the behavior of people in such situations. It is a leader instrument in conflict resolution assessment that is used often by Human Resources and Organizational Development consultants to facilitate learning about how conflict handling styles affect personal and group dynamics. TKI builds a user's profile by means of 30 single choice questions. The test provides scores for the previous five modes (competing, collaborating, etc), representing the preferences of that person when she has to face conflicts. These scores are normalized to obtain percentiles from a 8000 people sample.

TKI characterizes a person behavior in conflict situations along two basic dimensions: assertiveness and cooperativeness. These two dimensions of behavior can be used to define five personality modes of dealing with conflicts: competing, collaborating, avoiding, accommodating and compromising. Our method takes into account these five personality modes. We have performed a first simulation to determine if the group personality composition influences the recommendation accuracy for the group. In fact, we have preliminary concluded that recommendation accuracy is improved for certain types of groups compared with different simple group recommendation algorithms. We have experimentally evaluated the behavior of this method in the Movie recommendation domain using the MovieLens data set with groups of users of different sizes and degrees of homogeneity. The novelty of our approach lies in the use of the member personalities to choose the most interesting movie that would better satisfy the whole group, using the type of personality trait of dealing with conflicts of each member to weight the influence of his/her ratings during the recommendation process.

We have included in the deliberative processes of D^2ISCO the use of real social networks topologies to reflect the real interactions of the users. Our proposal relates also with other works that have employed more social issues in order to include the group member interactions to perform the recommendations. For example, the work described in [20] uses individual satisfaction and the transmission of emotions in order to recommend a sequence of video clips for a group. They consider that a member changes the selection of her best clip according to the clip selected during the previous selection step. This change can be reflected in the recommendation algorithm as an individual satisfaction function that computes the individual affective state. This state influences on the affective state of the other members, getting carried by others emotional state should be taken into account during the group recommendation. Other interesting results from this work are that the most common strategies employed by individuals in real group recommendation are non-dominant strategies such as the average, least misery and average without misery. Additionally, they point out the existence of a tendency where the social status influences on the selection.

4.1 Using the D^{2}ISCO Deliberation Capabilities for Group Recommendations

The development of recommender systems for groups including deliberative capabilities is not a simple task from the software engineering point of view. Such a system involves the management of several users with different preferences that interact remotely to agree on a common decision. This way, we require the infrastructure to build a distributed system where, usually, every user is represented by an agent that manages their preferences. Each agent will be in charge of deliberate with other agents an find a satisfactory solution for the user is representing.

D^{2}ISCO allows including deliberative capabilities in the distributed CBR systems to reach a consensus between the members of the group. Moreover, we have included two novel factors that enables us to improve the recommendation process for groups of users: The personality of every individual and their interconnections.

To include the personality and topology factors in the D^{2}ISCO framework we must split the users into different groups according to their social relationships. Then every group will begin a deliberation process through the reasoning capabilities detailed in Section 2. When every subgroup has found a local solution suitable for its members, a new deliberation process begins where each subgroup is an entity represented by an agent. These new agents will find a global solution that tries to satisfy the preferences of every subgroup.

The reason why this new organization of the structure of the group will affect and improve the result of the recommendation is mainly because with the social network topology we give a more realistic structure and organization of the agents, which is closer to how the argumentations would take place in a real group when they argue on which movie to watch. For a given a group of friends, we draw a network where each node represents a person, and each connection represents that a particular person has a relation with the one he is connected to. Each node discusses his/her opinions with all her neighbors in the network. When two nodes are not connected, it means that the people they are representing don't know each other or that they're not close, this structure is reflecting a face to face discussion where they would have never argue or have to come to a solution between each other. For example if someone brings along his/her boy/girlfriend and the rest of the group doesn't know him, this new member will never start arguing or try convincing someone he has never met before, therefore when we represent this relations we only connect him with the ones that do know him. On the other hand in the all connected network topology every agent will debate with all the rest, no matter if the person he represents knows the other one or not.

Next we summarize our approach:

1. Given: a set of users $U = \{u_1, u_2, \ldots, u_i, \ldots, u_n\}$ (where u_q is the user that launches the query), the set of relationships of an user $rel(u_i)$, and the personality weight $per(u_i)$.
2. Split U into several subsets $S = \{S_1, S_2, \ldots, S_j, \ldots, S_m\}$ where $S_j \subseteq U$.
3. For every $S_j \in S$ compute
   $I_j = obtainSubRecommendation(S_j, rel(u_i \in S_j), per(u_i \in S_j))$
4. Being $I = \{I_1, I_2, \ldots, I_j, \ldots, I_m\}$ the set of items recommended by each subset,
   For every $I_j \in I$
   $w_j = \sum rel(u_i \in S_j).r + \sum per(u_i \in Sj).p$
5. Return the item I_j with the maximum weight w_j

Note that this algorithm implies several processes that must be carefully addressed. Firstly we found the process of splitting the users according to the topology of the social network. In our preliminary experiments we are identifying the k users with more relationships as the "group leaders" and then assigning every remaining user to one of these leaders. Moreover, there is a second way to organize the users: some social networks allow to group users according to the different groups they have joined or by common interests. For example, Figure 5 shows the representation of a real social network in Facebook[5]. This social network allows to classify the users into groups (represented using different areas in the figure, that we have intentionally highlighted). This feature enables us to split the U set into subsets S_j that reflect the relationships of the users in a very realistic manner. Furthermore, Figure 5 serves to confirm our premise that real interactions between users do not follow an all-with-all topology as was illustrated in Figure 1.

After splitting the users, our algorithm applies the method in charge of obtaining the recommendation from every subset S_j. This method is based in the D²ISCO argumentation protocol (detailed in Section 2) that uses arguments and counterarguments in the deliberation process. Finally the global process that combines the partial results (step 4), takes into account the number of relationships inside every subset S_j $(rel(u_i \in S_j))$ and their accumulated personality weight $per(u_i \in S_j)$. The number of relationships is directly obtained from the social network just by counting if two users are friends or not. However, the personality weight is much more difficult to measure. To obtain it, we are using the TKI tests detailed previously. We are summarizing the 5 different modes in just one value. This value is high if the person has a competing and collaborative profile and is low if the person has an avoiding or accommodating

[5] This figure has been obtained using the touch graph application in Facebook. `http://apps.facebook.com/touchgraph/`

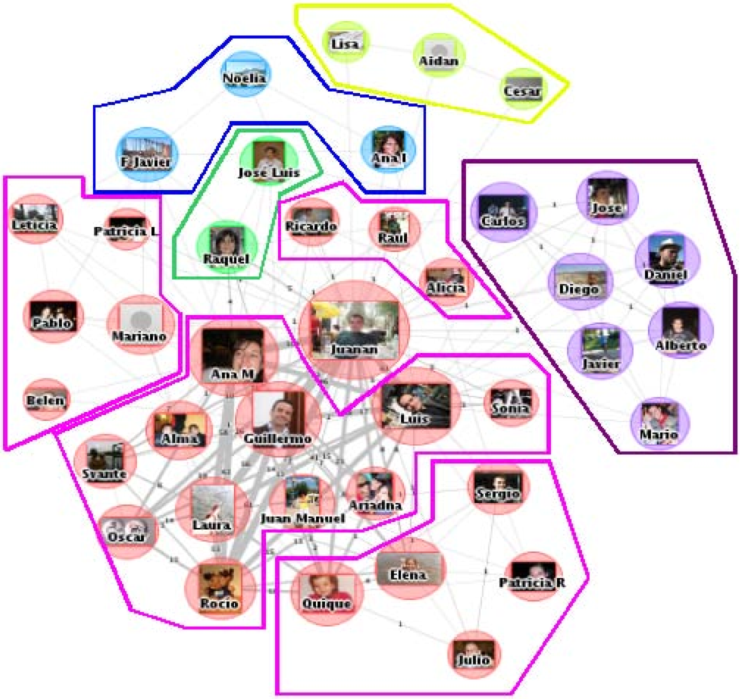


Fig. 5. Users organization in a real social network

personality. Informally, we are giving more importance to the subset that contains more related users and has the highest personality. Parameters r and p are weights for both factors (related users and personality). Parameter r defines how the number of relations of every subset determines the final recommendation. In addition, parameter p weights the impact of the accumulated personality in the recommendation. This accumulated personality is the addition of the personality factor of every user in a subgroup S_j.

4.2 Case Study: Recommender for Groups

To illustrate the advantages of our approach for group recommendations we have ran an preliminary experiment that aids groups of users to choose a movie. A movie recommender system is typically based on rating movies the user has seen and recommending other movies user might like. MovieLens[6] is a good recommendation engine based on collaborative filtering that evaluates user tastes based on ratings to movies seen before. MovieLens is traditionally cited in research works because their developers provide the collected user ratings as a complete anonymous dataset.

[6] `http://movielens.umn.edu/`

Our experiment is based on MovieLens dataset but complemented with a set of 70 students from a AI course at the Universidad Complutense de Madrid. They have filled in some surveys with their personal and group preferences about movies and we used them to develop and test our approach. Our experiment requires the following sources of information:

- $User_i$ ratings for a large set of movies. To obtain a proper rating set we mix a set of ratings extracted from the Movielens Database with ratings obtained from our users . To obtain these ratings we propose our users to rate a list of 50 heterogeneous movies selected from the Movielens data set. In average, users rated 33 movies. Finally we obtain our dataset with a total of 400 users (70 students and the rest anonymous from movielens database) and ≈300k ratings.
- Conflict personality values obtained using the TKI test described.
- Affinity between students to organize groups to run our simulations. We used a test to measure the people affinity to go to the cinema together. This feature represents the social network topology, i.e., if they are friends or not, and then run our simulations using these groups.
- A set of 15 movies that represents the Movie Listings of a cinema. This set was again chosen heterogeneously from the most recent movies in the Movielens Database.
- "Group goes to the cinema" simulation to test our proposal. Our students are joined in groups according to their affinity and they talk for a while to decide which 3 movies they would watch together from the Movie Listings. We also ask these users individually to choose his/her favorite movie from the listings .

With these users and data we have performed an preliminary experiment to test that recommender systems for groups improve accuracy when using the *conflict personality values* and the *social organization* into groups. Further details of this experiment can be found in [17]. Preliminary results indicate that our recommender algorithm obtains better results for groups with people having heterogeneous personalities and having at least a leader in the group. Further systematic experiments need to be done in this area.

5 Conclusions

This paper describes D^{2}ISCO: a framework to design and implement deliberative and collaborative CBR systems. We have performed a case study for a collaborative music recommender system from a catalog of 300 items and networks of up to 50 nodes where the local catalogs overlap. We have presented the results of an experiment of the accuracy of the system results using a fuzzy version of the argumentation system AMAL and a network topology based on a simulated social network. We have generated several social networks using Pearson's relation between ratings. We measure the performance for random and Pearson-based networks with different number of nodes. Our reasoning process and architecture of recommender always reports better results with the networks generated using Pearson's coefficients. This fact confirms our premise that social networks will improve the recommendation process. Regarding the argumentation process we have shown that the fuzzy improvement of the standard AMAL protocol is better with a bigger number of nodes. Besides, although the original AMAL protocol tends to offer high ratings, the fuzzy decision system approximates better the preferences of the user, and the differences with the real ratings are lower meanwhile returning high rated items. This is due to the improvement in the negotiation and decision system thanks to the fuzzy reasoning.

We also present an approach to apply our techniques to group recommendations but including a novel factor: members personality. The novelty of our approach lies in the use of the member personalities to choose the most interesting movie that would better satisfy the whole group, using the type of personality trait of dealing with conflicts of each member to weight the influence of his/her ratings during the recommendation process. We have included in the deliberative processes of D²ISCO the use of real social networks topologies to reflect the real interactions of groups of users. Our case study in the domain of movie recommendations shows that both the social network topology of the agents plus the personality profile of the users are factors than can improve the accuracy of group recommenders compared to classic approaches based on the aggregation of the individual preferences.

D²ISCO has been integrated as a part of jCOLIBRI 2 [5] an established framework in the CBR community. As an ongoing work we are designing a set of reusable templates for collaborative and distributed systems, we are proposing a declarative characterization of this kind of systems based on its observable characteristics, like the number of nodes, topology of the network, number of cases of the local case bases, overlapping between the case bases,and the type of argumentation and reasoning processes. We are evaluating the templates in the context of recommender systems. The recommender domain is specially appropriate to work with social networks topologies, because the recommendations can be biased towards the social environment of each node.

References

1. Leake, D.B., et al.: Case-based reasoning: Experiences, lessons, and future directions. AAAI Press/MIT Press, Menlo Park (1996)
2. Plaza, E., McGinty, L.: Distributed case-based reasoning. Knowledge Eng. Review 20(3), 261–265 (2005)
3. McGinty, L., Smyth, B.: Collaborative case-based reasoning: Applications in personalised route planning. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 362–376. Springer, Heidelberg (2001)
4. Ontañón, S., Plaza, E.: Arguments and counterexamples in case-based joint deliberation. In: Maudet, N., Parsons, S., Rahwan, I. (eds.) ArgMAS 2006. LNCS (LNAI), vol. 4766, pp. 36–53. Springer, Heidelberg (2007)
5. Díaz-Agudo, B., González-Calero, P.A., Recio-García, J.A., Sánchez-Ruiz-Granados, A.A.: Building cbr systems with jcolibri. Sci. Comput. Program. 69(1-3), 68–75 (2007)

6. Díaz-Agudo, B., González-Calero, P.A., Recio-García, J.A., Sánchez, A.: Building cbr systems with jcolibri. Special Issue on Experimental Software and Toolkits of the Journal Science of Computer Programming 69(1-3), 68–75 (2007)
7. Bridge, D., Göker, M.H., McGinty, L., Smyth, B.: Case-based recommender systems. Knowledge Engineering Review 20(3), 315–320 (2006)
8. Grandison, T., Sloman, M.: A Survey of Trust in Internet Applications. IEEE Communications Surveys and Tutorials 3(4), 2–16 (2000)
9. Golbeck, J., Hendler, J.A.: Inferring binary trust relationships in web-based social networks. ACM Trans. Internet Techn. 6(4), 497–529 (2006)
10. McDonald, D.W.: Recommending collaboration with social networks: a comparative evaluation. In: CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 593–600. ACM, New York (2003)
11. Golbeck, J.: Generating predictive movie recommendations from trust in social networks. In: Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) iTrust 2006. LNCS, vol. 3986, pp. 93–104. Springer, Heidelberg (2006)
12. Ziegler, C.N., Golbeck, J.: Investigating interactions of trust and interest similarity. Decision Support Systems 43(2), 460–475 (2007)
13. Jameson, A., Smyth, B.: Recommendation to groups. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 596–627. Springer, Heidelberg (2007)
14. González-Sanz, S., Recio-García, J.A., Díaz-Agudo, B.: D^2ISCO: Distributed Deliberative CBR Systems with jCOLIBRI. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 321–332. Springer, Heidelberg (2009)
15. Ontañón, S., Plaza, E.: An argumentation-based framework for deliberation in multi-agent systems. In: Rahwan, I., Parsons, S., Reed, C. (eds.) Argumentation in Multi-Agent Systems. LNCS (LNAI), vol. 4946, pp. 178–196. Springer, Heidelberg (2008)
16. Zimmermann, H.J.: Fuzzy set theory—and its applications, 3rd edn. Kluwer Academic Publishers, Norwell (1996)
17. Recio-García, J.A., Jimenez-Diaz, G., Sánchez-Ruiz-Granados, A.A., Díaz-Agudo, B.: Personality aware recommendations to groups. In: Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, pp. 325–328. ACM, New York (2009)
18. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22(1), 5–53 (2004)
19. Thomas, K., Kilmann, R.: Thomas-Kilmann Conflict Mode Instrument, Tuxedo, N.Y. (1974)
20. Masthoff, J., Gatt, A.: In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. User Modeling and User-Adapted Interaction 16(3-4), 281–319 (2006)

Chapter 10

Personality and Social Trust in Group Recommendations

10.1 Citation

Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B. Personality and Social Trust in Group Recommendations, in: Procs. of the 22th International Conference on Tools with Artificial Intelligence, ICTAI'10., IEEE Computing Society, Arras, France, pages: 121-126, 2010. ISBN 978-0-7695-4263-8. ICTAI is ranked as CORE B in the CORE ranking.

10.2 Contributions covered by this paper

In this paper we have studied how to compute a factor that reflects users' personality in conflict situations and identified the social variables that influence peoples' trust in order to compute a trust factor that reflects the tie strength between users. We have later use these two social factors in a social recommendations system in the movies domain and proved that the inclusion of these factors improves other recommenders that do not use social information.

2010 22nd International Conference on Tools with Artificial Intelligence

Personality and Social Trust in Group Recommendations

Lara Quijano-Sánchez, Juan A. Recio-García, Belén Díaz-Agudo
Universidad Complutense de Madrid, Spain,
email: {lara.quijano, jareciog,}@fdi.ucm.es, belend@sip.ucm.es

***Abstract*—In this paper we describe some new ideas to improve recommendations to groups of people. Our approach maximizes the global satisfaction for the group taking into account people personality and the social relationships among people in the group. We present some results with two cases of study based on the movie recommendation domain with heterogeneous groups. The first case study uses synthetically generated groups of people to test how the group composition affects the accuracy of the recommendation. Our second case study uses real users and groups where the topology of the groups is based on a social network. This second case of study with real users confirms the wide conclusions of the preliminary experiment with synthetic data, which allows us to conclude that it is possible to realize trustworthy experiments with synthetic data.**

I. Introduction

Recommender systems represent a wide range of applications with a raising impact in the current web [1], [2]. Although most of the most popular recommender systems are focused on recommending items for individual users, the need of systems capable of performing recommendations for groups of people is getting more interest as there are many activities that are carried out in groups, like going to the cinema with friends, watching TV at home or listening music in the car. Our recent work [3] involves the improvement of current group recommendation techniques by introducing a novel factor: the personality of every individual. Intuitively, when a group of friends chooses a movie there are some members that are only happy if they impose their opinion, whereas other individuals do not care letting other people decide. Therefore, we have used a personality test to obtain the different roles that people play when interacting in a decision making process and studied how individual personalities influence the results and the satisfaction for the whole group.

Besides personality, this paper introduces the novelty of taking into account the social structure of the group to influence the recommendation process. Our approach reflects in a realistic way the relationships between groups of users connected in a social network. These relationships are measured through social factors, like the distance in the social network or the number of common friends, that are extracted from the social networks and that are used to compute the trust values between the members of the group.

In this paper we describe our theories about making recommendations for groups of people with different personalities and connected through social network structures. Our method proposes making recommendations to groups using existing techniques of collaborative filtering [4], taking into account the group personality composition and the social connections between the individuals of the group. We have tested our method in the movie recommendation domain using two test datasets. The first case study uses synthetically generated data to create simulated groups of people to test how the group composition affects the accuracy of the recommendation. Our second case study uses real users and groups where the topology of the groups is based on a social network. This second case of study with real users confirms the wide conclusions of the preliminary experiment with synthetic data, which allows us to conclude that it is possible to realize trustworthy experiments with synthetic data.

The paper runs as follows: Section II describes existing techniques to obtain recommendations for groups and gives some details of our personality aware recommendation process and our proposal for including social network topologies in the decision process. Section III explains our case of study: Movie Recommendation. Section IV shows the results of our experiment. Section V concludes the paper and presents the main lines of future work.

II. Recommendation to groups

Recommender systems have traditionally recommended items to individual users, but there has recently been a body of work about recommenders that extend their recommendations to groups of users [2]. When moving from individuals to groups many new issues arises. For example, acquiring the preferences of the group, helping the group to decide the more convenient option, or explaining the recommendation to the group. Depending on the size and homogeneity of the group the recommender system has to choose the option that satisfies the biggest number of people taking into account the individual user preferences. As stated in [2] the main approaches to generate a preference aggregation based on the individual user preferences are (a) merging the recommendations made for individuals, (b) aggregation of ratings for individuals and (c) constructing a group preference model.

The most common employed approaches in group recommenders are (b) and (c). They are used in many different fields like selecting the background music of a fitness center [5] recommendation of video clip sequences [6], movies [7] among others. These strategies usually try to maximize the "average satisfaction" of the group. The work described in [7] recommends movies for groups based on the inferred

1082-3409/10 $26.00 © 2010 IEEE
DOI 10.1109/ICTAI.2010.92

ratings by MovieLens and using the "least misery" strategy to generate the preference aggregation. This strategy supposes that a small group of people will be as happy as its least happy member. The work in [8] criticizes the aggregation strategies like the one employed in PolyLens because they claim that these strategies combine the ratings always in the same way without considering how the members in the group interact with each other.

A. Personality Aware Recommendation to Groups

Most of the previous works in group recommendation consider the preferences of every member of the group with the same degree of importance and try to satisfy the preferences of every group member. However, groups of people can have very different characteristics like size and can be made of people with similar or antagonistic personal preferences. It is a fact that when we face a situation in which the concerns of people appear to be incompatible a *conflict situation* arises. Our approach determines that the general satisfaction of the group is not always the aggregation of the satisfaction of its members as different people have different expectations and behaviour in conflict situations that should be taken into account. In [3] we have presented a method for recommendation to groups where we distinguish between different types of individual personalities in a group. Our research characterizes people using the Thomas-Kilmann Conflict Mode Instrument (TKI) [9] that describes a person behavior in conflict situations along two basic dimensions: assertiveness and cooperativeness. These two dimensions of behavior can be used to define five personality modes of dealing with conflicts: competing, collaborating, avoiding, accommodating and compromising. We had our users fill up the TKI test to determine their personality, once each user u_i completes it, we calculate a value that represents how selfish or cooperative they are, we call this value the (CMW_i) (*Conflict Mode Weight*,), it represents the predominant behaviour in that particular user according to his TKI evaluation. We note that people with strong personality have a high CMW value, while a CMW value represents an easy going personality. The CMW function returns values in a range of [0,1], being 0 the reflection of a very cooperative person and 1 the reflection of a very selfish one. We studied how the group personality composition influences the recommendation accuracy for the group, and how it is improved for certain types of groups compared with different simple group recommendation algorithms.

Our approach creates a group recommendation by mixing individual recommendations. The individual recommender is a collaborative system based on the ratings of other users. Every user rates movies that (s)he has seen before, and the recommendation process consists comparing her/his ratings with the rating of other users to obtain her/his most similar users (the ones that rated the same movies in a similar way). Finally, the ratings of the most similar users are used to infer unknown ratings in movies and to create a new recommendation.

Our proposal uses the type of personality to weight the influence of his ratings during the recommendation process. If we consider $ir_{i,m}$ the rating of a given user u_i to a certain product m we can see in the following algorithm to compute *personality based recommendations* (pbr(u,m)), this is, how the personality can be taken into account with the aggregation function of average satisfaction.

$$pbr(u_i, m) = \frac{1}{|G|} \sum_{j \in G \wedge j \neq i} (ir_{i,m} + pd(u_i, u_j)) \qquad (1)$$

$$\text{where } pd(u_i, u_j) = (CMW_j - CMW_i) \cdot \alpha$$

represents the personality difference, α value has been experimentally selected and it is employed to modify the impact of the personality differences on the modified rating, $|G|$ represents the number of components of the group and CMW_j is the conflict mode weight of the user j.

What we observe is the difference between the personality pairs in the individuals of the group. It is based in a modification on the average satisfaction aggregation method. This strategy reflects that assertive characters will have more influence in the average satisfaction than the cooperative characters. This factor is computed as the distance of CMW values in the personality difference function $pd(u_i, u_j)$. We can find the algorithms applied for other merging functions like least misery in [3].

In this paper we use this personality based recommendation method in a first experiment using simulated users and groups with different features to see how the personality composition of the member of the groups affects the recommendation results. Then, in a second experiment we compare these results with real users and groups of people to test the accuracy of our simulated dataset. Before presenting the case study we describe how our approach consider the group structure into the recommendation process.

B. Social Trust in Recommendation to Groups

Current research has pointed out that people tend to rely more on recommendations from people they trust (friends) than on recommendations based on anonymous ratings [10]. This factor is even more important when we are performing a group recommendation where users have to decide an item for the whole group. This kind of recommendations usually follows an argumentation process, where each user defends his preferences and rebuts other's opinions. Here, the trust between users is the major factor when users must change their mind to reach a common decision.

A promising approach is to collect the trust knowledge from existing social networks. Social networks have been one of the most important topics in the last few years, with nets like Facebook, Twitter and MySpace, among others. The use of social networks and trust when building a recommender system is not new [11], [12]. One contribution of our approach is the use of generic social network topologies to reflect the real interactions of the users in the group. Our working hypothesis is that instead using the topology "all connected", which is the default topology used in typical group recommendation approaches, with the social network topology we give a more realistic structure and organization of the group, which is closer to how the argumentations would take place in a real group when they argue on which movie to watch. Our main goal is to improve the recommendations by taking into account both, the network topology and the group personality composition. Moreover, our goal is to identify which are the most important factors of the social networks that must be taken into account when computing the trust between users. Examples of these factors are: the number of shared messages, common pictures, direct friends, etc. To perform this task we review several existing works [13], [14] and select the most relevant and feasible factors. These factors are detailed in Section III-B and their impact is reported in Section IV.

III. Case of Study: Movie Recommendation

We have evaluated our method in the movie recommendation domain. We have initially realized our experiment with synthetic data because we wanted to explore extreme cases that could appear in conflict situations. We wanted to have control of the distribution of the data, which didn't happen if we used real data. This synthetic data let us explore every group composition and personality distribution within the group. It also lets us reproduce the behaviour of large groups that are very difficult to organize in experiments with real users. Next we performed a second experiment using real data in order to verify the results that we obtained first. This experiment with real users confirms the wide conclusions of the preliminary experiment with synthetic data.

We built the recommender systems using the jCOLIBRI framework [15]. jCOLIBRI is currently a reference platform in the Case Based Reasoning (CBR) community and includes an extension to build recommender systems.

A. Experimental set-up with synthetic people

Our experiment runs as follows: (1) we generated randomly many groups of users with different personality profiles and social topology; (2) we developed an individual recommender following the collaborative approach; (3) we created 3 different group recommender systems that use the individual system: a standard recommender that only aggregates preferences; a group recommender using only the personality profile, as explained in section II-A; and a final one that takes into account the personality and social topology, as reflected in equation 2. Finally, (4) we have compared the results obtained with different recommenders and different synthetic group configurations, because we wanted to study if these configurations affected the final recommendation.

These 4 group recommenders use the formula shown in equation 2. Depending on the recommender we set up the value of some parameters to 0. The baseline of our experiment is an standard recommender without personality or social factors (referred as *Base* in the results). Next, the group recommender that only takes into account the personality always uses a trust function that always returns 0 (we refer to this recommender as *Personality*). Finally, our complete *social group recommendation method* (referred as *Personality & Trust*) uses the complete equation:

$$sgr(u_i, m) = \frac{1}{|G|} \sum_{j \in G \wedge j \neq i} (ir_{i,m} + pd(u_i, u_j) + trust(u_i, u_j)) \tag{2}$$

In order to simulate the social network inside each group we have randomly generated friendship links between users. We have defined a *trust function* that analyzes these links to compute the trust among users depending on their distance inside the social network. As we present in Section III-B, the second experiment computes the trust value using several factors obtained from the real social network users belong to. However, we have chosen a simple approach in this initial experiment to infer trust along the network (as users are synthetically generated):

$$trust(a, b) = \begin{cases} 1 & \text{if } distance(a, b) == 1 \\ 0.5 & \text{if } distance(a, b) == 2 \\ 0 & \text{a.o.c.} \end{cases}$$

To generate the group of users we have used a set of 100 people. Every person was assigned to a random value CMW to reflect their personality –in works with real data values, this is the value that was computed using the results of the TKI personality test [9]. This value is employed in function CMW_i that will return the personality weight of every individual. We basically define five different types of personality according to this range: very selfish, selfish, tolerant, cooperative and very cooperative. For example if we consider a selfish person his CMW value must be contained in a range of [0.8,1.0]. When we realized the TKI test to real users, like in [3], there were some of these ranges that were unexplored because people usually don't have such extreme personalities, this is why decided to use the synthetic data. To be able to study the effects of the different types of personalities we generated 20 users for each type of personality. We grouped users in sets of 3, 5, 10, 15, 20 and 40 people. For each group size we selected the components of the group so that the *personality distributions* presented all the possible combinations: groups of very selfish, selfish, tolerant, cooperative, very cooperative, very selfish & very cooperative, very selfish & tolerant, ... and so on until we reach 13 possible combinations. In the end we had 76 groups (13 different distributions for each size, except for the 40 people group where we only had 11 combinations due to the resemblance of personalities in such big groups).

The next element required by our experiment is the evaluation function to measure the accuracy the recommendation. We have to figure out which movies would each of our users have chosen individually from a movie listing of a cinema, and afterwards determine which of that movies the group as a whole would have finally decided to watch. Our evaluation function is based on the content of movies. As we have explained the recommender systems we are evaluating are rating-based (collaborative). It implies that the content or features description of the recommended item is not taken into account during the recommendation process. Instead collaborative filtering there is an alternative recommendation technique called content based recommendation [1] that compares the features of new items to the items already selected by the user, and sorts them according to their similarity following the typical retrieval algorithms in CBR. We use this kind of content-based technique in the evaluation function to figure out which movies each user likes.

We selected a list of 50 heterogeneous movies from the MovieLens data set [16] and we rated them (with a range of 0.0 to 5.0) for each user according to a random profile we assigned. These profiles were constructed according to typical preferences in movies of real life people according to their age, sex and preferences. For example, the ratings that a user with a childish profile would give were very high ratings to animation, children or musical movies and very low ratings to drama, horror, documental, etc.

Afterwards, for each user we obtained which movies would be rated with 4.5 or more and define the set of favourites movies for each user. With this information we used a content-based similarity test to organize the listing of the cinema in order of preference. We chose the top 3 and marked them as the individual favourites if. Secondly we needed to obtain the decision of the group. Now that we knew which movies would the individual users argue for, we reproduced a real life situation were everyone discussed their preferences, taking into account the personalities and the friendship between them and then we finally obtained the real favourite movies for the group rgf. We use this information to evaluate the accuracy of our recommender by comparing how many of the first three recommended movies -the proposed group favourites gf- belong to the rgf set of that group.

B. Experimental Setup with real users

Although the first experiment with synthetic data let us explore extreme group configurations, we required a second experiment with real users to validate the obtained results (see Section IV). In order to perform our second experiment in the movie recommendation domain with real users, we create two events in two different social networks, Facebook[1] and Tuenti[2]. In these events we ask some of our friends in the social network for completing three questionnaires[3]. The first questionnaire serves to obtain the personality profile by asking the 30 questions from the TKI personality test [9]. Second questionnaire gets the individual preferences of the user about cinema. Users evaluate 50 heterogeneous movies from the MovieLens data set [16](rating them with a range of 0.0 to 5.0). This way, we can compute the collaborative filtering algorithm to compute the individual predictions $ir_{i,m}$. Finally, third test asks users to choose their 3 favorite movies from a list of 15 recent movies (of the 2009 year), that represents a movie listing from a cinema. This movie listing was chosen heterogeneously from the MovieLens database. These movies are the ones they would actually like to watch or had enjoyed best. The goal of this test is to measure the accuracy of the individual recommender. The answers to these questionnaires are analyzed to define the user profile of each participant. 58 real users have participated in our experiment. To measure the accuracy of the group recommendation we create groups with our participants and we ask them to simulate that they are going to the cinema together. We provide them the 15 movies that represent our movie listing and we ask them to choose which 3 movies they actually would watch together. We manage to gather 15 groups of 9, 5 or 3 members. The three movies that each group chooses are stored as the *real group favourites* set *rgf*. This way, to evaluate the accuracy of our recommender we can compare the set proposed by the recommender –the *gf* set– with the real preferences *rgf*. We measure the number of movies in *gf* that are also in *rgf*. Once we have the tests, we need the personality and trust factors. To obtain the personality value, we calculate the CMW_i (*Conflict Mode Weight*) from the results of the first test given to the users. To compute the trust factor we reviewed several existing related works [13], [14] to decide which are the specific factors that must be taken into account. We have selected 10 factors that are combined to get a final trust value. Moreover, we have evaluated (see Section IV) which factors have the highest impact in the recommendation process. The specific trust factors obtained from the social network are:

- f_1: Distance in the social network.
- f_2: Number of common friends.
- f_3: Intensity of the relationship: how often they write each other on their walls.
- f_4: Intimacy of the relationship: We classify relationships by finding keywords that represent different intimacy levels.
- f_5: Duration: how long they know each other.
- f_6: Reciprocal services: number of posted videos/-songs/webs, shared games/ applications.
- f_7: Structural variable: common interests described in the users profile like movies, books, or general interests.
- f_8: Social distance: how many of the following properties are shared: political, educational, religious and demographical information.
- f_9: Status: Value depending on the kind of status: couple, family, best friends..etc
- f_{10}: Pictures: Percentage of of pictures where they appear together.

The final trust value $trust(u_i, u_j)$ is a weighted average of the previously described factors:

$$trust(i,j) = t_{ij} = \sum_{k=1}^{10} \alpha_k \cdot f_k(u_i, u_j) \tag{3}$$

These weights have been experimentally obtained using a genetic algorithm (GA). Our GA manages a population of vectors of weights (α_i). These vectors can be combined and mutated. The fitness function to maximize is the group recommendation accuracy.

[1] http://www.facebook.com
[2] http://www.tuenti.com
[3] Questionnaires are accessible at http://www.lara.warhalla.com/(spanish)

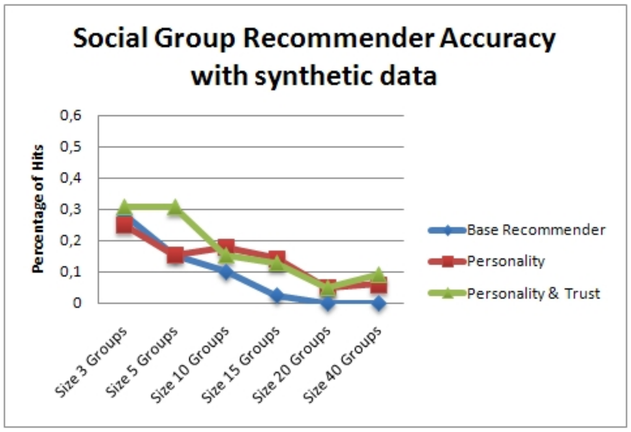


Fig. 1. Results for synthetically generated users.

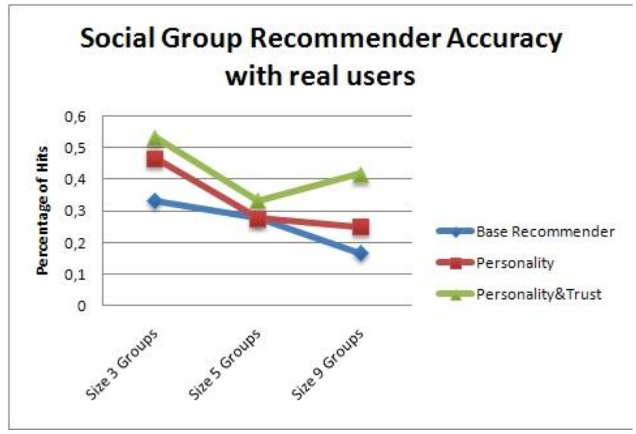


Fig. 2. Results for real users.

Again, we build three different recommenders: (1) *Base* is a standard group recommender using the standard average satisfaction aggregating function; (2) *Personality Aware Recommender* only uses the personality data; and (3) *Personality & Trust* recommender takes into account both the social trust and the personality of each member. Next section details the results of our experiment.

IV. Evaluation Results

As the individual recommender has been only used as a base line for the group recommender we do not discuss its performance. However, let's notice that by improving this system the whole system will improve because the whole group recommender system is based on the individual preferences of each user. Another important factor in the recommender's performance is the configuration of the set of 15 movies that conform the movie listing of the cinema and the similarity of these movies to the set of 50 movies that the users had rated. These data was randomly chosen, so the performance of the recommender may change depending on the values.

The figures that analyze the different strategies of aggregation show three different lines that represent the results of the recommender when considering their friendship relations and their personality–*Personality & Trust* item–, just their personality –*Personality* item–, and just the simple base group recommender system *BaseRecommender*.

Figure 1 summarizes the results for synthetically generated users. This figure shows better results when combining personality and social trust in the group recommendation process. Therefore, we can conclude that recommender systems for groups could be improved when using a social factors.

Regarding the size of the groups we can clearly conclude that our recommender algorithm obtains better results for small groups than for big groups. It reflects the real world because with more people there are more different opinions and it is more difficult to arrive to a consensual solution.

On the other hand, when comparing how the distribution of the personalities in the group affected the percentage of hits, in Figure 3 we can see that the percentage was higher around

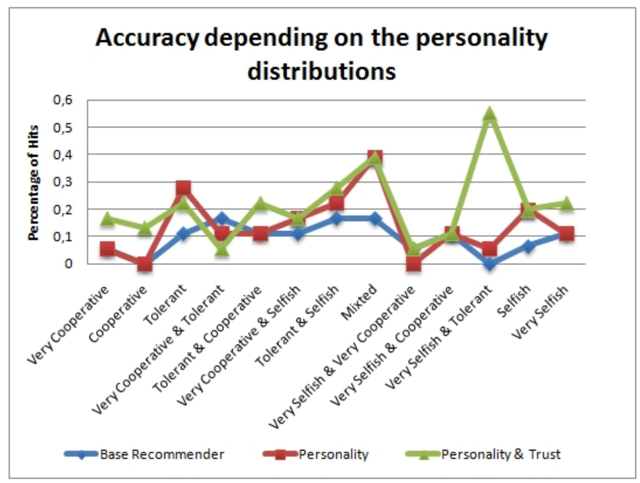


Fig. 3. Results depending on personality distribution.

the center of the graphic, which matches with the groups that have more varied personalities. Therefore heterogeneously in the personality of the group implies a better performance in the recommender. We can also observe that the right side of the graphic, representing groups of people with at least one leader role, has better percentage than the groups situated on the left representing groups of very cooperative people, were no one will impose their opinion.

Regarding the experiment with real users, Figure 2 shows the performance of this experiment (when considering personality and social factors, just personality and the simple base group recommender). Figure 2 summarizes the average results taking into account all the groups. We conclude that our *social group recommendation* method, explained in section II-A, obtains the best results. This figure confirms our theories and shows that we improve the recommendations when taking into account the social trust and the personality of each individual. It also reflects a similar behaviour when comparing these results with the synthetically generated data of Figure 1. In both Figures the *personality based recommendation* works better

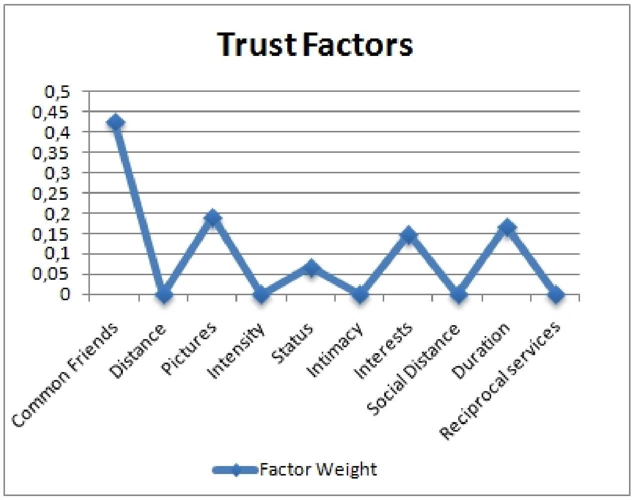


Fig. 4. Importance of the trust factors

than the standard recommender but worse than the *personality plus trust recommendation*. This second experiment only takes into account small and medium groups due to the evident impossibility of reproducing the experiment with large groups that are not real in recommendation processes.

Regarding the importance of each of the factors that conform the trust value, we can see in Figure 4 how they are taken into account in order to maximize the performance of our recommender. These weights (α_i) were obtained using the Genetic Algorithm, explained in section III-B. We can see that the most relevant is the number of friends in common, followed by the pictures, the duration, the common interests and the status. We think this experimental result is very relevant and should be taken into account when implementing real applications. Besides it relates with the used factor of friendship used in the experiment with synthetic data.

Finally, we have also studied if the accuracy of the recommendation was linked to the distribution of the personalities trying to compare it with the previous results with synthetic data. To do so, we calculated the standard deviation of the personality values for each group. After running several statistical studies, we concluded that there is not correlation at all between the distribution of the personalities of the group and the accuracy of our algorithms. This is due to the small deviation in real group of users, this is: normal groups are usually conformed by mixed personalities.

V. Conclusions and Future Work

In this paper we have proposed and experimented with a novel method of making recommendations for groups taking into account the group personality composition and the social structure of the group. We have shown that personality profiles and the social relationships between the users improve the accuracy of the recommendation for a group of people.

We have tested our method in the movie recommendation domain using two test datasets. The first experiment uses synthetically generated data to create simulated groups of people to test how the group composition affects the accuracy of the recommendation. We have also performed another experiment with real data, where we created two events in two different social networks, Facebook and Tuenti. In both experiments we have used groups of different size and personal preferences, where we have proved that by introducing the trust factor and the personality awareness we do improve the results of the recommendations. Our working hypothesis is that the personality and the social organization of the structure of the group will affect and improve the result of the recommendation, mainly because with the social network topology we give a more realistic structure and organization of the group. Our experiments confirm our hypothesis. One main conclusion of this paper is that it is possible to realize trustworthy experiments with our synthetically generated data as the second case study with real users confirms the wide conclusions of the preliminary experiment with synthetic data. As future work we are integrating our algorithms in the jCOLIBRI framework and we are creating a system with memory of the previous recommendations as a necessary step when providing a whole set of recommendations.

References

[1] Bridge, D., Göker, M.H., McGinty, L., Smyth, B.: Case-based recommender systems. Knowledge Engineering Review **20** (2006) 315–320

[2] Jameson, A., Smyth, B.: Recommendation to groups. The Adaptive Web **LNCS 4321** (2007) 596–627

[3] Recio-García, J.A., Jimenez-Diaz, G., Sánchez-Ruiz, A.A., Díaz-Agudo, B.: Personality aware recommendations to groups. In: Procs. of the 2009 ACM Conference on Recommender Systems, ACM (2009) 325–328

[4] Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Communications ACM **35** (1992) 61–70

[5] McCarthy, J.F., Anagnost, T.D.: Musicfx: An arbiter of group preferences for computer aupported collaborative workouts. In: CSCW. (1998) 363–372

[6] Masthoff, J., Gatt, A.: In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. User Modeling and User-Adapted Interaction **16** (2006) 281–319

[7] O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: A recommender system for groups of user. In: ECSCW. (2001) 199–218

[8] Chen, Y.L., Cheng, L.C., Chuang, C.N.: A group recommendation system with consideration of interactions among group members. Expert Syst. Appl. **34** (2008) 2082–2090

[9] Thomas, K., Kilmann, R.: Thomas-Kilmann Conflict Mode Instrument. Tuxedo, N.Y. (1974)

[10] Sinha, R.R., Swearingen, K.: Comparing recommendations made by online systems and friends. In: DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries. (2001)

[11] Golbeck, J.A.: Computing and applying trust in web-based social networks. PhD thesis, College Park, MD, USA (2005)

[12] Avesani, P., Massa, P., Tiella, R.: A trust-enhanced recommender system application: Moleskiing. In: SAC '05: Proceedings of the 2005 ACM symposium on Applied computing, NY, USA, ACM (2005) 1589–1593

[13] Gilbert, E., Karahalios, K.: Predicting tie strength with social media. In: CHI '09: Procs of the 27th Int. Conf. on Human factors in computing systems, New York, NY, USA, ACM (2009) 211–220

[14] Golbeck, J.: Combining provenance with trust in social networks for semantic web content filtering. In: Provenance and Annotation of Data, Int. Provenance and Annotation Workshop, IPAW 2006. Volume 4145 of LNCS., Springer (2006) 101–108

[15] Recio-García, J.A., Díaz-Agudo, B., González-Calero, P.A.: Prototyping Recommender Systems in jCOLIBRI. In: Proceedings of the 2008 ACM conference on Recommender Systems, NY, USA, ACM (2008) 243–250

[16] Bobadilla, J., Serradilla, F., Hernando, A.: Collaborative filtering adapted to recommender systems of e-learning. Knowl.-Based Syst. **22** (2009) 261–265

Chapter 11

Happy Movie: A Group Recommender Application in Facebook

11.1 Citation

Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B, Jimenez-Díaz, G. Happy Movie: A Group Recommender Application in Facebook. in: Procs. of the 24th International Florida Artificial Intelligence Research Society Conference, FLAIRS. Palm Beach, Florida, USA, AAAI Press, pages 419-420, 2011. ISBN 978-1-57735-501-4. FLAIRS is ranked as CORE C in the CORE ranking.

11.2 Contributions covered by this paper

In this paper we have introduced our social group recommender application, *HappyMovie*. In this application we have switched from theoretical experimentation to the possibility of testing our social group recommendation methods in a real life scenario, the social network Facebook. Having our recommender enbeded in a real social network allows us to elicit the social factors in a more efficient and dynamic way.

Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference

Happy Movie: A Group Recommender Application in Facebook

Lara Quijano-Sánchez, Juan A. Recio-García, Belén Díaz-Agudo, and **Guillermo Jiménez-Díaz**
Dep. Ingeniería del Software e Inteligencia Artificial
Universidad Complutense de Madrid, Spain
email:{lara.quijano, jareciog, gjimenez}@fdi.ucm.es; belend@sip.ucm.es

Abstract

In this paper we introduce our recommender *Happy Movie*, a Facebook application for movie recommendation to groups. This system exploits information about the social relationships and behaviour of the users to provide better recommendations. Our previous works have shown that social factors improve the recommendation results. However it required many questionnaires to be filled for obtaining the social information, so we have moved to a social network environment where this information is easily available.

1 Introduction

Recommender systems were born from the necessity of having some kind of guidance when searching through complex product spaces. These systems share many features and methods with Case-Based Reasoning (CBR), as product recommendations can be seen as a kind of experience. More precisely, group recommenders are built to help groups of people who share a common activity decide in conflict situations.

Our previous works (Recio-García et al. 2009; Quijano-Sánchez, Recio-García, and Díaz-Agudo 2009; 2010) presented our approach, named GRUPITO, of making recommendations for groups of people based on three important features: personality, social trust and memory of past recommendations. This way we simulate in a more realistic way the argumentation process followed by groups of people when deciding a joint activity. Although our theories for making recommendations to groups have been proven in simulated environments, this paper presents the instantiation of our model in a real-life scenario: the social network Facebook.

2 Facebook application: Happy Movie

Happy Movie is a Facebook application for recommending movies to a group of users. Although this application has been initially designed for the recommendation of movies, it is important to note that our proposal can be easily adapted for other domains. In order to use our application, users only have to start their Facebook account and look for *Happy Movie* in the applications section. The required steps to obtain a movie recommendation for the group with Happy movie are explained below:

- Creating the user profile in the application: This profile is based on three different aspects: personality, individual preferences and trust to other users. To obtain the personality users must complete a personality test. Later they have to rate a set of movies, where we obtain their personal preferences. Finally, to obtain the last factor – trust– the application explores the information stored in the Facebook personal profile. It calculates the trust that the user has with all the other users that have joined the event up to now.

- Creating the activity: The organizer decides to create an activity and she starts the application to create a new event. Organizers must establish some data like place, date or invited users. Any user attending the event can see the date and place of the event and a proposal of three movies, which are the best ones that our group recommender has found for the current group of users attending the event. When users participate in the event they are also able to invite any Facebook friends they wish and they can retire from the event at any time.

- Recommendation: When the application has obtained the three factors that identify each user that joins the event (personality, individual preferences and trust between other users) it provides a group recommendation using the method explained in (Quijano-Sánchez, Recio-García, and Díaz-Agudo 2010).

- Having the recommendation made: When the event is created it looks up for the current movie listing from the selected place and provides a list of 3 recommended movies. This list is automatically updated every time a user joins the event or retires from it. When the expiration date is reached users can see the final movie list. In that moment they are allowed to vote each movie individually. This process lets us decide which movie they are finally watching and, more important, it provides the required feedback to evaluate the quality of our recommendation.

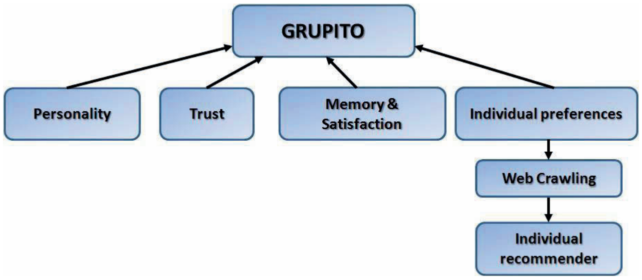


Figure 1: Facebook application architecture

3 A modular architecture for group recommendations

The architecture of *Happy Movie* is represented in Figure 1. We can see that the application is divided into seven different modules: *Personality, Trust, Memory & Satisfaction, Individual preferences, Web Crawling, Individual recommender* and *GRUPITO*. This section details these modules.

- Personality Module: This module fulfils the task of obtaining a value that represents the personality of each user. To do so, each user must answer to a personality test that measures people's behaviour in conflict situations. Users will only have to do this test the first time that use *Happy Movie*.
- Trust Module: Once *Happy Movie* is running, the trust module must perform its estimation every time a user joins the event. When this happens the trust module explores the users who are currently in the list of attending users and calculates the trust of each of them with the user who has recently joined the event.To do so, the profiles of the two users will be analysed, comparing different social factors. A detailed explanation of the trust factors obtained from Facebook and the combination process is provided in (Quijano-Sánchez, Recio-García, and Díaz-Agudo 2010).
- Memory & Satisfaction Module: In this module we store all the recommendations made for every user and every group. Having recommendations with memory allows our system to avoid repeating previous recommendations, and it ensures a certain degree of fairness. If one member accepts a proposal that she is not interested in, next time she will have some kind of preference, so that in the long run all the members of the group are equally satisfied.
- Individual preferences Module: It consists on a test of the individual preferences of each user in the application's domain. These preferences are stored as the individual case base of each user.
- Web Crawling Module: This module searches the web and finds the current movie listing, then it searches the complete file of every movie in it. The retrieved movies, with all their specific information, are sent to the individual recommender module and to the GRUPITO module as they are the products to be recommended.
- Individual Recommender Module: Our group recommendation strategies combine individual recommendations to find an item (movie) suitable for any user in the group.
- GRUPITO, Group Recommendations Using Personality, Interests and Trust Organizations: We include the personality and trust factors in the group recommendation method. The main ideas of these approach is detailed in (Quijano-Sánchez, Recio-García, and Díaz-Agudo 2010).

4 Conclusions

In this paper we have introduced our Facebook application Happy Movie. In our previous works (Recio-García et al. 2009; Quijano-Sánchez, Recio-García, and Díaz-Agudo 2009; 2010) we presented a standalone group recommender that uses a method based on the personality of every user and the trust between users. Now, we have moved this standalone recommender to an application in a social network where we can benefit from the information stored in it. Embedding the application in Facebook also makes it more reachable to everybody who has an account in it, so our users can easily benefit from our services of recommendation.

5 Acknowledgments

Supported by Spanish Ministry of Science & Education (TIN2009-13692-C03-03) and Madrid Education Council and UCM (Group 910494). We also thank the friends and students who have participated in the experiment.

References

Quijano-Sánchez, L.; Recio-García, J.; and Díaz-Agudo, B. 2009. Social based recommendations to groups. In *Procs. of the 14th UK Workshop on Case-Based Reasoning*, 46–57. CMS Press, University of Greenwich.

Quijano-Sánchez, L.; Recio-García, J. A.; and Díaz-Agudo, B. 2010. Personality and social trust in group recommendations. In *Procs of the 22th International Conference on Tools with Artificial Intelligence, ICTAI'10.*, 121–126. IEEE Computing Society.

Recio-García, J. A.; Jimenez-Diaz, G.; Sánchez-Ruiz, A. A.; and Díaz-Agudo, B. 2009. Personality aware recommendations to groups. In *Procs. of the 2009 ACM Conference on Recommender Systems*, 325–328. ACM.

Chapter 12

Using personality to create alliances in group recommender systems

12.1 Citation

Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B. Using personality to create alliances in group recommender systems in: Procs. of the 19th International Conference on Case Based Reasoning, ICCBR'11. Springer-Verlag Berlin Heidelberg, Greenwich, London, pages: 226-240, 2011. ISBN 978-3-642-23290-9. ICCBR is ranked as CORE C in the CORE ranking.

12.2 Contributions covered by this paper

In this paper we have studied a new technique of using social factors in group recommender systems by using these social factors to identify leaders and simulate how they influence others so that their preferred items are the ones chosen by the group.

Using Personality to Create Alliances in Group Recommender Systems*

Lara Quijano-Sánchez, Juan A. Recio Garcia, and Belén Díaz-Agudo

Dep. Ingeniería del Software e Inteligencia Artificial,
Universidad Complutense de Madrid, Spain
jareciog@fdi.ucm.es, belend@sip.ucm.es

Abstract. Our recent work analyses the accuracy of group recommenders when using information about the personality and the social connections between the members of the group. The goal in this paper is the use of personality and trust as the mean to define alliances to reach agreements inside a group of people. The approach reproduces the behaviour of real users when negotiating a common item to consume using three variables: personality, trust and personal preferences. We run an experiment in the movie recommendation domain where we use a personality test to identify the group leaders and test the number of people they are able to convince about a certain item to consume.

1 Introduction

Recommender systems have been one of the main application areas of the techniques commonly used in the Case-Based Reasoning field [1,2]. The analogies between Case Based Reasoning (CBR) and recommenders are obvious. Recommender systems manage items instead of cases but the retrieval methods are very similar. Once the best item is obtained it is proposed directly to the user without requiring adaptation. Moreover, both techniques pay an important attention to the learning processes that improve the performance of the systems by taking into account the preferences or experiences of the users. In a general way we could apply two different approaches. Collaborative recommenders use the ratings already assigned by the users to several products. Users are selected according to their similarity with the target individual (by comparing the ratings given to the products). Most similar users are used as predictors and their ratings are combined to estimate the rating that the target user would assign to a new product. On the other side, the content-based approach compares each item to be proposed with the items already rated by the target user. Then the ratings of the most similar rated items are combined to provide an estimation.

Our recent work [3,4,5,6] analyses the accuracy of group recommenders when using information about the personality and the the social connections between the members of the group. Typically a group recommender uses several subsets

* Supported by the Spanish Ministry of Sci. and Ed. (TIN2009-13692-C03-03).

A. Ram and N. Wiratunga (Eds.): ICCBR 2011, LNAI 6880, pp. 226–240, 2011.

of preferences -one per person- that are combined to create a global recommendation suitable for everyone in the group. Simpler existing works on group recommender systems are based on the aggregation of the preferences of every member of the group, where each member is considered with the same degree of importance [7,8]. However, groups of people can have very different characteristics like size and can be made of people with similar or antagonistic personal preferences. It is a fact that when we face a situation in which the concerns of people appear to be incompatible, a conflict situation arises.

Our previous approaches determine that the general satisfaction of the group is not always the aggregation of the satisfaction of its members as different people have different expectations and behaviour in conflict situations. The *personality* factor reflects the cooperativeness or selfishness of each user when selecting a product for the whole group. This fact is taken into account in recent works that agree on the need of adapting the recommendation process to the group composition. Furthermore, it is also known that the user preferences can be affected by other people of the group and can change over the time [9,2,10]. Personality allows us measuring the degree of acceptance of the products proposed by other users and the way of solving conflicts. Our research characterizes people using the Thomas-Kilmann Conflict Mode Instrument (TKI) [11] that describes a person behavior in conflict situations.

The concept of *trust* [12], can be defined as the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible. Trust networks consist of transitive trust relationships between people, organizations and software agents connected through a medium for communication and interaction. Note that trust is also related to tie strength and previous works have reported that both are conceptually different but there is a correlation between them [13].

In this paper we describe a new approach to solve conflict situations by modeling users interaction in group recommender systems. Instead of computing a global recommendation for the group of people based on the individual preferences and personality of its members, we propose a model where each user negotiates to convince other members about a common item to consume. We exploit the principle of *homophily*, people that share interest with their friends and tend to be friends with people who share their interests. This feature has been shown to exist in many social networks [14,15]. In our model, users with strong personalities try to create *alliances* with other users to support their personal preferences. This way, influencer users obtain the required votes to get their proposal chosen by the group. These influencers, or leaders, try to influence other users and they use their leadership to create the alliance.

Influencers, are typically characterized as thought leaders, or just plain interesting personalities who have the ability to influence potential users. In practice, these individuals may be identified as highly connected individuals or individuals that bridge (also called *connectors* [16]) two relatively large sub-communities. This social behaviour has been extensively researched in the social sciences over the past few decades [17],[14],[18].

Our new approach uses personality and trust as the mean to define *alliances* inside a group of people. An alliance is defined as a subgroup that agree about the same recommendation result. A leader creates alliances with other users (s)he trusts in order to support a concrete product p . The product in the alliance with the bigger number of members is chosen as the global recommendation result. A total agreement situation leads to an alliance including all the people of the group.

Summing up, in this paper we propose a model based on alliances to provide recommendations to groups. We identify leaders by a personality test. Potential allies are obtained by computing the trust between users. Leaders negotiate with their closer friends to conform an alliance that has the majority of votes required to get the influencer's favourite items.

The paper runs as follows. Section2 introduces related work. In Section 3 and 4 we explain an overview of our previous research, a generic architecture for group recommendations, ARISE, that uses personality and trust values in order to improve group recommendations. Section 5 describes the method based on alliances that we propose in this paper. Section 6 describes a case study in the movie recommendation domain and presents some results on the use of alliances in the group decision making. Section 7 concludes the paper.

2 Related Work

Related works about creating alliances and the role of influencers are shown in some online social communities. A *coalition* from social agents area is defined as a temporary association between agents in order to carry out joint projects. The aim is to achieve complex projects by using a better distribution of competencies. An example is the approach of [19] to solve a cooperative game. Different works study automatic methods for coalition formation [20] or properties like efficiency, optimality or stability of the coalition structure [21,22]. Our approach is also related with voting games [23], a popular model of collaboration in multiagent systems. In such games, each agent has a weight (intuitively corresponding to resources he can contribute), and a coalition of agents wins if its total weight meets or exceeds a given threshold.

Our theory is based on the idea of a distributed group recommender system based on previous research on distributed Case Based Reasoning. Distributed CBR assumes multi-case base architectures involving multiple processing agents differing in their problem solving experiences [24]. In this new scenario each case base contains a list of contents, like products, rated by the user. These ratings represent the users explicit preferences that belong to the user model. These individual ratings are later combined with the ratings from other users to obtain a joint recommendation for the group. CBR literature proposes several ways to combine several experiences to obtain improved solutions in distributed architectures. One important method is the *ensemble effect* explained in [25] which proves that the argumentation of two agents improves the results obtained by one only agent working with the same experiences. This conclusion was the precursor of a research line focused on finding the best argumentation protocols to allow CBR agents to discuss about a common problem. In [25] they came up with the AMAL protocol that enables several CBR agents to argue about a common problem by means arguments and counterarguments. We have adapted the idea of agents giving arguments to validate their proposal, to an approach where the agents are influencers who give arguments to try to convince other users they are close to, to support their proposal.

The motivation and main contribution of this work is to use the ideas of alliances formation and collaboration between agents to improve group recommender systems. However in our model people of the same alliance do not collaborate to solve a complex project but reach an agreement on the item to be consumed by the whole group. So, our model does not represent knowledge about agent competencies or resources to contribute. It represents information about people's preferences, personality and trust that are used to convince the other members in the group.

In our method, leaders, who we call influencers, try to wield influence over friends to achieve their own goals. This must be taken into account when recommending items to groups of friends. The main problem when applying this model is the identification of potential influencers and influenced friends. However social networks provide (partially) these data. We can compute the trust between users to measure the closeness of their relationship and therefore the possibility of influence. However, social connections aren't enough for identifying influencers. To do so, we propose to measure the personality of the users.

3 ARISE: Generic Architecture for Group Recommenders Using Social Elements

Our approach, presented in [4,5,6] determines that the general satisfaction of the group is not always the aggregation of the satisfaction of its members, as groups of people can have very different characteristics. The inclusion of social elements into a group recommendation strategy is what we call ARISE[1] (Figure 1). This architecture allows us to simulate in a more realistic way the decision process followed by groups of people when choosing a joint activity.

The architecture of ARISE [6] is divided in six different modules: personality, trust, memory and satisfaction individual preferences estimation, explicit individual preferences, and product data. The information provided by each module is combined by the ARISE's group recommendation methods described in Section 4. Next, we summarize modules functionality:

- **Personality Module.** When making group decision processes there are situations where the concerns of people appear to be incompatible and *conflict situation* arises. Different people have different expectations and behaviour in conflict situations that should be taken into account. We have studied the different behaviours that people have in conflict situations according to their

[1] ARISE stands for Architecture for Recommenders Including Social Elements.

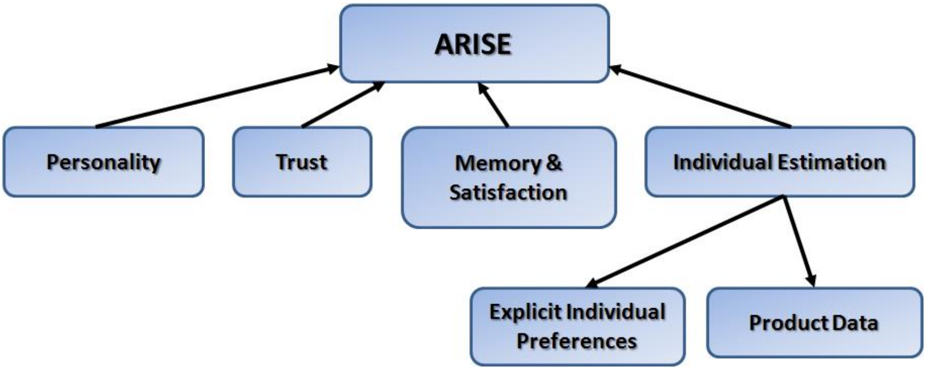


Fig. 1. Facebook application architecture: ARISE

personality. Personality module fulfils the task of obtaining a value that represents the personality of each user. This value, p, is a number $\epsilon[0,1]$ where 1 represents a very strong personality and 0 a very easy going personality. In the ARISE architecture it is described as a high-level module that can be implemented in different ways. We obtain this factor using a popular personality test called TKI [11]. We have chosen this test because it takes very little time to answer it and the questions about the users personality are asked in an indirect way, not digging into too personal questions. In that way users do not resent from a excessively tedious test to answer.

- **Trust Module.** Current research has pointed out that people tend to rely more on recommendations from people they trust (friends) than on recommendations based on anonymous ratings [26]. In this module we evaluate information stored in our users profiles inside a social network, Facebook. With this information we compute the trust between users. Examples of these *social factors* are distance in the social network, number of common friends, intensity, intimacy or duration of the relationship.

  The details of the trust and the personality computation are fully detailed in [4,5].
- **Memory and Satisfaction Module.** After applying the personality and trust factors we must assure a certain degree of satisfaction between all the members of the group. We propose the use of a memory of past recommendations. Having recommendations with memory means that we are able to create a system that remembers all the previous recommendations for a given group. We believe that this is a necessary step when providing a whole set of fair recommendations.
- **Individual Preferences Estimation.** Our recommendation strategies predict the rating that each user would assign to every item in the catalogue and then these estimated ratings are combined to obtain a global prediction for the group. Finally, the product with the highest prediction is proposed. Therefore, a basic building block of the architecture is the module in charge

of the computation of the individual predictions. For the construction of the individual recommender we use the jCOLIBRI framework [27]. jCOLIBRI is currently a reference platform in the Case-Based Reasoning (CBR) community that facilities the design of different types of CBR applications and it has a specific extension for developing recommender systems.

Independently of the approach chosen to implement this generic module of the ARISE's architecture, there are two components (or submodules) that are always required by the individual recommender: A) the explicit individual preferences, which spans any kind of information about the user that is required to predict the rating for a new item. Commonly, it just consists on the ratings given to some products in the catalogue. B) the product data set, which provides the information about the items in the catalogue that should be recommended to the group.

4 Group Recommendation Methods in ARISE

Our group recommendation method is based on the typical preference aggregation approaches. These approaches [7,8] aggregate the users individual predicted ratings $pred(u,i)$ to obtain an estimation for the group $\{gpred(G,i)|u \in G\}$. Then the item with the highest group predicted scoring is proposed, this group recommendation method is what we call a base group recommender.

$$gpred(G,i) = \bigsqcup_{\forall u \in G} pred(u,i) \tag{1}$$

Here G is a group of users, which user u belongs to. This function provides an aggregated value that predicts the group preference for a given item i. By using this estimation, *our group recommender proposes the set of k items with the highest group predicted scoring.*

In our proposal, we modify the individual ratings with the personality and trust factors. This way, we modify the impact of the individual preferences as shown in Equation 2.

$$\begin{aligned} gpred(G,i) &= \bigsqcup_{\forall u \in G} pred'(u,i) \\ pred'(u,i) &= \bigsqcup_{\forall v \in G} f(\, pred(u,i)\, ,\, p_u\, ,\, t_{u,v}\,) \end{aligned} \tag{2}$$

where $gpred(G,i)$ is the group rating prediction for a given item i, $pred(u,i)$ is the original individual prediction for user u and item i, p_u is the personality value for user u and $t_{u,v}$ is the trust value between users u and v.

There are several ways to modify the predicted rating for a user according to the personality and trust factors. These strategies will be depicted in Section 4.2. Next, we will explain the aggregation functions that can be applied to combine the individual estimations.

4.1 Aggregation Functions

A wide set of aggregation functions has been devised for combining individual preferences [9], being the average and least misery strategies the most commonly used. In the experiments presented in this paper we use the average satisfaction strategy, ir refers to the common arithmetic mean, which is a method to derive the central tendency of a sample space. It computes the average of the predicted ratings of each member of the group. The function representing this strategy is:

$$gpred(G,i) = \frac{1}{|G|} \sum_{u \in G} pred'(u,i) \tag{3}$$

Where $pred'(u,i)$ is the predicted rating for each user u, and every item i. $gpred'(G,i)$ is the final rating of item i for the group.

4.2 Modifying Individual Predictions with Social Elements

Our recommendation approaches [5] consist on evaluating the different behaviours that people have when reaching a decision making process. To do so we modify the predictions made by the individual recommender with the personality and trust factors. In that way not all the predictions are taken into account equally. We use two different methods to compute the new individual rating ($pred'(i,u)$) used in Equation 2.

- **Delegation-based method:** The idea behind this method is that users create their opinions based on the opinions of their friends. The estimation of the delegation-based rating ($dbr(u,i)$) given an user u and an item i is computed in this way:

$$pred'(u,i) = dbr(u,i) = \frac{1}{|\sum_{v \in G} t_{u,v}|} \sum_{v \in G \wedge v \neq u} t_{u,v} \cdot (\, pred(v,i) + p_v \,) \tag{4}$$

  In this formula, we take into account the recommendation $pred_{v,i}$ of every friend v for item i. This rating is increased or decreased depending on her personality (p_v), and finally it is weighted according to the level of trust ($t_{u,v}$). Note that this formula is not normalized by the group size and uses the accumulated personality. Therefore, this formula could return a value out of the ratings range. This is simply managed by the recommender by choosing the closest value within the valid range.
- **Influence-based method:** This method simulates the influence that each friend has in a given person. Instead of creating a new preference, it supposes that the user may modify her preference for an item depending on the preferences given by her friends to the same item, as shown in the following formula:

$$pred'(u,i) = ibr(u,i) = pred(u,i) + (1 - p_u) \frac{\sum_{v \in G \wedge v \neq u} t_{u,v} \cdot (\, pred(v,i) - pred(u,i)\,)}{|G| - 1} \quad (5)$$

In this formula, the individual rating for the item ($pred_{u,i}$) is modified according to its difference with the ratings of other users ($pred_{v,i} - pred_{u,i}$). This difference takes into account the trust between users ($t_{u,v}$). Finally, the accumulated difference is weighted according to our personality in an inverse way ($1 - p_u$).

Next section presents the main contribution of this paper, a new group recommendation strategy, that uses the information retrieved by the ARISE architecture, personality, trust and personal preferences in order to provide a group recommendation based on alliances. It consists on a new approach to modify individual predictions with social elements, different from the *delegation-based* and *influence-based* methods that we have just explained.

5 Alliance Based Approach

Alliance based approach first computes personality and trust for every user in the group as explained in section 3. Next step uses this information to identify the leader users and her close friends set. Every user with a personality higher than a threshold α is considered a group leader. In Section 6 we use α as the 85% of the highest personality value in the group. Note that the number of group leaders is not fixed. We have empirically discovered in our case of study that our method performs better when we obtain a number of leaders close to half of the size of the group. For every leader in the group l, we obtain her close friends set $cfs(l)$. This set is obtained using the trust values computed between the leader and every other user in the group and then selecting the users that the leader trusts higher. This set represents all the "possible alliance mates". If the trust between a user, u_i and the leader l is higher than another threshold, β, she is included in her $cfs(l)$.

Negotiation between l and $cfs(l)$ begins to agree on a common product that the leader l likes. This negotiation process allows us to determine whether the proposal made by the leader is accepted or not. It runs as follows:

1. For every user in the group we obtain the individual estimation of ratings of the products in the catalogue. We use the *Individual preferences estimation* module of the ARISE architecture (see Section 3) by applying an individual recommendation approach with the information retrieved in the *explicit individual preferences* module. The construction of the recommender runs as follows.
2. Analyze the recommendations made to the leaders and identify which are their favourite items. This set of items, $lfi(l)$ (leaders favourite items), are the ones that each leader proposes to her close friends set $cfs(l)$ in order to

form the alliance. Note that the size of $lfi(l)$ is not fixed, it can be adjusted depending on the size of the catalogue of items. There are n ($n = |l|$) sets of leaders favourite items ($lfi(l)$), one for each leader in the group.
3. Propose the leaders individual favorite items $lfi(l)$ to leader l "possible alliance mates". A proposal is accepted if the estimated rating that a user u_i, with $u_i \epsilon cfs(l)$, has of the proposed item p_i, with $p_i \epsilon lfi(l)$, is higher than a certain threshold δ. This threshold δ is modified depending on the users personality (it will be bigger with stronger personalities) and also depending on the trust with the leader (if the user has a strong trust on the leader the threshold will be lower). See Equation 6 in Section 6.
4. When an user accepts the proposal we include her in the alliance of that leader. We note that the leader has θ ($\theta = |lfi(l)|$) attempts to "persuade" each one of the users in her $cfs(l)$, one attempt for every item in the set of the leaders favourite items. To be part of the alliance a user just has to accept one item of the proposed list. As we have said before, a leader l creates alliances $alli(l, p)$ with other users supporting a concrete product p_i. If the size of the alliance $|alli(l, p)|$ is greater than a half of the group, the items in $lfi(l)$ are directly chosen as the items for the group. If there is no majority we will choose the items proposed by the larger coalition.

6 Case Study: Movie Recommendation

In this section we evaluate the alliance based approach for group recommendation using the movie recommendation domain. The goal of the experiment is improving other group recommender approaches. We compare the results obtained using alliances with a base group recommender system using the average satisfaction aggregation function and also with our previous approaches using personality and trust [4,5]. The construction of the alliances recommender involves the processing of several factors that are obtained in different ways. The personality values are obtained through the TKI tests [11], whereas trust values are directly extracted from a social network where all the users belong to. Next we explain how we extract the information required from our users, how we measure the results, the configuration of our alliances recommender and the results of the experiment.

6.1 Experimental Setup

In order to perform our experiment in the movie recommendation domain, we created two events in two different social networks, Facebook[2] and Tuenti[3]. In these events we asked some of our users to complete three questionnaires[4]. The first questionnaire serves to obtain the personality of each user, is the one run by the *personality module*. Second questionnaire gets the individual preferences

[2] http://www.facebook.com
[3] http://www.tuenti.com. The most popular social network in Spain.
[4] Questionnaires are accessible at http://www.lara.warhalla.com/ (in Spanish.)

of the user about cinema. Users have to evaluate 50 heterogeneous movies from the MovieLens data set [28](rating them with a range of 0.0 to 5.0). These 50 movies are the list of products that are assigned to each agent, and they are stored in the *Explicit individual preferences module*.

Finally, third test asks users to choose their 3 favourite movies from a list of 15 *recent* movies (of the 2009 year), that represents a movie listing from a cinema. This list of 15 products is the one gathered by the *Product Data module*. The movie listing was chosen from movies of the MovieLens database using a diversity function. The 3 movies selected by each user are included as her individual favourites, *if*. These movies are the ones she would actually like to watch or had enjoyed best. The answers to these questionnaires are analysed to define the user profile of each participant. 58 real users have participated in our experiment.

To measure the accuracy of the group recommendation we brought our users together in person and ask them to mix differently several times and simulate that they are going to the cinema together, forming different groups that would actually come out in reality. We provide them the 15 movies that represent our movie listing and we ask them to choose in the group which 3 movies in order they actually would watch together. We manage to gather 10 groups: 6 groups of 5 members and 4 groups of 9 members. The three movies that each group chooses are stored as the *real group favourites* set –*rgf*–. This way, to evaluate the accuracy of our recommender we can compare the set proposed by the recommender –the *pgf* set– with the real preferences *rgf*. The evaluation metrics applied to compare both sets are explained in Section 6.2.

Our group recommendation strategies combine individual recommendations to find an item (movie) suitable for any user in the group. This individual recommender is built using the jCOLIBRI framework [29] and follows a content based approach [30] to find the most similar movie rated by the user. It uses product descriptions and returns the collection of products that are more similar to the aimed product, assigning the rating given by the user as a prediction. This set of movies is different for each user and it has the information retrieved from the second questionnaire.

6.2 Evaluation Metrics

Our experiment requires an evaluation function to measure the accuracy of the group recommendation. To do so, we compare the results of our recommender system to the real preferences of the users (that is, what would happen in a real life situation). When we started our evaluation process we took into account the number of estimated movies that we were going to take into account. We are not interested on a long list of ordered items that estimates movies a user or group should watch. Real users are only interested on a few movies they really want to watch. This fact discards several evaluation metrics that compare the ordering of the items in the real list of favourite movies and the estimated one (MAE, nDCGs, etc.). On the other hand, the number of relevant and retrieved items in our system is fixed. Therefore, we cannot use general measures like recall or precision. However, there are some metrics used in the Information Extraction field [31] that limit the retrieved set. This is the case of the *precision@n* measure that computes the *precision* after n items have been retrieved. In our case, we can use the *precision@3* to evaluate how many of the movies in pgf are in the rgf set (note that $|rgf| = 3$). This kind of evaluation can be seen from a different point of view: we are usually interested on having at least one of the movies from *pgf* in the *rgf* set. This measure is called *success@n* and returns 1 if there is at least one hit in the first n positions. Therefore, we could use *success*@3 to evaluate our system computing the rate of recommendations where we have at least one-hit in the real group favourites list. For example, a 90% of accuracy using *success*@3 represents that the recommender suggests at least one correct movie for the 90% of the evaluated groups. In fact, *success*@3 is equivalent to having *precision*@3 $> 1/3$. We can also define a 2*success*@3 metric (equivalent to *precision*@3 $> 2/3$) that represents how many times the estimated favourites list *pgf* contains at least two movies from *rgf*. Obviously, it is much more difficult to achieve high results using this second measure.

6.3 Alliance Recommender System

For each group we build the alliance recommender using the following steps:

1. We obtain the members of the group and we calculate an estimation of their individual preferences with content based individual recommender system. After this process what we have is an estimated rating of each user for each of the 15 movies in the movie listing from the cinema.
2. We identify the leaders of the group, which are those who have a personality that is higher than the 85% of the personality value of the user with the strongest personality in the group (threshold α).
3. For each of the leaders we try to find alliances. To find the possible candidates that could form the alliance we select those users who have a trust with the leader higher than the 75% of the trust value of the most trusted user of the leader (threshold β).
4. To accept a user as part of the coalition, we propose the 3 movies that the leader of the group has with the higher rating, that as we remember we obtained from the individual recommender. If the users predicted rating for that movies is higher than threshold δ then the user is accepted as part of the alliance. Threshold δ is obtained with the following formula:

$$\delta = ir_{u,5} - t_i + p_r \tag{6}$$

where $ir_{u,5}$ is the predicted rating of the best fifth item for the user, $t_i = \mu * trust_{u,leader}$, and $p_r = \lambda * p_u$. $trust_{u,leader}$ represents the existing trust between the user and the leader, p_u is the personality value of the user and μ and λ have been experimentally obtained. ($\mu > 0.4$ and $\lambda < 0.5$).

We have built another alliance recommender system simplifying this last formula, we call it *Alliance-based Recommender simpler version*, we have done this to study the influence and benefits of using the trust and personality factors in order to vary the threshold of acceptance of a proposed item δ. This variation of our method obtains the threshold δ with this simplified formula:

$$\delta = ir_{u,5} \tag{7}$$

where $ir_{u,5}$ is the predicted rating of the best fifth item for the user.

5. After forming all the alliances we compare the sizes of the alliances. If the size of the alliance is greater than a half of the group we propose as selected items, the favourites of the leader, which are the 3 movies that the leader of the group has with the higher rating.

6.4 Experimental Results

In Figure 2 we have analyzed the performance of the base recommender, a group recommender using the same data-set but applying our influence-based recommendation method, a group recommender using the same data-set applying our delegation-based recommendation method, a group recommender with the simplified version of our alliances approach (the one that does not use personality and trust factors in order to calculate the threshold of acceptance of each item) and finally a recommender with our alliances approach. We can see that we have improved the performance of the basic recommender in a 10% with the success@3 and in a 40% with the 2success@3. Results also show that with the 2success@3 measure the alliances approach obtains the best results. As we have explained before this measure is much more difficult to obtain than the success@3 measure, so with this results we validate our alliances method and conclude that

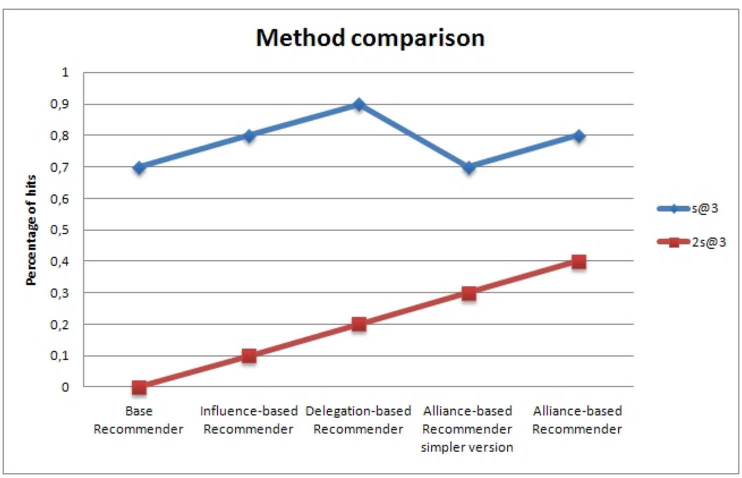


Fig. 2. Comparison of the results obtained with the base recommender, the influence-based recommender, the delegation-based recommender, the alliance-based recommender simpler version and alliance-based recommender

with it we improve our previous group recommendation strategies. From this Figure we also observe that it is essential to include the personality and trust factors in order to calculate the threshold of acceptance of each item, because with the simplified version of our alliance approach the results with the success@3 measure are equal to the base recommender so we do not improve with it the group recommendation. We must note that we still can validate our strategy because for the 2success@3 even with the simplified version of our alliance approach results are better than the ones obtained by the base, influence-based, and delegation-based recommenders.

7 Conclusions

In this paper we have proposed and evaluated a group recommendation strategy based on alliances for the recommendation of products in social networks. In previous papers we have already experimented with a novel method of making recommendations for groups taking into account the group personality composition and the social structure of the group. Once shown that personality profiles can improve a recommendation for a group of people, we have extended this approach by reflecting in a more realistic way the social relationships between the users involved in the recommendation. We have tested our method in the movie recommendation domain and shown that group recommendation using alliances improves the base group recommender system using the average satisfaction aggregation function. Results also have shown that with the 2success@3 measure the alliances approach obtains the best results and improve our previous group recommendation strategies. We have also observed that it is essential to include the personality and trust factors in order to calculate the threshold of acceptance of each item in the recommender system. Our proposed alliance based approach for group recommendation is based on identifying users with strong personalities try to create *alliances* with other users to support their personal preferences. This way, influencer users obtain the required votes to get their proposal chosen by the group. These influencers, or leaders, try to influence other users and they use their leadership to create the alliance. The proposed method first computes personality and trust for very user in the group and then uses this information to identify the leader users and her close friends set. Negotiation between the leader and people from her close friends set begins to agree on a common product that the leader likes. This negotiation process allows us to determine whether the proposal made by the leader is accepted or not. Our ongoing work consists on making further evaluations of our alliances method by embedding it into a social network application, where we will be able to continue our experiments with larger and more general populations.

References

1. Bridge, D., Göker, M.H., McGinty, L., Smyth, B.: Case-based recommender systems. Knowledge Engineering Review 20, 315–320 (2006)

2. Jameson, A., Smyth, B.: Recommendation to groups. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 596–627. Springer, Heidelberg (2007)
3. Recio-García, J.A., Jimenez-Diaz, G., Sánchez-Ruiz, A.A., Díaz-Agudo, B.: Personality aware recommendations to groups. In: Procs. of the 2009 ACM Conference on Recommender Systems, pp. 325–328. ACM, New York (2009)
4. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B.: Personality and social trust in group recommendations. In: Procs. of the 22th Int. Conference on Tools with Artificial Intelligence, ICTAI 2010, pp. 121–126. IEEE Computer Society, Los Alamitos (2010)
5. Quijano-Sánchez, L., Recio-García, J., Díaz-Agudo, B., Jiménez-Díaz, G.: Social factors in group recommender systems. In: ACM-TIST, TIST-2011-01-0013 (in press, 2011)
6. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B., Jiménez-Díaz, G.: Happy movie: A group recommender application in facebook. In: 24th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2011 (2011)
7. Masthoff, J., Gatt, A.: In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. User Modeling and User-Adapted Interaction 16, 281–319 (2006)
8. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: a recommender system for groups of users. In: ECSCW 2001: Proceedings of the Seventh Conference on European Conference on Computer Supported Cooperative Work, pp. 199–218. Kluwer Academic Publishers, Norwell (2001)
9. Masthoff, J.: Group modeling: Selecting a sequence of television items to suit a group of viewers. User Modeling and User-Adapted Interaction 14, 37–85 (2004)
10. Chen, Y.L., Cheng, L.C., Chuang, C.N.: A group recommendation system with consideration of interactions among group members. Expert Syst. Appl. 34, 2082–2090 (2008)
11. Thomas, K., Kilmann, R.: Thomas-Kilmann Conflict Mode Instrument, Tuxedo, N.Y (1974)
12. Josang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision, pp. 618–644 (2007)
13. Levin, D.Z., Cross, R., Abrams, L.C.: The strength of weak ties you can trust: the mediating role of trust in effective knowledge transfer. Management Science 50, 1477–1490 (2004)
14. Miller Mcpherson, L.S.L., Cook, J.M.: Birds of a feather: Homophily in social networks, pp. 415–444 (2001)
15. Lazarsfeld, P., Merton, R.: Friendship as a social process: A substantive and methodological analysis. In: Berger, M., Abel, T., Page, C. (eds.) Freedom and Control in Modern Society, pp. 18–66. Van Nostrand, New York (1954)
16. Gladwell, M.: The tipping point: How little things canmake a big difference. Little Brown, Boston (2000)
17. Burt, R.S.: Toward a structural theory of action: Network models of social structure, perception and action, pp. 1336–1338 (1982)
18. McPherson, M., Smith-Lovin, L.: Homophily in voluntary organizations: Status distance and the composition of face-to-face groups, pp. 370–379 (1987)
19. Chalkiadakis, G., Elkind, E., Markakis, E., Polukarov, M., Jennings, N.R.: Cooperative games with overlapping coalitions. J. Artif. Intell. Res. (JAIR) 39, 179–216 (2010)

20. Barton, L., Allan, V.H.: Methods for coalition formation in adaptation-based social networks. In: Klusch, M., Hindriks, K.V., Papazoglou, M.P., Sterling, L. (eds.) CIA 2007. LNCS (LNAI), vol. 4676, pp. 285–297. Springer, Heidelberg (2007)
21. Airiau, S., Sen, S.: On the stability of an optimal coalition structure. In: Proceedings of 19th European Conference on Artificial Intelligence, ECAI 2010, Lisbon, Portugal, August 16-20, pp. 203–208 (2010)
22. Nuno, D., Ao, S.J.S., Helder, C.: Agent-based social simulation with coalitions in social reasoning. In: Moss, S., Davidsson, P. (eds.) MABS 2000. LNCS (LNAI), vol. 1979, pp. 244–265. Springer, Heidelberg (2001)
23. Elkind, E., Chalkiadakis, G., Jennings, N.R.: Coalition structures in weighted voting games. In: 18th European Conference on Artificial Intelligence, ECAI 2008. Frontiers in Artificial Intelligence and Applications, vol. 178, pp. 393–397. IOS Press, Amsterdam (2008)
24. McGinty, L., Smyth, B.: Collaborative case-based reasoning: Applications in personalised route planning. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 362–376. Springer, Heidelberg (2001)
25. Ontañón, S., Plaza, E.: An argumentation-based framework for deliberation in multi-agent systems. In: Rahwan, I., Parsons, S., Reed, C. (eds.) Argumentation in Multi-Agent Systems. LNCS (LNAI), vol. 4946, pp. 178–196. Springer, Heidelberg (2008)
26. Sinha, R.R., Swearingen, K.: Comparing recommendations made by online systems and friends. In: DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries (2001)
27. Recio-García, J.A., Díaz-Agudo, B., González-Calero, P.A.: Prototyping Recommender Systems in jCOLIBRI. In: Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 243–250. ACM, NY (2008)
28. Bobadilla, J., Serradilla, F., Hernando, A.: Collaborative filtering adapted to recommender systems of e-learning. Knowl.-Based Syst. 22, 261–265 (2009)
29. Díaz-Agudo, B., González-Calero, P.A., Recio-García, J.A., Sánchez-Ruiz-Granados, A.A.: Building cbr systems with jcolibri. Sci. Comput. Program. 69, 68–75 (2007)
30. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)
31. Oard, D.W., Baron, J.R., Hedin, B., Lewis, D.D., Tomlinson, S.: Evaluation of information retrieval for e-discovery. Artif. Intell. Law 18, 347–386 (2010)

Chapter 13

User satisfaction in long term group recommendations

13.1 Citation

Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B. User satisfaction in long term group recommendations in: Procs of the 19th International Conference on Case Based Reasoning, ICCBR'11. Springer-Verlag Berlin Heidelberg, Greenwich, London, pages: 211-225, 2011. ISBN 978-3-642-23290-9. ICCBR is ranked as CORE C in the CORE ranking.

13.2 Contributions covered by this paper

In this paper we have created a new social group recommendation technique that uses the memory of past recommendations to ensure a homogeneous level of satisfaction between group members. We have proven that this method provides the highest and less uneven level of satisfaction with the group recommendations in the long run.

User Satisfaction in Long Term Group Recommendations*

Lara Quijano-Sánchez, Juan A. Recio-García, and Belén Díaz-Agudo

Dep. Ingeniería del Software e Inteligencia Artificial,
Universidad Complutense de Madrid, Spain
{lara.quijano,jareciog}@fdi.ucm.es, belend@sip.ucm.es

Abstract. In this paper we introduce our application *HappyMovie*, a Facebook application for movie recommendation to groups. This system takes advantage of social data available in this social network to promote fairness for the provided recommendations. Group recommendations are based in the individual satisfaction of each individual. The (in)satisfaction of users modifies the typical aggregation functions used to estimate the value of an item for the group. This paper proposes a memory of past recommendations to compute the satisfaction of users when similar items (movies, in this case) are recommended several times.

1 Introduction

Recommender systems were born from the necessity of having some kind of guidance when searching through complex product spaces. More precisely, group recommenders were built to help groups of people, who share a common activity, decide in conflict situations. Our previous works [1,2,3,4] presented our approach of making recommendations for groups of people connected through social network structures. In them we introduced a method, based on three important features: personality, social trust and memory of past recommendations to ensure social fairness. We have proven that our theories for making recommendations to groups of people connected through social network structures improve the current existing methods.

Our current work consists on providing an Facebook application reachable to a great deal of people, where we can continue our research and experiments and also extend the group recommendations to different domains. Besides, by having the application located in a social network, we can extract the information regarding the users from it. In that way we don't have to bombard our users with questionnaires in order to build the personal profile that we need to do the recommendation, because we can extract from their Facebook profiles a great deal of the information we need. We have moved our standalone group recommendation application to a public application where everybody can benefit of it. Our method includes the analysis of the group personality composition and the trust

* Supported by the Spanish Ministry of Science and Education (TIN2009-13692-C03-03).

A. Ram and N. Wiratunga (Eds.): ICCBR 2011, LNAI 6880, pp. 211–225, 2011.

between users to improve the accuracy of group recommenders. This way we simulate in a more realistic way the argumentation process followed by groups of people when agreeing on a common activity. Our recommendation method and the architecture of our Facebook application are valid for any domains with rated products. However, we have applied them to an specific domain, the movie domain, and we have created *Happy Movie*.

The main goal of this paper is to analyse the impact of the last social factor involved in our group recommendation method: group satisfaction to ensure social fairness. The use of group satisfaction is based on some results from organizational behaviour and social psychology that have highlighted the concept of *emotional contagion* [5]. This social aspect states that the satisfaction of an individual is likely to depend on other individuals of the group [6,7]. In this context, social fairness is understood as the intention of maximizing the personal satisfaction of every user in the group and minimizing their differences. To achieve this goal, we propose a memory of past recommendations that is used to compute an individual satisfaction value that is later combined to estimate the global satisfaction of the group for the provided recommendations. This group satisfaction is critical in recommender systems that propose the same kind of items several times for the same group of people. Some examples are movies, music, leisure trips and any other social activity.

Several works have focused in avoiding repeated recommendations and recommendations that tend to be detrimental for the same group members repeatedly. MusicFX [8] employs a weighted random policy for selecting one of the top radio stations selected by the recommender, instead of always selecting the top category. Another solution is taking into account the history of the results produced by the recommender. For example, in FlyTrap [9] the previous selections are taken in consideration. This way, when they choose the next song to be played, abrupt changes of genre do not appear. Another system that takes into account the previous selections is PoolCasting [10]. It uses a Case-Based Reasoning system to generate a sequence of songs customized for a community of listeners. To select each song in the sequence, first a subset of songs musically associated with the last song of the sequence is retrieved from a music pool; then the preferences of the audience expressed as cases are reused to customize the selection for the group of listeners; finally, listeners can revise their satisfaction (or lack of) for the songs they have heard.

The process that we have followed when making group recommendations which ensure social fairness is very similar to the Case Base Reasoning (CBR) cycle [11]. CBR is a successful and established methodology in Artificial Intelligence that has inspired us in the implementation of our recommender with memory. In our system each recommendation provided by the group recommender will be stored as a new case that can be used later to improve the next recommendation. This fact corresponds with the *retain phase* in the CBR cycle. This way, we acquire the experiences that will be useful for the resolution of future recommendations because we will know which products have been recommended to a group. We also store how much satisfied each member of the group is with this recommendation, so we are able to adjust the satisfaction factor in future recommendations. Before making the following recommendation we will check the previous situation, which in the CBR cycle will be the *retrieve phase*. Once we obtain that information we can perform a new recommendation, but taking into account what we have retained (the products that we have already recommended and how satisfied each of the members of the group are). This will be equivalent to the *reuse phase* in the CBR cycle. Last but not least, we will modify the recommendation so that the proposed products are not repeated and we assure a certain degree of fairness when we benefit the preferences of each users. This last phase, the *revise* one, will close de CBR cycle.

The next section details our recommender application. In Section 3 we explain the recommendation techniques used by the application to select items for the group. Experimental evaluation is exposed in Section 4 and finally conclusions are detailed in Section 5.

2 Facebook Application: Happy Movie

HappyMovie is a Facebook application where we provide a group recommendation for groups of people that wish to go together to the cinema. Although this application has been initially designed for recommending movies, this domain will be extended as our method is valid for any other domain with rated products.

In the following sections we are going to explain the uses of the application and its architecture.

2.1 Uses of Happy Movie

In order to use our application, users only have to start their Facebook account and look for *HappyMovie* in the applications section. We explain the uses of the application through an example of a given group of people connected in the social network. The necessary steps to obtain a movie recommendation for a group with Happy movie are explained below:

- **Creating the user profile in the application:** Before any user has access to the movie recommendation results we have to create their individual "recommendation profile" which is necessary for our recommendation method. This profile is based on three different aspects: personality, individual preferences and trust with other users. To obtain the personality and preferences, users must answer two different tests. The first one is the personality test, where users have to choose a series of characters to whom they feel identified, as shown in Figure 1 (left image). Once they have answered the personality test, users have to rate a set of movies (at least 20 movies), where they suggest their personal preferences , as shown in Figure 1 (right image). Finally, to obtain the last factor –trust– the application reads the information stored in the Facebook personal profile. It calculates the trust that the user has with all the other users that have joined the event up to now.

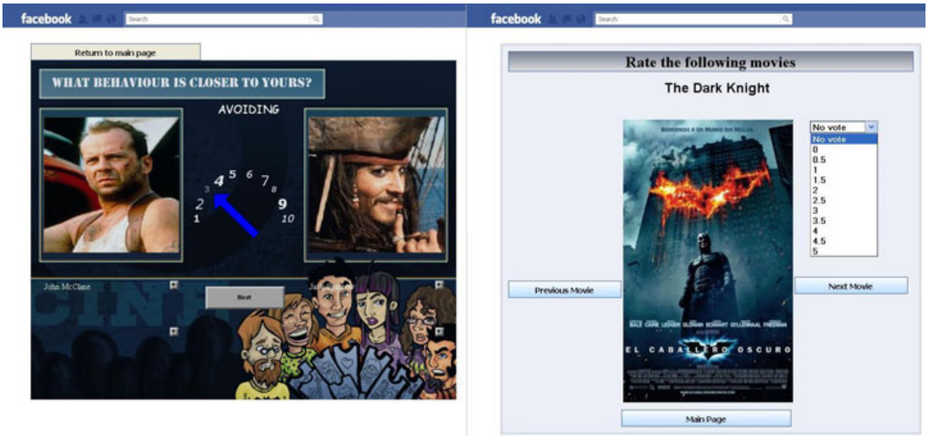


Fig. 1. Personality and Preferences test in *HappyMovie*

- **Creating the activity:** The organizer user, U_1 decides to organize an activity and starts the application to create a new event "*Let's go to the cinema*" as shown in Figure 2 (left picture). To create the event, organizers must establish some data like place, date or invited users. Once the event has been created any user attending the event can see the date and place of the event and a proposal of three movies, that are the best ones that our group recommender has found for the current group of users attending the event.
  For example, when U_1 starts the application for the first time she has the role of *organizer*. As an *organizer* she firstly invites some of her Facebook friends to the event. Lets say that she invites U_2 and U_3. Next, she chooses the place and date where the event will take place.
  Once the *organizer* finalizes this initial configuration she will continue with a role of *common user*. When users participate in the event as *common users* they are also able to invite any Facebook friends they wish and they can retire from the event at any time. For example, U_2 accepts the invitation of the event and later she invites her Facebook friend U_4. On the other hand, U_3 initially accepts the invitation and joins the event but later she decides against going, so she retires from it.
- **Recommendation method:**When the application has obtained the three factors that identify each user that joins the event (personality, individual preferences and trust between other users) it provides a group recommendation using our concrete method which is explained in Section 3.
- **Having the recommendation made:** When the event is created it looks up for the current movie listing from the selected city and provides a list of 3 movies, which represent the best 3 movies that the recommender has found in the movie listing for the users that have joined the event up to now, this is shown in Figure 2 (right picture). This list is automatically updated every time a user joins the event or retires from it. This process keeps going on until the day that the *organizer* has selected as final date. In our example, it

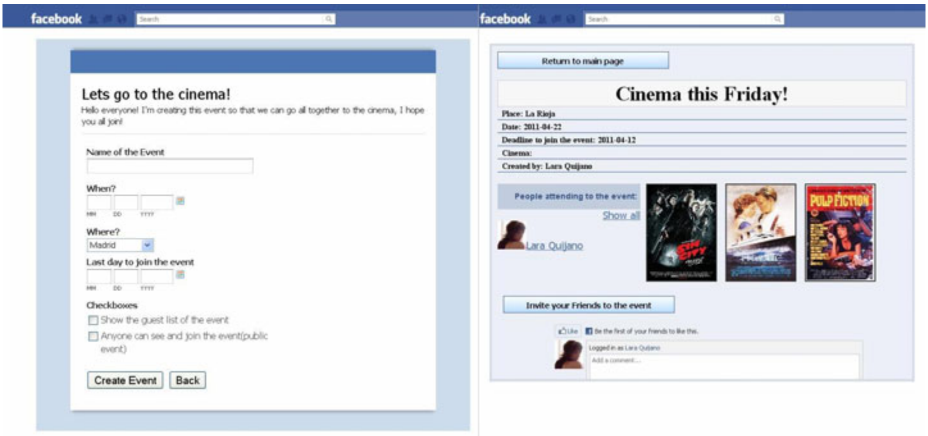


Fig. 2. How to create an activity in *HappyMovie* and how events look like in *HappyMovie*

initially provides a recommendation for users U_1, U_2 and U_3 when they first join the event. Later, when U_3 retires from the event a new recommendation is made for users U_1 and U_2. Finally when U_4, who was invited by U_2, joins the event another new set of 3 movies appears for users U_1, U_2 and U_4. When the expiration date is reached users can see the final movie list. In that moment they are allowed to vote each of the movies individually. This process lets us decide which movie they are finally watching and, more important, it provides the required feedback to evaluate the quality of our recommendation.

2.2 Happy Movie's Modular Architecture

The architecture of *HappyMovie* is represented in Figure 3. We can see that the application is divided in seven different modules: TKI Metaphor, Facebook Profile Analysis, Satisfaction Data Base, Web Test, Web Crawling, Content Based Estimation and *HappyMovie*'s group recommender. Next sections explain what are the basis of each of these modules.

– **TKI Metaphor:** This module fulfils the task of obtaining a value that represents the personality of each user. To do so, each user must answer to a personality test that measures people's behavior in conflict situations. In our previous works [1,2,3] we used the TKI personality test [12], that consists on 30 questions where the user has to decide how she will react in the exposed situations. As we have prove in our previous works this is a tedious test to answer but its results are very reliable. To make the application more easy going we have introduced a movie metaphor as an alternative method to obtain the users personality in conflict situations. This interactive metaphor consists on displaying two movies characters with opposite personalities for five

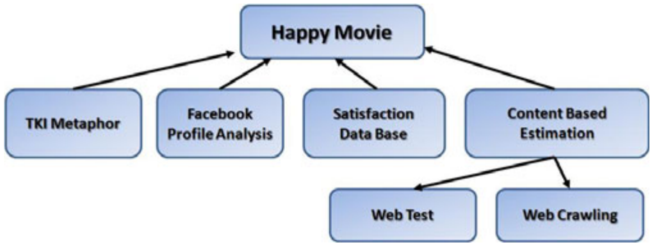


Fig. 3. Facebook application arquitecture

different modes of responding to conflict situations (this personality modes are the ones presented in the TKI experiment). One character represents the essential characteristics of the category, while the other one represents all the opposite ones. What the user has to do is to choose with whom of each pair of characters she feels more identified by simple moving a graded arrow, as seen in Figure 1 (left image).

- **Facebook Profile Analysis:** Once *Happy Movie* is running, the trust module must perform its estimation every time a user joins the event. When this happens the trust module explores the users who are currently in the list of attending users and calculates the trust of each of them with the user who has recently joined the event. To do so, the profiles of the two users will be analysed, comparing different social factors. The Trust Module is the module that has more benefits due to embedding the application in a social network. Previously, with a standalone application, the task of obtaining the data required to compute the trust between users was very tedious. Now, we are able to calculate the trust between users extracting the specific information from each of their own profiles in the social network. Users in Facebook can post on their profiles a huge amount of personal information that can be analysed to compute the trust to other users: distance in the social network, number of shared comments, likes and interests, personal information, pictures, games, duration of the friendship, ... We analyse 10 different trust factors comparing the information stored in their Facebook profiles. Next, these factors are combined using a weighted average. A detailed explanation of the trust factors obtained from Facebook and the combination process is provided in [3].
- **Satisfaction Data Base:** In this module we store all the recommendations that have been made for every user and every group. This specific feature of our application is fully detailed in Section 3.1. This module manages the following information: Each group to whom each user has participated on, each movie that each group has watched and the satisfaction of each user with each of her groups.
- **Web Test:** It consists on a test of the individual preferences of each user. Each time that the user uses the application she can modify her preferences or evaluate more, however it is only compulsory the first time she uses the application. In our specific domain, movies, the user will see the title of the

movie and the movie poster. Our users are provided with 50 movies from which they have to rate (in a rank of 0-5) at least 20, this is shown in Figure 1 (right image). These preferences are stored as the individual case base of each user.

- **Web Crawling:** This module searches the web and finds the movie listing of the city that the *organizer* has selected. Once it has that information it searches the complete file of each of the movies in the movie listing. Later, it analyses the file and extracts all the data required to define the movie. Each specific data is a field that the individual recommender contrasts. For example, in our particular case of study, these fields are main actors, director, year,etc. The recovered items, with all their specific information, are sent to the individual recommender module and to the group recommender module as they are the products to be recommended.
- **Content Based Estimation:** This is the individual recommender module, it is built using the jCOLIBRI framework extension to build recommender systems [13] and follows a content based approach [14] that uses descriptions of the products to be recommended and returns the collection of products that are more similar to the aimed product. In these particular case of study, *HappyMovie*, it returns the best three movies a user should watch individually. As it is a content based recommender system it has stored a case base, and the recommender compares all the considered items to be recommended with this case base. This case base is different for each user and has the information retrieved from the Web Test module.

3 Group Recommendation Methods

We have developed a group recommendation method which is based on the typical preference aggregation approaches. These approaches [5,15] aggregate the users individual predicted ratings $pred(u, i)$ to obtain an estimation for the group $\{gpred(G, i)|u \in G\}$. Then the item with the highest group predicted scoring is proposed.

$$gpred(G, i) = \bigsqcup_{\forall u \in G} pred(u, i) \tag{1}$$

Here G is a group of users, which user u belongs to. This function provides an aggregated value that predicts the group preference for a given item i. By using this estimation, *our group recommender proposes the set of k items with the highest group predicted scoring.*

In our proposal, we modify the individual ratings with the personality, trust and satisfaction factors. This way, we modify the impact of the individual preferences as shown in Equation 2.

$$\begin{aligned} gpred(G, i) &= \bigsqcup_{\forall u \in G} pred'(u, i) \\ pred'(u, i) &= \bigsqcup_{\forall v \in G} f(\, pred(u, i)\, ,\, p_u\, ,\, t_{u,v}\, ,\, s_u) \end{aligned} \tag{2}$$

where $gpred(G, i)$ is the group rating prediction for a given item i, $pred(u, i)$ is the original individual prediction for user u and item i, p_u is the personality value for user u, $t_{u,v}$ is the trust value between users u and v, and s_u is the satisfaction of user u within the group.

A wide set of aggregation functions has been devised for combining individual preferences [16], being the average, least misery and most pleasure strategies the most commonly used:

- **Average Satisfaction:** Refers to the common arithmetic mean, which is a method to derive the central tendency of a sample space. It computes the average of the predicted ratings of each member of the group. The function that represents this strategy is:

$$gpred(G, i) = \frac{1}{|G|} \sum_{u \in G} pred'(u, i) \tag{3}$$

  Where $pred'(u, i)$ is the predicted rating for each user u, and every item i. $gpred(G, i)$ is the final rating of item i for the group.
- **Least Misery:** This strategy considers that a group is as happy as its least happy member. The final list of ratings is the minimum of each of the individual ratings. A disadvantage can be that if the majority really likes one item, but one person does not, then it will never be chosen.

$$gpred(G, i) = \min_{u \in G} pred'(u, i) \tag{4}$$

- **Most Pleasure Strategy:** It is the opposite of the previous strategy, Least Misery, it chooses the highest rating for each item to form the final list of predicted ratings.

$$gpred(G, i) = \max_{u \in G} pred'(u, i) \tag{5}$$

Once we have introduced the typical aggregation approaches we can explain the estimation functions. We use three different methods to compute $pred'_{i,u}$, that as we have explained before, is a modification of the predicted rating for a user according to the personality, trust and satisfaction factors. The main ideas of these approaches are explained below:

- **Personality-based method (*pbm*):** Takes into account the differences in the personalities between pairs of individuals in the group. It is based on the modified average satisfaction employed in our previous work [1]. This strategy reflects that assertive characters will have more influence in the aggregated scoring than the cooperative characters. Our approach uses the type of personality to weight the influence of the estimated ratings during the recommendation process.
- **Delegation-based method (*dbm*):** The idea behind this method is that users create their opinion based on the opinions of their friends. It tries to simulate the following behaviour: when we are deciding which item to choose within a group of users we ask the people who we trust. Moreover, we also

take into account their personality to give a certain importance to their opinions (for example, because we know that a selfish person may get angry if we do not choose her preferred item).

- **Influence-based method (*ibm*):** Simulates the influence that each friend has in a given person. It supposes that the user may modify her preference for an item depending on the preferences of her friends to the same item. For example, if our rating for an item is 3 and our friend has a 5 rating for the same item, we could think on modifying our rating to 4. Depending on the trust with this friend, we decide the level of variation for our rating (i.e. 3.5 if the trust is low, and 4.5 if trust is high). Furthermore, the variation of our rating also depends on our personality. If we have a strong personality we will not be willing to change our rating, but if we have a weak personality we could be easily influenced by other users.

Once the estimation methods are outlined, next section details how to include the memory of users satisfaction in the recommendation process.

3.1 Including User Satisfaction in the Recommendation Process

Our approach for including fairness is based on the satisfaction of the users that conform a group. We propose a modification of the previous methods including a satisfaction parameter that measures the degree of happiness of every user regarding past recommendations. Our goal is to maximize the satisfaction among the group by promoting the items preferred by most dissatisfied users. To achieve this goal we need to keep track of past recommendations and the evolution of the satisfaction of each user. In *HappyMovie* this task is delegated to the Satisfaction Data Base module (see Figure 3).

Having recommendations with memory means that we are able to create a system that remembers all the previous recommendations for a given group. It is a Case-Based reasoning approach to the recommendation process. and a necessary step when providing a whole set of fair recommendations. This way, if one member accepts a proposal that she was not interested in, next time she will have some kind of priority in the recommendation process. This means that her opinion will have a higher weight next time. These weights will also be influenced by the different personalities of each group member. For example, someone with a strong personality that has been negatively affected would be immediately compensated next time; however someone with mild personality would not have problems giving in several times.

The satisfaction value s_u is the level of satisfaction of a user u. A user who is extremely happy with the recommendations will have this satisfaction measure value close to 1. However, the more upset with the recommendations she is, the more that this value will decrease, reaching down to 0 in the worst case. An important and interesting issue of this approach is the time scope of the memory of user's satisfaction. We can update the s_u value to reflect the satisfaction according to the last immediate group recommendation or to take into account previous ones. Therefore, the satisfaction value for an execution t of the recommender may depend on the satisfaction of the user with the items recommended in t but it also depends on her satisfaction in the previous recommendations $t-1, t-2, \ldots$. Therefore we manage two satisfaction values:

– Instant satisfaction (is_u): reflects the immediate satisfaction of the user with the last recommendation. This is, her conformance with the last item recommended to the group. Its value can be obtained by estimating the preference of the user for the item selected to the group among all the items available.
– Global satisfaction: (s_u): measures the average satisfaction of the user among time. It is updated every time a recommendation is made:

$$s_u(t) = (1-\delta) \cdot is_u(t) + \delta \cdot s_u(t-1) \qquad (6)$$

In this equation we use the $\delta \in [0..1]$ parameter to adjust the impact of the previous satisfaction when updating that value. Somehow, this parameter measures the degree of forgetfulness about past (in)satisfacion. For example, some people could easily remember that they were not taken into account for the last recommendation when facing a new decision making process to select a similar item. On the other hand, other users won't ever take it into account. The measurement of this value belongs to the domain of the social sciences and is out of the scope of this paper. We have estimated it experimentally as exposed in next section.

It is important to note that the instant satisfaction value is also weighted depending on the personality of the user to reflect the importance of satisfying that concrete user.

In next Section we explain the details of our experimental evaluation to measure the impact of memory in the recommendation process of *HappyMovie*.

4 Case Study: Experimenting with Memory

Our goal is to estimate what is the best recommendation strategy for long term recommendations in *HappyMovie*. This strategy will be updated once we have real user data available. However, initially we must estimate this strategy to provide recommendations to our users. Therefore we have designed an experiment with synthetically generated data about users and movies. We must note that the validity of experimenting with this synthetically generated users has been already proven in our previous studies [3]. We simulate user preferences and personality to simulate different scenarios where several groups of users choose a movie for going to the cinema. In our previous work [4] we have evaluated the estimation strategies –personality (*pbm*), delegation (*dpm*) and influence(*ibm*)– without taking into account the memory of past recommendations. These experiments were performed with data from real volunteers that simulated going to the cinema together. Results shown that *dbm* and *ibm* provide better results than *pbm* when including the trust factor (t_u). However for our simulation it is impossible to generate synthetically that value and therefore, we have focused on the *pbm* to evaluate which is the best aggregation function: average satisfaction, least misery or most pleasure.

4.1 Experimental Set-Up

The experiment configuration runs as follows. We have simulated 1000 groups of individuals going to the movies together 15 times. Each group consists of 10 individuals. Although the composition of the group does not change, having such a large number of groups let's us include in the simulation any kind of variation in their composition. Movies are described by means of a vector that represents the degree of conformance with several genres (terror, action, romance). These genres were obtained from the MovieLens database [17]. Correspondingly, cinema preferences of each individual are represented in the same way. Movies and individual preferences are generated randomly and compared using the cosine distance to obtain what would be the real rating of an individual for a given movie. This real rating (referred as $rr(u, m)$ with range [0..10]) will be later used to evaluate the performance of our recommender.

Our recommendation method is based on an individual recommender that estimates the rating $pred(u, m)$ given by a user to a movie. This recommender has been implemented using the jCOLIBRI framework [13] and follows the collaborative filtering approach described in [18] based on the Pearson Correlation. This method requires a population of previous individuals that have rated several items (movies). These users and their ratings are also generated synthetically.

Finally, the personality of each user is assigned according to the probability distribution inferred from the 50 volunteers that took part in our first experiments. It is important to note that we could also apply the distribution used by the original TKI test[1].

On each round of the simulation (there are 15 rounds per group) we generate a random movie listing composed of 10 movies. Our group recommender predicts what is the best movie for the group $gpred(G, m)$. Then, the proposed movie is compared with the real preferences of each individual to compute their instant satisfaction and the global satisfaction of the group. To obtain the instant satisfaction we order the movie listing for each individual according to the real rating that she would assign to each movie. Next we compare what is the position of the movie selected for the group in that list. Instant satisfaction will be higher if $gpred(G, m)$ is in the first positions and lower it is at the end of the list. Instead of using directly the position of the movie recommended for the group in the individual ordered list, we slightly modify that position according to the personality of the individual. A user with a strong personality won't be happy if the movie is in the second or third position of her preference list because she will expect to see her first favourite movie. On the other hand, an individual with mild personality won't mind to watch a movie in the middle of her preferences list. We refer to this value as the *dislike* factor. And it is linearly weighted to compute the instant satisfaction:

$$is_u = (size(ml) - dislike(u, m))/size(ml) \tag{7}$$

$$dislike(u, m) = index(m, ml) + p_u \tag{8}$$

[1] TKI personality distribution is obtained from a population of 8000 individuals from U.S.A.

where ml is the movie listing proposed to the users, and $size(ml)$ returns its size. $m \in ml$ is the movie recommended by our system for the group. The position of a movie in the movie listing once ordered according to the user preferences is obtained by means of the $index(m, ml)$ function. Finally, p_u is the personality of the user with range $[-1, 1]$.

Once is_u is obtained, the global satisfaction s_u is obtained. We have configured a δ value of 0.5 to represent balanced impact of previous satisfaction.

4.2 Results

We have run the experiment with the three aggregation functions: average satisfaction (AS), least misery (LM) and most pleasure (MP). These aggregation functions combine the individual prediction for each movie. This prediction is obtained by means of the Personality-based method. It is computed as follows: (1) the individual estimation of the rating given to the movie is returned by the collaborative individual recommender. (2) This rating is weighted according to the personality of the user p_u. (3) Resulting rating is again modified according to the user satisfaction s_u. Step (3) tries to promote those movies that have a high estimated rating for an unsatisfied user. On the other hand it decreases the final rating of movies with low estimation (to avoid their selection for the group). If a user is satisfied, ratings are slightly modified. Analogously, step (2) takes into account the personality of the user to promote the movies preferred by users with strong personalities.

In our evaluation we have studied the effect of the previous modifications. We refer to BASE when we only apply step (1) to obtain the individual prediction (note that it is the standard aggregation function and the baseline of our metric). The measures including only steps (1) and (2) are referred as PERSONALITY as they only take into account the personality factor. Finally, the complete method including steps (1), (2) and (3) is named as (PERSONALITY_MEMORY) because it includes both the personality factor and the satisfaction memory.

To measure the performance of the group recommender we use the average accumulative satisfaction of the group. The accumulative satisfaction is the sum of the individual satisfaction s_u of a user after n cinema events. This way, a user that had a high satisfaction in several events will finish with a high accumulative satisfaction and a user that was not taken into account will have a low value. To reflect the satisfaction of the group we compute the mean of the accumulative satisfaction of every user in that group.

However, our goal is to maximize the mean satisfaction but minimizing the standard deviation. The mean represents the global satisfaction of the group and reflects the goodness of the items recommended by our system. On the other hand, the standard deviation reflects the differences in the satisfaction levels within the group. It is a measure of the social fairness. As conclusion, our evaluation function is the mean value minus the standard deviation of the accumulative satisfaction of the group ($\bar{x} - \sigma$), where a higher is better.

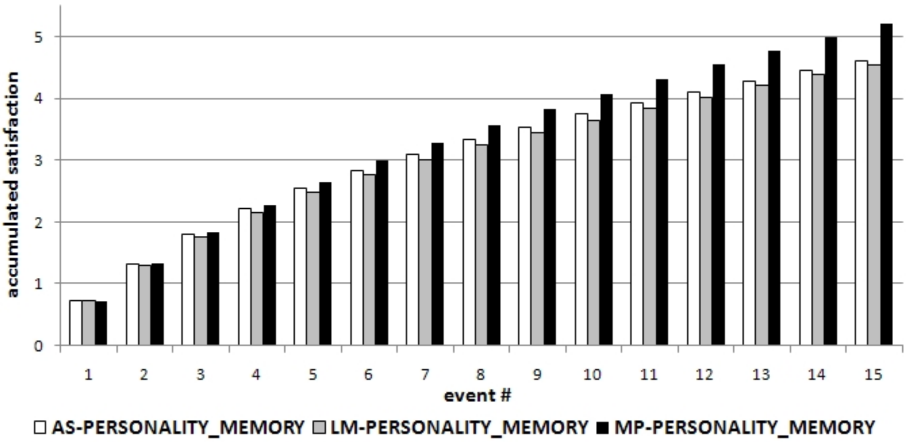


Fig. 4. Evaluation results

Results of this evaluation function are shown in Figure 4. For clarity reasons BASE and PERSONALITY methods are not shown. All of them obtain worse results than the PERSONALITY_MEMORY approach. Using just personality is slightly better than the BASE methods. As we could expect, the inclusion of personality and memory provides the best performance, being the most pleasure (MP) aggregation function the optimal approach. It is followed by average satisfaction (AS) and finally we find least misery (LM). This behaviour can be explained thinking about the nature of these aggregation functions and the bias that we promote when including the memory of user satisfaction. The main consequence of including the memory of user satisfaction is that we minimize the standard deviation within the group (i.e. maximize the fairness). By definition, LM gives not very good results in average but maximizes the fairness. However, AS and MP provide respectively good and very good recommendations in average but don't care about the fairness. With the inclusion of the satisfaction in the estimation functions we remove this drawback found in AS and MP. Therefore, they provide high values in average and minimize the standard deviation. Concretely, MP returns the item with the higher rating for the user, even if that item is a very worse recommendation for other users. This fact maximizes the mean satisfaction, and our bias ensures that others' satisfaction does not decrease too much. This is the reason it obtains the highest performance.

When analysing the mean and the standard deviation separately, we have not found very significant differences in the mean satisfaction. However the standard deviation is highly influenced by the inclusion of the memory. We have also measured the impact of modifying the size of the group, the movie listing length and the number of events, not finding any correlation between these variables and the system performance.

5 Conclusions

In this paper we have introduced our Facebook application Happy Movie. It is a group recommender for the movies domain that takes advantage of the social variables available in social networks that can be exploited to improve the performance of the system. We propose the inclusion of the following social factors: personality of every group member, trust between users, and a memory of users satisfaction to promote fairness. In this paper we have focused in this last factor –memory of users satisfaction– as we propose a CBR approach to modify the items presented to the group depending on the evolution of this satisfaction.

Our approach can be applied with several aggregation functions –that provide global recommendation for the group– and different estimation measures that predict the rating a user would assign to a given item. We have run an experiment with synthetic data to obtain the best approach for the *HappyMovie* application. Results show that optimal performance is obtained by means of the most pleasure aggregation function together with the inclusion of personality and memory in the estimation. Our future work consists on confirming these results with real users that could use our application to organize their cinema events.

Acknowledgments. Supported by Spanish Ministry of Science & Education (TIN2009-13692-C03-03) and Madrid Education Council and UCM (Group 910494).

References

1. Recio-García, J.A., Jimenez-Diaz, G., Sánchez-Ruiz, A.A., Díaz-Agudo, B.: Personality aware recommendations to groups. In: Bergman, L.D., Tuzhilin, A., Burke, R.D., Felfernig, A., Schmidt-Thieme, L. (eds.) Procs. of the 2009 ACM Conference on Recommender Systems, pp. 325–328. ACM, New York (2009)
2. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B.: Social based recommendations to groups. In: Procs. of the 14th UK Workshop on Case-Based Reasoning, pp. 46–57. CMS Press, University of Greenwich (2009)
3. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B.: Personality and social trust in group recommendations. In: Procs. of the 22th International Conference on Tools with Artificial Intelligence, ICTAI 2010 (to appear, 2010)
4. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B., Jiménez-Díaz, G.: Social factors in group recommender systems. In: ACM-TIST, TIST-2011-01-0013 (to be published, 2011)
5. Masthoff, J., Gatt, A.: In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. User Modeling and User-Adapted Interaction 16, 281–319 (2006)
6. Barsade, S.G.: The ripple effect: Emotional contagion and its influence on group behavior. Administrative Science Quarterly 47, 644–675 (2002)
7. Hatfield, E., Cacioppo, J., Rapson, R.: Emotional Contagion. Studies in Emotion and Social Interaction. Cambridge University Press, Cambridge (1994)

8. McCarthy, J.F., Anagnost, T.D.: MusicFX: An arbiter of group preferences for computer aupported collaborative workouts. In: CSCW 1998: Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work, pp. 363–372. ACM, New York (1998)
9. Crossen, A., Budzik, J., Hammond, K.J.: Flytrap: intelligent group music recommendation. In: IUI, pp. 184–185 (2002)
10. Baccigalupo, C., Plaza, E.: A case-based song scheduler for group customised radio. In: Weber, R., Richter, M.M. (eds.) ICCBR 2007. LNCS (LNAI), vol. 4626, pp. 433–448. Springer, Heidelberg (2007)
11. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues,methodological variants, and system approaches. Artificial Intelligence Communications 7, 39–59 (1994)
12. Thomas, K., Kilmann, R.: Thomas-Kilmann Conflict Mode Instrument, Tuxedo, N.Y. (1974)
13. Díaz-Agudo, B., González-Calero, P.A., Recio-García, J.A., Sánchez-Ruiz-Granados, A.A.: Building cbr systems with jcolibri. Sci. Comput. Program. 69, 68–75 (2007)
14. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)
15. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: a recommender system for groups of users. In: ECSCW 2001: Proceedings of the Seventh Conference on European Conference on Computer Supported Cooperative Work, pp. 199–218. Kluwer Academic Publishers, Norwell (2001)
16. Masthoff, J.: Group modeling: Selecting a sequence of television items to suit a group of viewers. User Modeling and User-Adapted Interaction 14, 37–85 (2004)
17. Bobadilla, J., Serradilla, F., Hernando, A.: Collaborative filtering adapted to recommender systems of e-learning. Knowl.-Based Syst. 22, 261–265 (2009)
18. Kelleher, J., Bridge, D.G.: An accurate and scalable collaborative recommender. Artif. Intell. Rev. 21, 193–213 (2004)

Chapter 14

Group recommendation methods for social network environments

14.1 Citation

Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B. Group recommendation methods for social network environments in: 3rd Workshop on Recommender Systems and the Social Web, en el 5th ACM International Conference on Recommender Systems, RecSys'11, Chicago, IL, USA, 2011. ISBN 978-1-4503-0683-6. RecSys is ranked as CORE B in the CORE ranking.

14.2 Contributions covered by this paper

In this paper we have made an analysis of the different social recommenders that can be built following our social approach. We have tested their performance in our social group recommender application *HappyMovie* and determined which aggregation functions perform better with which group configurations. Besides we have proven the importance of including social factors in group recommendation processes by performing several experiments with and without the social factors where the results have shown that group recommender configurations that use social factors always produce the best recommendations.

HappyMovie: A Facebook Application for Recommending Movies to Groups

Lara Quijano-Sanchez
Complutense University of Madrid
lara.quijano@fdi.ucm.es

Juan A. Recio-Garcia
Complutense University of Madrid
jareciog@fdi.ucm.es

Belen Diaz-Agudo
Complutense University of Madrid
belend@sip.ucm.es

***Abstract*—The goal of this paper is to show a movie recommender system for groups of people, integrated in the social network Facebook through an application called *HappyMovie*. This application tries to mitigate certain limitations in existing group recommender systems, like obtaining the users profile or offering trading methods for users in order to reach a final agreement. The method used to make the group recommendation is based on three important features: personality, social trust and memory of past recommendations. This way we simulate in a more realistic way the argumentation process followed by groups of people when deciding a joint activity.**

***Keywords*-Recommender Systems; Groups; Social Networks; Personality; Trust**

I. INTRODUCTION

Nowadays, it is very difficult to find a web site that does not provide some type of recommender system to guide users in the acquisition of products. The huge growth in the volume and complexity of information and the wide set of search activities that people can daily perform has unleashed this type of systems. We perform a lot of our everyday activities in the company of other people (like going to restaurants, cinemas...). Group recommendations take into account the interests of many people at the same time. Therefore, it is necessary to study how to combine the preferences of the individuals, prioritizing certain considerations when doing it. There are many existing systems and open lines of research in this area [1], [2], [3]. A new interesting research line has recently emerged related to Social Networks. This is due to the facility that group recommenders have in taking advantage of these networks, by using the information contained in them in order to improve their performance.

Group recommendation approaches are typically based on generating an aggregated preference using the user's individual preferences. As stated in [4] the main approaches are (a) merging the recommendations made for individuals, (b) aggregation of ratings for individuals and (c) constructing a group preference model. Masthoff [5] presents a compilation of the most important preference aggregation techniques. These basic approaches merge the ratings predicted individually for each item to calculate a global prediction for the group. The selection of a proper aggregation strategy is a key element in the success of the recommendation. A matter that we have taken into account when designing our application, as we present in Section II.

We have developed a movie group recommender connected to Facebook where the final recommendation is influenced by the personality of each member of the group and the way in which they are connected through their social relationships, basically of friendship, defined in the social network. In this way we can offer a product to a group of people that fits the individual needs of every member and tries to achieve a maximized satisfaction. Note that the proposed methods can be applied to other domains and social networks.

Our group recommendation method is explained in Section II. Section III introduces our Facebook application *HappyMovie*. Finally Section IV concludes the paper.

II. GROUP RECOMMENDATION METHOD

We have developed a group recommendation method which is based on the typical preference aggregation approaches. These approaches [2], [3] aggregate the users individual predicted ratings $pred(u,i)$ to obtain an estimation for the group $\{gpred(G,i)|u \in G\}$. Then the item with the highest group predicted scoring is proposed.

$$gpred(G,i) = \bigsqcup_{\forall u \in G} pred(u,i) \tag{1}$$

Here G is a group of users, which user u belongs to. This function provides an aggregated value that predicts the group preference for a given item i. By using this estimation, *our group recommender proposes the set of k items with the highest group predicted scoring*. In our proposal, we modify the individual ratings with the personality, trust and satisfaction factors. This way, we modify the impact of the individual preferences as shown in Equation 2.

$$\begin{aligned} gpred(G,i) &= \bigsqcup_{\forall u \in G} pred'(u,i) \\ pred'(u,i) &= \bigsqcup_{\forall v \in G} f(\, pred(u,i)\, ,\, p_u\, ,\, t_{u,v}\, ,\, s_u) \end{aligned} \tag{2}$$

where $gpred(G,i)$ is the group rating prediction for a given item i, $pred(u,i)$ is the original individual prediction for user u and item i, p_u is the personality value for user u, $t_{u,v}$ is the trust value between users u and v, and s_u is the satisfaction of user u within the group.

1082-3409/11 $26.00 © 2011 IEEE
DOI 10.1109/ICTAI.2011.44

A wide set of aggregation functions has been devised for combining individual preferences [5], being the average and least misery the most commonly used:

- **Average Satisfaction:** Computes the average of the predicted ratings of each member of the group. The function that represents this strategy is:

$$gpred(G,i) = \frac{1}{|G|}\sum_{u\in G} pred'(u,i) \tag{3}$$

  Where $pred'(u,i)$ is the predicted rating for each user u, and every item i. $gpred(G,i)$ is the final rating of item i for the group.
- **Least Misery:** Considers that a group is as happy as its least happy member. The final list of ratings is the minimum of each of the individual ratings.

$$gpred(G,i) = \min_{u\in G} pred'(u,i) \tag{4}$$

Once we have introduced the typical aggregation approaches we can explain the estimation functions. We use two different methods to compute $pred'_{i,u}$, that as we have explained before, is a modification of the predicted rating for a user according to the personality, trust and satisfaction factors. The main ideas of these approaches are explained below:

- **Delegation-based method:** The idea behind this method is that users create their opinions based on the opinions of their friends. The estimation of the delegation-based rating ($dbr(u,i)$) given an user u and an item i is computed in this way:

$$pred'(u,i) = dbr(u,i) = \frac{1}{|\sum_{v\in G} t_{u,v}|} \sum_{v\in G\wedge v\neq u} t_{u,v}\cdot(\, pred(v,i) + p_v \,) \tag{5}$$

  In this formula, we take into account the recommendation $pred_{v,i}$ of every friend v for item i. This rating is increased or decreased depending on her personality (p_v), and finally it is weighted according to the level of trust ($t_{u,v}$). Note that this formula is not normalized by the group size and uses the accumulated personality. Therefore, this formula could return a value out of the ratings range. As we are only interested in giving a final ordered list of the users preferences in the products of a given catalogue, it is not necessary to normalize the results given by our formula.
- **Influence-based method:** This method simulates the influence that each friend has in a given person. Instead of creating a new preference, it supposes that the user may modify her preference for an item depending on the preferences given by her friends to the same item, as shown in the following formula:

$$pred'(u,i) = ibr(u,i) = \; pred(u,i) \; + (1-p_u) \frac{\sum_{v\in G\wedge v\neq u} t_{u,v}\cdot(\, pred(v,i) - pred(u,i)\,)}{|G|-1} \tag{6}$$

  In this formula, the individual rating for the item ($pred_{u,i}$) is modified according to its difference with the ratings of other users ($pred_{v,i} - pred_{u,i}$). This difference takes into account the trust between users ($t_{u,v}$). Finally, the accumulated difference is weighted according to our personality in an inverse way ($1-p_u$).

Regarding the impact of our method, we point readers to [6], [7]. These papers report an average improvement of 12% when including personality and social factors in the group recommendation process. To prove this, we tested our methods in the movie recommendation domain with a group of real users. We used groups of different size and personal preferences, where we proved that by using the *Delegation-based method* or the *Influence-based method* we do improve the results of the recommendations. We created 3 different group recommender systems, a standard recommender that only aggregates preferences; and two recommenders that reflect our theories, one with the *Delegation-based method* and the other with the *Influence-based method*. When we studied the performance of this experiment results confirmed our theories, and showed better results when combining personality, satisfaction and social trust in the group recommendation.

Next we present our movie group recommender application *HappyMovie*.

III. A Group Recommendation Application of Movies: *HappyMovie*

With our Facebook application *HappyMovie* we provide group recommendations to users connected through social networks. We have included our previously developed group recommendation methods [6], [7] to a public application where everybody can benefit of it. Next we explain what users can expect of *HappyMovie*.

A. A global view of the application

In order to enter to *HappyMovie* users have just to reproduce what they do to access any other application in Facebook. The main page of the application (Figure 1) shows three buttons: Perform preferences test, Perform personality test and Create new event. Along with these buttons, we can see the number of pending invitations to join events that the user has and the events that she is attending to. However, as we describe next, these options are not always available. In the main page the system checks that the user has answered both the preferences test (see Figure 3) and the personality test (see Figure 2). Until they have been completed, the possibility of creating new events or accepting invitations is disabled. After having answered the two mentioned test, the personality test button disappears (as it can only be answered once). However the preferences test can be accessed at any time. This is because the more accurate idea of the individual preferences of the user that the recommender has, the best prediction the recommender will provide. Under the list of invitations, all the events that the user is attending appear. The system is in charge of erasing the events when the final date has expired and also to erase all the users that were attending to that event. When a user has correctly answered to both tests she has full access to *HappyMovie*'s functionality, creating new events, inviting friends to events, extending their preferences, see the events..etc. We now proceed to detail each of the different actions that a user can carry out:

- **Accepting an invitation**: In this page all the pending invitations of the user are shown giving her the possibility to accept or refuse.
- **Create an Event**: (See Figure 4) the mechanism is very similar to the one that Facebook has. It consists on four fields which are mandatory: Name of the event, when and where it will take place and last day to join the event (deadline). When users fill up the questionnaire the event is created and it is shown in the main page.
- **Events**: This page, as we can see in Figure 5, contains all the corresponding data about the event: Assistants to the event, celebration place, data of the event, deadline data, wall of the event, inviting to friends button, button to erase yourself from the event, button to return to the main page. Everyday the best three new recommendations for the current group that the system finds are proposed, actualizing itself when the movie listing from the selected city changes and/or when a new user enters or leaves the event. Initially a user can invite friends and erase herself from the event, but when the deadline date arrives these two options are disabled, leaving the group fixed as it was in that moment and giving the final three movies to watch. At this point a possibility to vote this final three recommendations will be qualified in order to give a final recommendation.
- **Inviting friends to the event**: This action is only possible inside events, giving to the users friends the possibility to join the event. When clicking the button a new questionnaire appears with a list of all the Facebook friends of the user. When the invitation is send, the questionnaire sends right into the Facebook profiles of the selected friends an invitation, giving them the option to accept it and enter to *HappyMovie* or reject it. This button will only be available while the deadline date of the event has not yet arrived.
- **Erase from an event**: It allows the user to erase herself from the event she has previously accepted to join. This option is only shown in the main page of each event, and will also be disabled when the deadline date is reached.

Figure 1. *HappyMovie* Initial Main Page

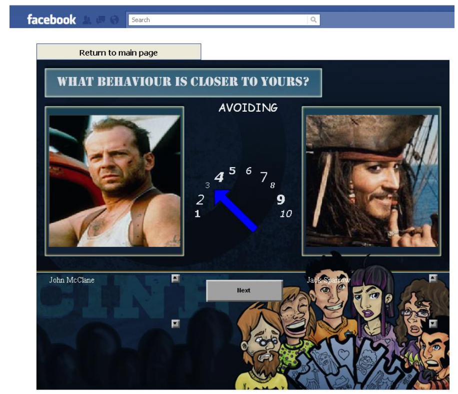


Figure 2. Personality test in *HappyMovie*

B. A modular architecture for group recommendations

Our goal with *HappyMovie* is to move the typical local systems into a new class of Web systems where the social relations are taken into account in the process of making recommendations to groups. With this type of applications we are able to offer recommendations to groups for all the people connected to these social networks. Besides we can obtain a lot of information from them without having to bother our users with a lot of questionnaires. It also provides us a lot of feedback that allows us to improve our methodology.

The architecture of *HappyMovie* is represented in Figure 6. The application is divided in seven different modules: TKI Metaphor, Facebook Profile Analysis, Satisfaction Data Base, Web Test, Web Crawling, Content Based Estimation and *HappyMovie*'s group recommender. Next sections explain what are the basis of each of these modules.

1) TKI Metaphor: There are different tests that can be used in order to obtain the different roles that people play when interacting in a decision making process. The one that we used in our previous studies [6], [7] was the TKI test [8] that consists on 30 multiple choice questions where the user has to decide how she will react in the exposed situation. We can describe an individual's behaviour along two basic dimensions in conflict situations: (1) assertiveness, the extent

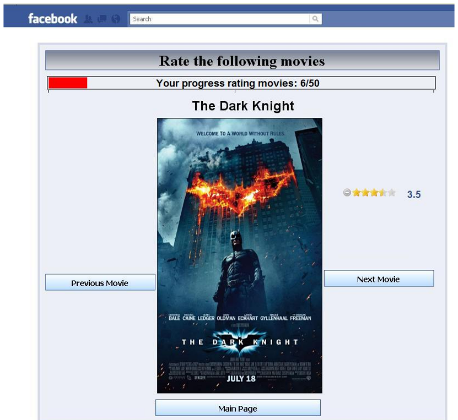


Figure 3. Preferences test in *HappyMovie*

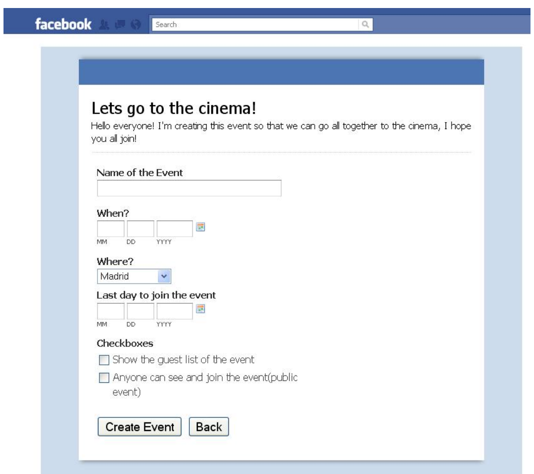


Figure 4. How to create an activity in *HappyMovie*

Figure 5. How events look like in *HappyMovie*

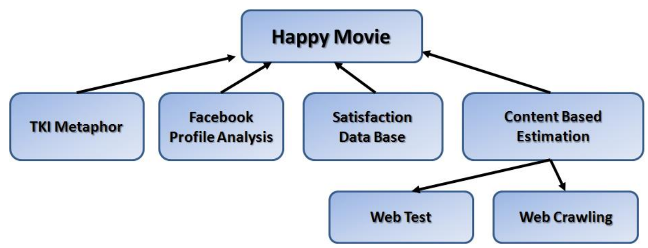


Figure 6. Facebook application architecture

to which the person attempts to satisfy her own concerns, and (2) cooperativeness, the extent to which the person attempts to satisfy the other person's concerns. These two basic dimensions of behaviour define five different modes of responding to conflict situations: Competing, Accommodating, Avoiding, Collaborating and Compromising.

Our approach combines these 5 modes to obtain a value representing the users personality. To obtain the score that the user has in each mode, the TKI personality test, proposes 30 situations where the user has to think how she will react. When we asked our users about the test, they described it as tedious, long and not very clear in some of the questions. To make the application more easy going we have studied the possibility of using a movie metaphor as an alternative method to obtain the users personality. This interactive metaphor consists on displaying two movies characters with opposite personalities for each of the five possible categories. One character represents the essential characteristics of the category, while the other one represents all the opposite ones. What the user has to do is to choose with whom of each pair of characters she feels more identified by simple moving an arrow. In order to determine which of the ways of testing the personality is better, the traditional test or the movie metaphor, we performed the following experiment:

Our goal was to evaluate if we could replace the TKI test with our metaphor in the personality module. We proved that we enhance the recommendations for a group using TKI test to perform an estimation of the user personality [6], so the new approach should perform equally. We also wanted to study if it is worth replacing the TKI test in order to make the application more usable and entertaining. To evaluate these two goals we measured 3 factors:

1) The time that takes to answer both tests. With this information we determine which one is more dynamic and less tedious.
2) Which of both tests was the one that our users preferred.
3) The difference in the results. We needed to measure if the results of the new test are similar enough to the

ones provided by the TKI test in order to replace one with the other, because we know that the results of the TKI test are a good estimation of the personality and provide good results with the group recommender system.

To analyse these factors we asked 50 users to answer to both tests, marking which one they preferred and specifying how long it took to answer each one. The results that we obtained are:

- It took an average of 15 minutes to answer to the TKI test and 5 minutes to complete the movie metaphor test. So the movie metaphor is proved to be more dynamic and less tedious.
- 100% of the users pointed out that they preferred the movie metaphor test.
- We compared the results that the TKI test provided for the five different personality modes (Competing, Accommodating, Avoiding, Collaborating and Compromising) with the results that the movie metaphor test offered for those five categories. The test that we used was the Mean Absolute Error (MAE) [9]. The average of the MAE results with the five personality modes was of 0.24 in a range of [0,2], which means an estimated error of 12%.

From these results we can conclude that it is possible replace the TKI personality test with the movie metaphor test because it provides a good estimation of the personality mode and it is suitable for our recommendation method. Additionally this test makes the application more usable and entertaining, from the users' point of view.

2) Facebook Profile Analysis: This trust module is the module that has more benefits due to embedding the application in a social network. Previously, with a standalone application, the task of obtaining the data required to compute the trust between users was very tedious. Now, we are able to calculate the trust between users extracting the specific information from each of their own profiles in the social network. Users in Facebook can post on their profiles a huge amount of personal information that can be analysed to compute the trust with other users: likes and interests, personal information, pictures, games...

We have used the method proposed in our previous studies [6], [7] to calculate the trust between users. The use of social networks and trust when building a recommender system is not new [10], [11]. To perform this task we reviewed several existing works [12], [13] and selected the most relevant and feasible factors. In order to move from theory to practice it is important to take into account that these elements are not easy to quantify and that obtaining them is limited by the purchasing power that Facebook APIs give us. In *HappyMovie* we analyse the following factors: common friends, pictures in common, common interests (music, movies, series..) and comments on each others Facebook walls. We have adjusted the weights of these factors when calculating the trust after an experiment with real users where they indicated us the real trust that they had between each other. The trust calculation is done every time that a user joins an event with all of the users who are also attending to it. It is only calculated one time for each pair of users. However, these values are updated periodically as Facebook profiles keep changing and so does the trust between two persons.

3) Satisfaction Data Base: In this module we store all the recommendations made for every user and group. Having recommendations with memory allows our system to avoid repeating previous recommendations, and ensures a certain degree of fairness. If one member accepts a proposal that she is not interested in, next time she will have some kind of preference. In the long run all the members of the group will be equally satisfied. The storage of the satisfaction consists on a data base where a value that represents the satisfaction of each user is stored. This value can later be applied to our recommendation formulas and modify the final result, giving a bigger influence to those users who are less satisfied. The satisfaction measure is updated every time that a user gets a recommendation.

We recall that when a user joins an event, and the deadline date has expired, she has the possibility of rating the three proposed movies. These ratings are stored in the data base and reflect how happy the user is with the obtained recommendations. Later on, in order to calculate the final global satisfaction of the group or the new satisfaction value of the user we just need to compare the results obtained in this test with the final proposed movie (This final movie is the one that has obtained the higher scoring in average).

4) Web Test: The goal of this preferences test is to know the taste in movies of users. When the individual recommendation is made these preferences will be taken into account discriminating the different movies according to the users preferences of actors, genre... In order to complete the test, users must rate at least 20 movies with a 5 star voting system, the progress of each user rating movies is shown at the top of the preferences test with a bar. Users have the possibility of modifying their ratings each time they use the application as new movies are periodically added to the test. This way we allow our users to build a more solid and up to date profile of their individual preferences. Consequently, this fact will have a positive impact in the performance of the recommender system. They also have the possibility of not rating every movie in the test either because they have not seen it or because they do not want to.

5) Web Crawling: This module searches the web and finds the movie listing of the city that has been selected. Once it has that information it obtains the description of each of the movies in the movie listing. Later, it analyses all the descriptions and extracts all the data required to define the movie. Each specific characteristic of the movie is a field that the individual recommender contrasts. The recovered items, with all their specific information, are sent to the Content Based Estimation module and to the group recommender module as they are the products to be recommended.

6) Content Based Estimation: This is the individual recommender module, it is built using the jCOLIBRI framework extension to build recommender systems [14] and follows a content based approach [15] that uses descriptions of the products to be recommended and returns the collection of products that are more similar to the aimed product. As it is a content based recommender system it manages a case base of products, and the recommender compares all the considered items to be recommended with this case base. This case base is different for each user and has the information retrieved from the Web Test module.

C. Testing Stage

For the realization of the testing stage we counted with the collaboration of a group of friends who have Facebook accounts. We asked them use our application to create events to go to the cinema. The testing consisted in performing all the different possible actions that the application has to offer, and the results of all of them have been positive, being our application fully operative at the moment.

IV. Conclusions

In this paper we have presented our Facebook application *HappyMovie*, a group recommender system based on the personality of each user and the trust among the people in the group. Our application benefits from the data stored in the social network Facebook, and uses it to complete the information about users that our system needs. In our previous works [6], [7] we presented a standalone group recommender. The contribution of our current work is embedding the application in Facebook, making it more reachable to everybody who has an account in it, and taking advantage of the social network information about its users. We have also modified our previous technique of obtaining the personality of each user and proved that with our new method, that consists on a movie metaphor test, our users are more satisfied because they spend less time answering the tests in order to build their recommender profile. Moreover, it is possible to replace the TKI test, that is the one that we used previously, with the new one because the results obtained with both tests are equitable.

In our work we have employed a lot of different aggregation functions to generate the recommendations for groups. We have also implemented two different methods to perform the group recommendation, the *delegation based method* and the *influence based method*. Any of these methods or aggregations functions can be chosen to operate in *HappyMovie*. We plan to evaluate the impact of these aggregation functions in the accuracy of our approach and to include them in our adaptive group recommender, where the recommendation algorithm adapts itself to the personality distribution of the group, its size and other characteristics.

References

[1] J. F. McCarthy and T. D. Anagnost, "MusicFX: An arbiter of group preferences for computer aupported collaborative workouts," in *CSCW '98*. ACM, 1998, pp. 363–372.

[2] J. Masthoff and A. Gatt, "In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems," *User Modeling and User-Adapted Interaction*, vol. 16, no. 3-4, pp. 281–319, 2006.

[3] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl, "Polylens: a recommender system for groups of users," in *ECSCW'01*, USA, 2001, pp. 199–218.

[4] A. Jameson and B. Smyth, "Recommendation to groups," in *The Adaptive Web, Methods and Strategies of Web Personalization*, ser. Lecture Notes in Computer Science, vol. 4321. Springer, 2007, pp. 596–627.

[5] J. Masthoff, "Group modeling: Selecting a sequence of television items to suit a group of viewers," *User Modeling and User-Adapted Interaction*, vol. 14, no. 1, pp. 37–85, 2004.

[6] L. Quijano-Sánchez, J. A. Recio-García, and B. Díaz-Agudo, "Personality and social trust in group recommendations," in *Conference on Tools with Artificial Intelligence, ICTAI'10.* IEEE Computing Society, 2010, pp. 121–126.

[7] L. Quijano-Sánchez, J. Recio-García, B. Díaz-Agudo, and G. Jiménez-Díaz, "Social factors in group recommender systems," in *ACM-TIST, TIST-2011-01-0013*. in press, 2011.

[8] K. Thomas and R. Kilmann, *Thomas-Kilmann Conflict Mode Instrument*. Tuxedo, N.Y., 1974.

[9] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

[10] J. A. Golbeck, "Computing and applying trust in web-based social networks," Ph.D. dissertation, College Park, MD, USA, 2005, chair-Hendler, James.

[11] P. Avesani, P. Massa, and R. Tiella, "A trust-enhanced recommender system application: Moleskiing," in *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*. NY, USA: ACM, 2005, pp. 1589–1593.

[12] E. Gilbert and K. Karahalios, "Predicting tie strength with social media," in *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*. ACM, 2009, pp. 211–220.

[13] J. Golbeck, "Combining provenance with trust in social networks for semantic web content filtering," in *Provenance and Annotation of Data, IPAW '06*, ser. Lecture Notes in Computer Science, vol. 4145. Springer, 2006, pp. 101–108.

[14] B. Díaz-Agudo, P. A. González-Calero, J. A. Recio-García, and A. A. Sánchez-Ruiz-Granados, "Building cbr systems with jcolibri," *S. C. P.*, vol. 69, pp. 68–75, 2007.

[15] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*, 2007, pp. 325–341.

Chapter 15

HappyMovie: A Facebook Application for Recommending Movies to Groups

15.1 Citation

Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B. HappyMovie: A Facebook Application for Recommending Movies to Groups, in: Procs. of the 23th International Conference on Tools with Artificial Intelligence, ICTAI'11., IEEE Computing Society, Boca Raton, Florida, USA, pages: 239-244, 2011. ISBN 978-1-4577-2068-0. ICTAI is ranked as CORE B in the CORE ranking.

15.2 Contributions covered by this paper

In this paper we introduce the usage and functionality of our social group recommender application *HappyMovie*. With this paper we confirm users low resentment when having to answer to the different tests presented in the application and conclude that this is due to our Facebook oriented model that is user-friendly, easily accessible, it has a lot of daily users and it's adapted to run questionnaires, applications and games.

HappyMovie: A Facebook Application for Recommending Movies to Groups

Lara Quijano-Sanchez
Complutense University of Madrid
lara.quijano@fdi.ucm.es

Juan A. Recio-Garcia
Complutense University of Madrid
jareciog@fdi.ucm.es

Belen Diaz-Agudo
Complutense University of Madrid
belend@sip.ucm.es

***Abstract*—The goal of this paper is to show a movie recommender system for groups of people, integrated in the social network Facebook through an application called *HappyMovie*. This application tries to mitigate certain limitations in existing group recommender systems, like obtaining the users profile or offering trading methods for users in order to reach a final agreement. The method used to make the group recommendation is based on three important features: personality, social trust and memory of past recommendations. This way we simulate in a more realistic way the argumentation process followed by groups of people when deciding a joint activity.**

***Keywords*-Recommender Systems; Groups; Social Networks; Personality; Trust**

I. INTRODUCTION

Nowadays, it is very difficult to find a web site that does not provide some type of recommender system to guide users in the acquisition of products. The huge growth in the volume and complexity of information and the wide set of search activities that people can daily perform has unleashed this type of systems. We perform a lot of our everyday activities in the company of other people (like going to restaurants, cinemas...). Group recommendations take into account the interests of many people at the same time. Therefore, it is necessary to study how to combine the preferences of the individuals, prioritizing certain considerations when doing it. There are many existing systems and open lines of research in this area [1], [2], [3]. A new interesting research line has recently emerged related to Social Networks. This is due to the facility that group recommenders have in taking advantage of these networks, by using the information contained in them in order to improve their performance.

Group recommendation approaches are typically based on generating an aggregated preference using the user's individual preferences. As stated in [4] the main approaches are (a) merging the recommendations made for individuals, (b) aggregation of ratings for individuals and (c) constructing a group preference model. Masthoff [5] presents a compilation of the most important preference aggregation techniques. These basic approaches merge the ratings predicted individually for each item to calculate a global prediction for the group. The selection of a proper aggregation strategy is a key element in the success of the recommendation. A matter that we have taken into account when designing our application, as we present in Section II.

We have developed a movie group recommender connected to Facebook where the final recommendation is influenced by the personality of each member of the group and the way in which they are connected through their social relationships, basically of friendship, defined in the social network. In this way we can offer a product to a group of people that fits the individual needs of every member and tries to achieve a maximized satisfaction. Note that the proposed methods can be applied to other domains and social networks.

Our group recommendation method is explained in Section II. Section III introduces our Facebook application *HappyMovie*. Finally Section IV concludes the paper.

II. GROUP RECOMMENDATION METHOD

We have developed a group recommendation method which is based on the typical preference aggregation approaches. These approaches [2], [3] aggregate the users individual predicted ratings $pred(u,i)$ to obtain an estimation for the group $\{gpred(G,i)|u \in G\}$. Then the item with the highest group predicted scoring is proposed.

$$gpred(G,i) = \bigsqcup_{\forall u \in G} pred(u,i) \qquad (1)$$

Here G is a group of users, which user u belongs to. This function provides an aggregated value that predicts the group preference for a given item i. By using this estimation, *our group recommender proposes the set of k items with the highest group predicted scoring*. In our proposal, we modify the individual ratings with the personality, trust and satisfaction factors. This way, we modify the impact of the individual preferences as shown in Equation 2.

$$\begin{aligned} gpred(G,i) &= \bigsqcup_{\forall u \in G} pred'(u,i) \\ pred'(u,i) &= \bigsqcup_{\forall v \in G} f(\,pred(u,i)\,,\,p_u\,,\,t_{u,v}\,,\,s_u) \quad (2) \end{aligned}$$

where $gpred(G,i)$ is the group rating prediction for a given item i, $pred(u,i)$ is the original individual prediction for user u and item i, p_u is the personality value for user u, $t_{u,v}$ is the trust value between users u and v, and s_u is the satisfaction of user u within the group.

1082-3409/11 $26.00 © 2011 IEEE
DOI 10.1109/ICTAI.2011.44

A wide set of aggregation functions has been devised for combining individual preferences [5], being the average and least misery the most commonly used:

- **Average Satisfaction:** Computes the average of the predicted ratings of each member of the group. The function that represents this strategy is:

$$gpred(G,i) = \frac{1}{|G|}\sum_{u\in G} pred'(u,i) \tag{3}$$

Where $pred'(u,i)$ is the predicted rating for each user u, and every item i. $gpred(G,i)$ is the final rating of item i for the group.

- **Least Misery:** Considers that a group is as happy as its least happy member. The final list of ratings is the minimum of each of the individual ratings.

$$gpred(G,i) = \min_{u\in G} pred'(u,i) \tag{4}$$

Once we have introduced the typical aggregation approaches we can explain the estimation functions. We use two different methods to compute $pred'_{i,u}$, that as we have explained before, is a modification of the predicted rating for a user according to the personality, trust and satisfaction factors. The main ideas of these approaches are explained below:

- **Delegation-based method:** The idea behind this method is that users create their opinions based on the opinions of their friends. The estimation of the delegation-based rating ($dbr(u,i)$) given an user u and an item i is computed in this way:

$$pred'(u,i) = dbr(u,i) = \frac{1}{|\sum_{v\in G} t_{u,v}|} \sum_{v\in G\wedge v\neq u} t_{u,v}\cdot(\, pred(v,i) + p_v\,) \tag{5}$$

In this formula, we take into account the recommendation $pred_{v,i}$ of every friend v for item i. This rating is increased or decreased depending on her personality (p_v), and finally it is weighted according to the level of trust ($t_{u,v}$). Note that this formula is not normalized by the group size and uses the accumulated personality. Therefore, this formula could return a value out of the ratings range. As we are only interested in giving a final ordered list of the users preferences in the products of a given catalogue, it is not necessary to normalize the results given by our formula.

- **Influence-based method:** This method simulates the influence that each friend has in a given person. Instead of creating a new preference, it supposes that the user may modify her preference for an item depending on the preferences given by her friends to the same item, as shown in the following formula:

$$pred'(u,i) = ibr(u,i) = \; pred(u,i) \; + (1-p_u) \frac{\sum_{v\in G\wedge v\neq u} t_{u,v}\cdot(\, pred(v,i) - pred(u,i)\,)}{|G|-1} \tag{6}$$

In this formula, the individual rating for the item ($pred_{u,i}$) is modified according to its difference with the ratings of other users ($pred_{v,i} - pred_{u,i}$). This difference takes into account the trust between users ($t_{u,v}$). Finally, the accumulated difference is weighted according to our personality in an inverse way ($1-p_u$).

Regarding the impact of our method, we point readers to [6], [7]. These papers report an average improvement of 12% when including personality and social factors in the group recommendation process. To prove this, we tested our methods in the movie recommendation domain with a group of real users. We used groups of different size and personal preferences, where we proved that by using the *Delegation-based method* or the *Influence-based method* we do improve the results of the recommendations. We created 3 different group recommender systems, a standard recommender that only aggregates preferences; and two recommenders that reflect our theories, one with the *Delegation-based method* and the other with the *Influence-based method*. When we studied the performance of this experiment results confirmed our theories, and showed better results when combining personality, satisfaction and social trust in the group recommendation.

Next we present our movie group recommender application *HappyMovie*.

III. A Group Recommendation Application of Movies: *HappyMovie*

With our Facebook application *HappyMovie* we provide group recommendations to users connected through social networks. We have included our previously developed group recommendation methods [6], [7] to a public application where everybody can benefit of it. Next we explain what users can expect of *HappyMovie*.

A. A global view of the application

In order to enter to *HappyMovie* users have just to reproduce what they do to access any other application in Facebook. The main page of the application (Figure 1) shows three buttons: Perform preferences test, Perform personality test and Create new event. Along with these buttons, we can see the number of pending invitations to join events that the user has and the events that she is attending to. However, as we describe next, these options are not always available. In the main page the system checks that the user has answered both the preferences test (see Figure 3) and the personality test (see Figure 2). Until they have been completed, the possibility of creating new events or accepting invitations is disabled. After having answered the two mentioned test, the personality test button disappears (as it can only be answered once). However the preferences test can be accessed at any time. This is because the more accurate idea of the individual preferences of the user that the recommender has, the best prediction the recommender will provide. Under the list of invitations, all the events that the user is attending appear. The system is in charge of erasing the events when the final date has expired and also to erase all the users that were attending to that event. When a user has correctly answered to both tests she has full access to *HappyMovie*'s functionality, creating new events, inviting friends to events, extending their preferences, see the events..etc. We now proceed to detail each of the different actions that a user can carry out:

- **Accepting an invitation**: In this page all the pending invitations of the user are shown giving her the possibility to accept or refuse.
- **Create an Event**: (See Figure 4) the mechanism is very similar to the one that Facebook has. It consists on four fields which are mandatory: Name of the event, when and where it will take place and last day to join the event (deadline). When users fill up the questionnaire the event is created and it is shown in the main page.
- **Events**: This page, as we can see in Figure 5, contains all the corresponding data about the event: Assistants to the event, celebration place, data of the event, deadline data, wall of the event, inviting to friends button, button to erase yourself from the event, button to return to the main page. Everyday the best three new recommendations for the current group that the system finds are proposed, actualizing itself when the movie listing from the selected city changes and/or when a new user enters or leaves the event. Initially a user can invite friends and erase herself from the event, but when the deadline date arrives these two options are disabled, leaving the group fixed as it was in that moment and giving the final three movies to watch. At this point a possibility to vote this final three recommendations will be qualified in order to give a final recommendation.
- **Inviting friends to the event**: This action is only possible inside events, giving to the users friends the possibility to join the event. When clicking the button a new questionnaire appears with a list of all the Facebook friends of the user. When the invitation is send, the questionnaire sends right into the Facebook profiles of the selected friends an invitation, giving them the option to accept it and enter to *HappyMovie* or reject it. This button will only be available while the deadline date of the event has not yet arrived.
- **Erase from an event**: It allows the user to erase herself from the event she has previously accepted to join. This option is only shown in the main page of each event, and will also be disabled when the deadline date is reached.

Figure 1. *HappyMovie* Initial Main Page

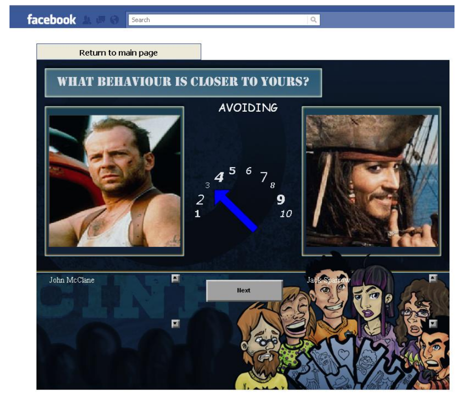


Figure 2. Personality test in *HappyMovie*

B. A modular architecture for group recommendations

Our goal with *HappyMovie* is to move the typical local systems into a new class of Web systems where the social relations are taken into account in the process of making recommendations to groups. With this type of applications we are able to offer recommendations to groups for all the people connected to these social networks. Besides we can obtain a lot of information from them without having to bother our users with a lot of questionnaires. It also provides us a lot of feedback that allows us to improve our methodology.

The architecture of *HappyMovie* is represented in Figure 6. The application is divided in seven different modules: TKI Metaphor, Facebook Profile Analysis, Satisfaction Data Base, Web Test, Web Crawling, Content Based Estimation and *HappyMovie*'s group recommender. Next sections explain what are the basis of each of these modules.

1) TKI Metaphor: There are different tests that can be used in order to obtain the different roles that people play when interacting in a decision making process. The one that we used in our previous studies [6], [7] was the TKI test [8] that consists on 30 multiple choice questions where the user has to decide how she will react in the exposed situation. We can describe an individual's behaviour along two basic dimensions in conflict situations: (1) assertiveness, the extent

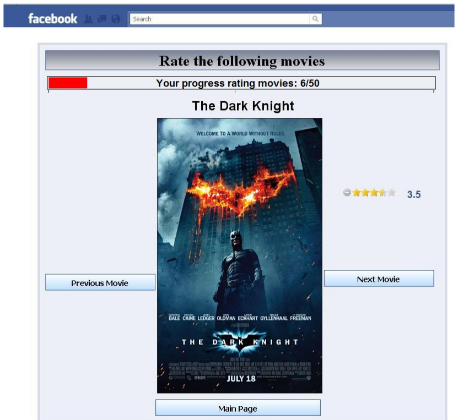


Figure 3. Preferences test in *HappyMovie*

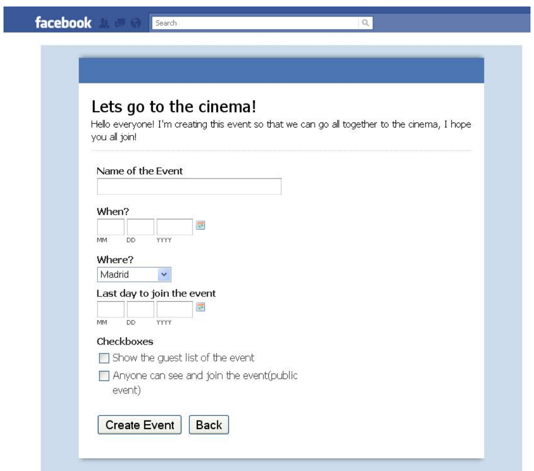


Figure 4. How to create an activity in *HappyMovie*

Figure 5. How events look like in *HappyMovie*

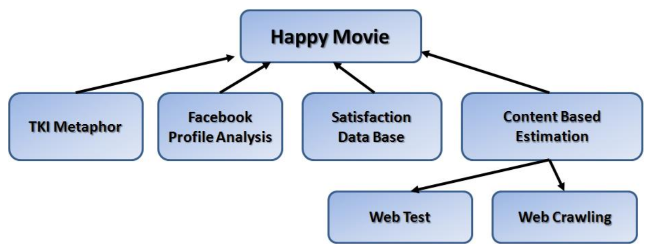


Figure 6. Facebook application architecture

to which the person attempts to satisfy her own concerns, and (2) cooperativeness, the extent to which the person attempts to satisfy the other person's concerns. These two basic dimensions of behaviour define five different modes of responding to conflict situations: Competing, Accommodating, Avoiding, Collaborating and Compromising.

Our approach combines these 5 modes to obtain a value representing the users personality. To obtain the score that the user has in each mode, the TKI personality test, proposes 30 situations where the user has to think how she will react. When we asked our users about the test, they described it as tedious, long and not very clear in some of the questions. To make the application more easy going we have studied the possibility of using a movie metaphor as an alternative method to obtain the users personality. This interactive metaphor consists on displaying two movies characters with opposite personalities for each of the five possible categories. One character represents the essential characteristics of the category, while the other one represents all the opposite ones. What the user has to do is to choose with whom of each pair of characters she feels more identified by simple moving an arrow. In order to determine which of the ways of testing the personality is better, the traditional test or the movie metaphor, we performed the following experiment:

Our goal was to evaluate if we could replace the TKI test with our metaphor in the personality module. We proved that we enhance the recommendations for a group using TKI test to perform an estimation of the user personality [6], so the new approach should perform equally. We also wanted to study if it is worth replacing the TKI test in order to make the application more usable and entertaining. To evaluate these two goals we measured 3 factors:

1) The time that takes to answer both tests. With this information we determine which one is more dynamic and less tedious.
2) Which of both tests was the one that our users preferred.
3) The difference in the results. We needed to measure if the results of the new test are similar enough to the

ones provided by the TKI test in order to replace one with the other, because we know that the results of the TKI test are a good estimation of the personality and provide good results with the group recommender system.

To analyse these factors we asked 50 users to answer to both tests, marking which one they preferred and specifying how long it took to answer each one. The results that we obtained are:

- It took an average of 15 minutes to answer to the TKI test and 5 minutes to complete the movie metaphor test. So the movie metaphor is proved to be more dynamic and less tedious.
- 100% of the users pointed out that they preferred the movie metaphor test.
- We compared the results that the TKI test provided for the five different personality modes (Competing, Accommodating, Avoiding, Collaborating and Compromising) with the results that the movie metaphor test offered for those five categories. The test that we used was the Mean Absolute Error (MAE) [9]. The average of the MAE results with the five personality modes was of 0.24 in a range of [0,2], which means an estimated error of 12%.

From these results we can conclude that it is possible replace the TKI personality test with the movie metaphor test because it provides a good estimation of the personality mode and it is suitable for our recommendation method. Additionally this test makes the application more usable and entertaining, from the users' point of view.

2) Facebook Profile Analysis: This trust module is the module that has more benefits due to embedding the application in a social network. Previously, with a standalone application, the task of obtaining the data required to compute the trust between users was very tedious. Now, we are able to calculate the trust between users extracting the specific information from each of their own profiles in the social network. Users in Facebook can post on their profiles a huge amount of personal information that can be analysed to compute the trust with other users: likes and interests, personal information, pictures, games...

We have used the method proposed in our previous studies [6], [7] to calculate the trust between users. The use of social networks and trust when building a recommender system is not new [10], [11]. To perform this task we reviewed several existing works [12], [13] and selected the most relevant and feasible factors. In order to move from theory to practice it is important to take into account that these elements are not easy to quantify and that obtaining them is limited by the purchasing power that Facebook APIs give us. In *HappyMovie* we analyse the following factors: common friends, pictures in common, common interests (music, movies, series..) and comments on each others Facebook walls. We have adjusted the weights of these factors when calculating the trust after an experiment with real users where they indicated us the real trust that they had between each other. The trust calculation is done every time that a user joins an event with all of the users who are also attending to it. It is only calculated one time for each pair of users. However, these values are updated periodically as Facebook profiles keep changing and so does the trust between two persons.

3) Satisfaction Data Base: In this module we store all the recommendations made for every user and group. Having recommendations with memory allows our system to avoid repeating previous recommendations, and ensures a certain degree of fairness. If one member accepts a proposal that she is not interested in, next time she will have some kind of preference. In the long run all the members of the group will be equally satisfied. The storage of the satisfaction consists on a data base where a value that represents the satisfaction of each user is stored. This value can later be applied to our recommendation formulas and modify the final result, giving a bigger influence to those users who are less satisfied. The satisfaction measure is updated every time that a user gets a recommendation.

We recall that when a user joins an event, and the deadline date has expired, she has the possibility of rating the three proposed movies. These ratings are stored in the data base and reflect how happy the user is with the obtained recommendations. Later on, in order to calculate the final global satisfaction of the group or the new satisfaction value of the user we just need to compare the results obtained in this test with the final proposed movie (This final movie is the one that has obtained the higher scoring in average).

4) Web Test: The goal of this preferences test is to know the taste in movies of users. When the individual recommendation is made these preferences will be taken into account discriminating the different movies according to the users preferences of actors, genre... In order to complete the test, users must rate at least 20 movies with a 5 star voting system, the progress of each user rating movies is shown at the top of the preferences test with a bar. Users have the possibility of modifying their ratings each time they use the application as new movies are periodically added to the test. This way we allow our users to build a more solid and up to date profile of their individual preferences. Consequently, this fact will have a positive impact in the performance of the recommender system. They also have the possibility of not rating every movie in the test either because they have not seen it or because they do not want to.

5) Web Crawling: This module searches the web and finds the movie listing of the city that has been selected. Once it has that information it obtains the description of each of the movies in the movie listing. Later, it analyses all the descriptions and extracts all the data required to define the movie. Each specific characteristic of the movie is a field that the individual recommender contrasts. The recovered items, with all their specific information, are sent to the Content Based Estimation module and to the group recommender module as they are the products to be recommended.

6) Content Based Estimation: This is the individual recommender module, it is built using the jCOLIBRI framework extension to build recommender systems [14] and follows a content based approach [15] that uses descriptions of the products to be recommended and returns the collection of products that are more similar to the aimed product. As it is a content based recommender system it manages a case base of products, and the recommender compares all the considered items to be recommended with this case base. This case base is different for each user and has the information retrieved from the Web Test module.

C. Testing Stage

For the realization of the testing stage we counted with the collaboration of a group of friends who have Facebook accounts. We asked them use our application to create events to go to the cinema. The testing consisted in performing all the different possible actions that the application has to offer, and the results of all of them have been positive, being our application fully operative at the moment.

IV. Conclusions

In this paper we have presented our Facebook application *HappyMovie*, a group recommender system based on the personality of each user and the trust among the people in the group. Our application benefits from the data stored in the social network Facebook, and uses it to complete the information about users that our system needs. In our previous works [6], [7] we presented a standalone group recommender. The contribution of our current work is embedding the application in Facebook, making it more reachable to everybody who has an account in it, and taking advantage of the social network information about its users. We have also modified our previous technique of obtaining the personality of each user and proved that with our new method, that consists on a movie metaphor test, our users are more satisfied because they spend less time answering the tests in order to build their recommender profile. Moreover, it is possible to replace the TKI test, that is the one that we used previously, with the new one because the results obtained with both tests are equitable.

In our work we have employed a lot of different aggregation functions to generate the recommendations for groups. We have also implemented two different methods to perform the group recommendation, the *delegation based method* and the *influence based method*. Any of these methods or aggregations functions can be chosen to operate in *HappyMovie*. We plan to evaluate the impact of these aggregation functions in the accuracy of our approach and to include them in our adaptive group recommender, where the recommendation algorithm adapts itself to the personality distribution of the group, its size and other characteristics.

References

[1] J. F. McCarthy and T. D. Anagnost, "MusicFX: An arbiter of group preferences for computer aupported collaborative workouts," in *CSCW '98*. ACM, 1998, pp. 363–372.

[2] J. Masthoff and A. Gatt, "In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems," *User Modeling and User-Adapted Interaction*, vol. 16, no. 3-4, pp. 281–319, 2006.

[3] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl, "Polylens: a recommender system for groups of users," in *ECSCW'01*, USA, 2001, pp. 199–218.

[4] A. Jameson and B. Smyth, "Recommendation to groups," in *The Adaptive Web, Methods and Strategies of Web Personalization*, ser. Lecture Notes in Computer Science, vol. 4321. Springer, 2007, pp. 596–627.

[5] J. Masthoff, "Group modeling: Selecting a sequence of television items to suit a group of viewers," *User Modeling and User-Adapted Interaction*, vol. 14, no. 1, pp. 37–85, 2004.

[6] L. Quijano-Sánchez, J. A. Recio-García, and B. Díaz-Agudo, "Personality and social trust in group recommendations," in *Conference on Tools with Artificial Intelligence, ICTAI'10.* IEEE Computing Society, 2010, pp. 121–126.

[7] L. Quijano-Sánchez, J. Recio-García, B. Díaz-Agudo, and G. Jiménez-Díaz, "Social factors in group recommender systems," in *ACM-TIST, TIST-2011-01-0013*. in press, 2011.

[8] K. Thomas and R. Kilmann, *Thomas-Kilmann Conflict Mode Instrument*. Tuxedo, N.Y., 1974.

[9] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

[10] J. A. Golbeck, "Computing and applying trust in web-based social networks," Ph.D. dissertation, College Park, MD, USA, 2005, chair-Hendler, James.

[11] P. Avesani, P. Massa, and R. Tiella, "A trust-enhanced recommender system application: Moleskiing," in *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*. NY, USA: ACM, 2005, pp. 1589–1593.

[12] E. Gilbert and K. Karahalios, "Predicting tie strength with social media," in *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*. ACM, 2009, pp. 211–220.

[13] J. Golbeck, "Combining provenance with trust in social networks for semantic web content filtering," in *Provenance and Annotation of Data, IPAW '06*, ser. Lecture Notes in Computer Science, vol. 4145. Springer, 2006, pp. 101–108.

[14] B. Díaz-Agudo, P. A. González-Calero, J. A. Recio-García, and A. A. Sánchez-Ruiz-Granados, "Building cbr systems with jcolibri," *S. C. P.*, vol. 69, pp. 68–75, 2007.

[15] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*, 2007, pp. 325–341.

Chapter 16

A Case-Based Solution to the Cold-Start Problem in Group Recommenders

16.1 Citation

Quijano-Sánchez, L., Bridge, D., Díaz-Agudo, B., Recio-García, J. A., A Case-Based Solution to the Cold-Start Problem in Group Recommenders, in: Procs. of the 20th International Conference on Case Based Reasoning, ICCBR'12, Lyon, France, pages: 342-356, 2012. ISBN 978-3-642-32985-2. ICCBR is ranked as CORE C in the CORE ranking.

Quijano-Sánchez, L., Bridge, D., Díaz-Agudo, B. y Recio-García, J. A. A case-based solution to the cold-start problem in group recommender, in Procs. of the Best Papers in Sister Conferences Track at the 23rd International Joint Conference on Artificial Intelligence, IJCAI'13. AAAI Press, Beijing, China, pages: 3042-3046, 2013. ISBN 978-1-57735-633-2. IJCAI is ranked as CORE A in the CORE ranking.

16.2 Contributions covered by this paper

In this paper we present a solution to the cold-start problem in group recommenders by including an analysis of the social factors that surround a group to identify the most similar users in the most similar group. This new social similarity identification allows us to copy ratings to users in cold-start in a more effective way.

A Case-Based Solution to the Cold-Start Problem in Group Recommenders

Lara Quijano-Sánchez[1], Derek Bridge[2],
Belén Díaz-Agudo[1], and Juan A. Recio-García[1]

[1] Department of Software Engineering and Artificial Intelligence,
Universidad Complutense de Madrid, Spain
[2] Department of Computer Science, University College Cork, Ireland
{lara.quijano,jareciog}@fdi.ucm.es, d.bridge@cs.ucc.ie,
belend@sip.ucm.es

Abstract. We extend a group recommender system with a case base of previous group recommendation events. We show that this offers a potential solution to the cold-start problem. Suppose a group recommendation is sought but one of the group members is a new user who has few item ratings. We can copy ratings into this user's profile from the profile of the most similar user in the most similar group from the case base. In other words, we copy ratings from a user who played a similar role in some previous group event. We show that copying in this way, i.e. conditioned on groups, is superior to copying nothing and also superior to copying ratings from the most similar user known to the system.

1 Introduction

Restaurants; tourist attractions; vacation destinations; movies, music & TV when broadcast in shared spaces. All these are examples of items that can benefit from *group recommender systems*, i.e. recommender systems whose suggestions take into account the preferences of the members of a group of people who will consume the items together [4]. Group recommenders typically work by either (a) merging the recommendations that would be made to the group members, (b) aggregating the predicted ratings of the group members, or (c) constructing a group preference model from the preferences of the group members [4].

In this paper, in the context of movie recommendation to groups of friends, we consider a group recommender system that takes the second of these approaches. It runs, and aggregates the results of, a *single-person recommender system* for each member of the group. Specifically, it runs a *user-based collaborative recommender system* [3] to predict movie ratings for each member of the group. It finds a neighbourhood of users who have similar movie ratings to those of the active user; it predicts user ratings for candidate movies that neighbours have rated but which the active user has not rated. The group recommender aggregates the predicted ratings for each group member to arrive at ratings and thence suggestions that it can make to the group as a whole. Methods for aggregating ratings

B. Díaz Agudo and I. Watson (Eds.): ICCBR 2012, LNCS 7466, pp. 342–356, 2012.

are reviewed in [5] and it is the *most pleasure* principle (see Section 3) that we use.

It is well-known that collaborative recommenders suffer from *cold-start problems* [3,13]. In particular, a user-based collaborative recommender finds it difficult to make good predictions for new users for whom it has few ratings: it cannot reliably find neighbours who have similar ratings to those of the new user. The group recommender inherits this problem too because it aggregates the predicted ratings for each group member. Solutions to the cold-start problem for single-person recommenders are summarized in [13]. Solutions include: non-personalized recommendations for cold-start users using population averages; intelligent ways to solicit more ratings (e.g. [2,11]); and hybrid recommenders that resort to content-based recommendations when there are insufficient ratings to make collaborative recommendations (e.g. [1,6]).

The contribution of this paper is to introduce and evaluate a *case-based reasoning* (CBR) solution to this problem. We use a case base in which each case records a previous group movie recommendation event. When a group requests a new recommendation but where one or more of the group members is in cold-start, we find a case that describes a previous recommendation event where there are users who are not in cold-start but who play similar roles in their group to the roles the cold-start users play in the active group. We copy ratings from the users in the case to the corresponding users in the active group and only then proceed to run the single-person recommender and to aggregate its results. It is natural to use a CBR approach because, in the movie domain and similar domains, similar events recur: the same group (perhaps with some small variations) repeats activities together; and some age, gender and personality distributions will tend to recur too (e.g. two adults with two children, or several friends in the same age range).

Case-based reasoning (CBR) has been used in recommender systems before (e.g. [12]) and explicit parallels between CBR and user-based collaborative recommenders have been drawn (e.g. [7]). But we are unaware of any previous use of CBR in group recommenders or in solutions to the cold-start problem.

Section 2 defines a single-person user-based collaborative recommender system; Section 3 describes two group recommenders that aggregate the predictions made for each group member by the single-person recommender; Section 4 describes how we have extended these group recommenders to use a case base of previous group recommendation events to solve the cold-start problem; Section 5 proposes systems against which the case-based system can be compared; Section 6 describes the dataset that we have used in our experiments; Section 7 presents our experimental method; Section 8 contains results; and Section 9 concludes and presents some ideas for future work.

2 Single-Person User-Based Collaborative Recommenders

As we have explained, our group recommender runs a user-based collaborative recommender for each person that is a member of the active group. Although the operation of user-based collaborative recommenders is well-known, we summarize it here in order to be explicit and to introduce some notation.

Suppose there are n users, $U = \{u : 1 \dots n\}$, and m items (e.g. movies), $I = \{i : 1 \dots m\}$. Let $r_{u,i}$ be the rating that user u assigns to item i. Ratings are on a numeric scale, e.g. 1 = terrible and 5 = excellent, but $r_{u,i} = \perp$ signals that u has not yet rated i.

Suppose we want to recommend to active user u_a one or more of a set of candidate target items $T_a \subseteq I$. For example, T_a could be the set of movies showing this week at u_a's local multiplex. The user-based collaborative recommender that we use works as follows [3,13]:

- For each $i \in T_a$,
  - The similarity between the active user u_a and each other user $u \neq u_a$ who has rated i, is computed using Pearson Correlation [3], ρ.
  - After computing the similarity between u_a and each other user u who has rated i, the k nearest neighbours are selected, i.e. the k for whom $\rho_{u_a,u}$ is highest. In our work, we use $k = 20$ and we only include neighbours for whom $\rho_{u_a,u} > 0$.
  - A predicted rating $\hat{r}_{u_a,i}$ for active user u_a and target item i is computed from the neighbours' ratings of i as follows:

$$\hat{r}_{u_a,i} \mathrel{\hat{=}} \bar{r}_{u_a} + \frac{\sum_{u=1}^{k}(r_{u,i} - \bar{r}_u)\rho_{u_a,u}}{\sum_{u=1}^{k} \rho_{u_a,u}} \tag{1}$$

- Having computed $\hat{r}_{u_a,i}$ for each $i \in T_a$, the system recommends to the active user the k' items from T_a whose predicted ratings are highest. We use $k' = 3$.

3 Group Recommenders

Let $G_a \subseteq U$ be an active group of users, in our case a group who intend going to see a movie together. The goal again is to recommend k' items from a set of T_a items. We will do this by computing a predicted rating $\hat{r}_{G_a,i}$ for active group G_a and each target item $i \in T_a$ and then recommending the k' items in T_a that have the highest predicted ratings.

3.1 Standard Group Recommenders

As we have explained, a common approach to group recommendation, and the one that we follow, is to aggregate the predicted ratings of the members of the group, $\hat{r}_{u_a,i}$ for each $u_a \in G_a$ for the various i in T_a. Possible aggregation functions include *least misery* (where the minimum is taken) and *most pleasure* (where the maximum is taken). We experimented with both before [8], and we found *most pleasure* to give better results, and so we adopt that here:

$$\hat{r}_{G_a,i} \mathrel{\hat{=}} \max_{u_a \in G_a} \hat{r}_{u_a,i} \tag{2}$$

We compute $\hat{r}_{G_a,i}$ for each $i \in T_a$ and recommend the k' with the highest aggregated predicted rating. We will designate this recommender by *Std*.

3.2 Social Group Recommenders

Our previous work showed an improvement in the accuracy of predicted group ratings by taking into account the *personality* of the users in the group and the strength of their connections, which we refer to as their *trust* [10]. We refer to our recommender that takes this extra social information into account as being *social* and the method it uses as being *delegation-based*

We obtain the personality of each user u, denoted $u.pers$, by making group members complete a personality test on registration with the recommender. The details of the personality test are in [15]. In a real application, such as the Facebook social group recommender that we have built [9], trust between users u and v $(u \in U, v \in U, u \neq v)$, $t_{u,v}$, can be based on distance in the social network, the number of friends in common, relationship duration, and so on.

Using the *most pleasure* principle again, we have:

$$\hat{r}_{G_a,i} \doteq \max_{u_a \in G_a} \mathrm{dbr}(\hat{r}_{u_a,i}, G_a) \tag{3}$$

Here the *most pleasure* principle is not applied directly to individual predicted ratings, $\hat{r}_{u_a,i}$. The ratings are modified by the dbr function, which takes into account personality and trust values within the group G_a to compute what we call a delegation-based rating (dbr).

Space limitations preclude a detailed description of the operation of dbr but it is described in [10]. In essence, it is a weighted average of multiple copies of $\hat{r}_{u_a,i}$, one copy for each other member of $u \neq u_a$ in group G_a. The weights are based on the trust between u_a and u, $t_{u_a,u}$, and a value that is computed from the difference in their personalities, $u_a.pers - u.pers$.

The recommender recommends the k' items i from T_a for which $\hat{r}_{G_a,i}$ is highest. We will designate this recommender by *Soc*.

4 Using CBR in Recommenders for Users in Cold-Start

As we have explained, an active user with few ratings is said to be in cold-start. The problem that this causes for the kind of recommenders that we have been discussing is that it becomes difficult to find a reliable neighbourhood of similar users from which predictions can be made. One solution is to copy some ratings into the profile of the active cold-start user from a similar user who has additional ratings. Similarity in this case (i.e. for finding a user from whom ratings can be copied) would be measured using demographic information (age, gender, etc.) [13] because the active user has insufficient ratings to find a similar user using Pearson correlation, ρ. Let v be the user who is similar to u_a and from whom ratings will be copied. Then u_a obtains ratings for all items i that v has rated $(r_{v,i} \neq \perp)$ but that u_a has not $(r_{u_a,i} = \perp)$.

A group recommender can take the same approach when members of the group are in cold-start: prior to predicting individual ratings, it can augment the ratings profiles of group members who are in cold-start with ratings that are copied from the profiles of similar users. But in a group recommender, we can go further than using just demographic information for finding the most similar users from whom ratings will be copied. In our work, we investigate how to reuse ratings from similar users in similar groups in a case-based fashion.

4.1 Case Representation

Assume a case base CB in which each case $c \in CB$ records a previous group movie recommendation event. Each case will have the following structure:

$$\langle id_c, \langle G_c, T_c \rangle, i_c \rangle$$

- id_c is a case identification number, used to distinguish the case from others, but otherwise not used by the CBR.
- The *problem description* part of the case comprises:
  - $G_c \subseteq U$, the group of users who used the recommender previously. For each user $u \in G_c$, we will know demographic information such as u's age ($u.age$) and gender ($u.gender$); u's ratings, $r_{u,i}$ for some set of items; and u's personality value, $u.pers$. And, for each pair of users $u \in G_c, v \in G_c, u \neq v$, we will know the trust value, $t_{u,v}$.
  - $T_c \subseteq I$, the set of items that the users were choosing between. In our case, these were the movies that were at the local multiplex on the occasion when this group used the recommender.
- The *solution* part of the case contains just $i_c \in T_c$, the item that the group agreed on. In our case, this is the movie that the group went to see together.

Cases could also contain some of the numbers calculated when making the recommendation to the group, for example, the predicted individual ratings, $\hat{r}_{u,i}$ for each $u \in G_c$ and for each $i \in T_c$. Or, cases could also contain the *actual* ratings that users assign to item i_c. In other words, having gone to see movie i_c, users may come back to the system and give an actual rating, r_{u,i_c}. We leave the possible exploitation of this additional information to future work.

4.2 CBR for Cold-Start Users in Groups

We will summarize the process by which the case base is used for cold-start users. Details of the similarity measures will be given in subsequent sections. As usual, the goal is to recommend k' items from a set of items, $T_a \subseteq I$, to an active group of users, $G_a \subseteq U$. The recommender will recommend the k' for which the predicted group rating, which is aggregated from the predicted individual ratings, is highest. Of course, if none of the users in G_a is in cold-start, then the system will work either in the fashion described in Section 3.1 or in the fashion described in Section 3.2.

But suppose, on the other hand, that one or more members of G_a are in cold-start. We define this simply using a threshold, θ: a user u_a is in cold-start if and only if the number of items s/he has rated is less than θ, $|\{i : r(u_a, i) \neq \perp\}| < \theta$. In this case, we need to use the CBR. For each user who is in cold-start, we will copy ratings from the *most similar user in the most similar group* in the case base. The details follow.

Case Retrieval. We can write the problem statement as $PS = \langle G_a, T_a \rangle$. We will find the *most similar case*, c^*, in the case base:

$$c^* \hat{=} \arg\max_{c \in CB} \mathrm{sim}(PS, c) \tag{4}$$

The similarity between a problem statement $PS = \langle G_a, T_a \rangle$ and a case $c = \langle id_c, \langle G_c, T_c \rangle, i_c \rangle \in CB$, $\mathrm{sim}(PS, c)$, is calculated on the basis of group similarity:

$$\mathrm{sim}(\langle G_a, T_a \rangle, \langle id_c, \langle G_c, T_c \rangle, i_c \rangle) \hat{=} \mathrm{gsim}(G_a, G_c) \tag{5}$$

This means that in our work case similarity only takes the groups, G_a and G_c, into account; it does not take into account the items, T_a and T_c. T_c contains the items that G_c contemplated in the past, but T_a contains items that G_a is contemplating right now, e.g. movies that have just come to town. These sets may or may not overlap. If they do, we have the basis for a refinement to the similarity we could use in case retrieval. We leave this to future work.

Case Reuse. Next, for each user u_a in G_a who is in cold-start, we find the *most similar user* u^* in case c^* who has rated movies that u_a has not. Let G^* be the group of people described in case c^*. We find:

$$u^* \hat{=} \arg\max_{u \in G^* \wedge \exists i, r_{u_a,i} = \bot \wedge r_{u,i} \neq \bot} \mathrm{psim}_{CB}(u_a, G_a, u, G^*) \tag{6}$$

In the case of more than one such user, we choose the one from whom we can copy the most ratings, i.e. the one who has most ratings for movies that u_a has not rated. Then, temporarily (for the purposes of making u_a's prediction for the items in T_a), we copy into u_a's profile the rating for each item i that u^* has rated ($r_{u^*,i} \neq \bot$) that u_a has not ($r_{u,i} = \bot$).

With each cold-start user's profile augmented in this way, we can then proceed to compute group recommendations in the fashion described in Section 3.1, which we will designate by *Std-CB*, or in the fashion described in Section 3.2, which we will designate by *Soc-CB*. But, it should now be less problematic finding neighbourhoods for the users who are in cold-start because they now have augmented user profiles.

4.3 The Most Similar Group

As we saw above, case retrieval in this system finds the most similar case to the problem statement, which is the one that contains the group that is most similar to G_a. This requires a definition of group similarity, gsim. We compute the similarity of any pair of groups, G and G', from the similarity of the users in the two groups, $\mathrm{psim}_{CB}(u, G, u', G'), u \in G, u' \in G'$. We will define $\mathrm{psim}_{CB}(u, G, u', G')$ in the next subsection.

So, the similarity of G to G' is the average similarity of each user u in G to his/her most similar user in G':

$$\mathrm{gsim}(G, G') \hat{=} \frac{\sum_{u \in G} \mathrm{psim}_{CB}(u, G, u^*, G')}{|G|} \tag{7}$$

where

$$u^* \mathrel{\hat{=}} \arg\max_{u' \in G'} \text{psim}_{CB}(u, G, u', G') \tag{8}$$

Note that the mapping from users $u \in G$ to users $u' \in G'$ is not bijective, meaning we do not prevent two or more people from G being associated with the same user $u' \in G'$. This fact allows us to easily compare groups of different sizes without further complications. It does mean that, if two or more users from G_a are in cold-start, they may all copy ratings from the same user $u' \in G$. (We could have taken the option of requiring bijective mappings, either by only comparing equal-sized groups or by introducing 'virtual' users to make groups equal-sized, and we have done this in on-going work. But it seemed an unnecessary and costly complication in our work on cold-start.)

4.4 The Most Similar User

Our CBR solution to the cold-start problem in group recommenders requires a definition of the similarity between two users, u and u', in different groups, $\text{psim}_{CB}(u, G, u', G')$ where $u \in G$ and $u' \in G'$. This plays two roles in the CBR. First, as Section 4.3 explains, it is used in *case retrieval*, since *the most similar user* is part of the definition of *the most similar group*. Second, as Section 4.2 explains, it is used in *case reuse*, since ratings are copied to each cold-start user from his/her corresponding *most similar user* in the most similar case.

To define $\text{psim}_{CB}(u, G, u', G')$, the similarity between two users in groups, we make use of their ratings, their demographic information (age and gender) and the social information (personality and trust). Specifically, we compute local similarities for each of these, and then combine them into a global similarity.

The local similarities are as follows. For their ratings, we use the Pearson correlation but normalized to $[0, 1]$, denoted here by $\rho_{[0,1]}$. For gender, we use an equality metric and for ages and personalities, we use the range-normalized difference:

$$\text{eq}(x, y) \mathrel{\hat{=}} \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases} \qquad\qquad \text{rn\_diff}_{attr}(x, y) \mathrel{\hat{=}} 1 - \frac{|x - y|}{range_{attr}} \tag{9}$$

For trust values, we compute the average trust value between user u and all other members of his group, $v \in G, u \neq v$, which we will denote by $\bar{t_u}$. Similarly, we compute the average trust value for the other user, $\bar{t_{u'}}$, and we use rn_diff to give the similarity of these two values. We do the same for the standard deviations of the trust values, σ_{t_u} and $\sigma_{t_{u'}}$. The global similarity, psim_{CB}, is simply an average of $\rho_{[0,1]}$, eq_{gender}, rn_diff_{age}, rn_diff_{pers}, $\text{rn\_diff}_{\bar{t}}$ and $\text{rn\_diff}_{\sigma_t}$.

5 Other Recommenders for Users in Cold-Start

An obvious question is whether it makes a difference that our case-based solution to the cold-start problem in group recommenders works on a group basis at all.

Why copy ratings from the most similar user in the most similar group? Why not copy ratings simply from the most similar user in the case base as a whole? Or why not copy ratings from the most similar user known to the system? Systems that work in these different ways will be useful for comparisons in our experiments, hence we define both of these more precisely now.

Consider the set of users who appear in at least one case in the case base:

$$U_{CB} \hat{=} \{u : \exists c = \langle id_c, \langle G_c, T_c \rangle, i_c \rangle \in CB \wedge u \in G_c\} \tag{10}$$

When trying to predict group G_a's rating for an item $i \in T_a$, then for any user $u \in G_a$ who is in cold-start, we could find, and copy ratings from, the most similar user in U_{CB}:

$$u^* \hat{=} \underset{u \in U_{CB} \wedge \exists i, r_{u_a,i} = \perp \wedge r_{u,i} \neq \perp}{\arg\max} \text{psim}_{U_{CB}}(u_a, u) \tag{11}$$

This is different from first finding the most similar case (in other words, the most similar group) and then, for each active user in cold-start, copying ratings from the most similar user in that group. Our case-based approach is conditioned on the groups; this alternative is not. Note that this alternative needs a new definition of the similarity between two people, $\text{psim}_{U_{CB}}$ in place of psim_{CB}. Above, we were able to compute and compare the average and standard deviations of the trust values between a user and all other members of his/her group. In this new setting, this no longer makes sense, since we are ignoring the groups. Hence, the global similarity $\text{psim}_{U_{CB}}$ will be the average of just $\rho_{[0,1]}$, eq_{gender}, rn_diff_{age} and rn_diff_{pers}. We will designate this recommender by *Std-UCB* (where it works in the fashion described in Section 3.1) and by *Soc-UCB* (where it works in the fashion described in Section 3.2).

The second of our two alternative cold-start recommenders ignores the case base altogether. It simply finds, and copies ratings from, the most similar user in U (the entire set of users), wholly ignoring whether they have previously participated in group recommendations or not. Hence,

$$u^* \hat{=} \underset{u \in U \wedge \exists i, r_{u_a,i} = \perp \wedge r_{u,i} \neq \perp}{\arg\max} \text{psim}_{U}(u_a, u) \tag{12}$$

Note that for the experiments in this paper, this requires yet another definition of the similarity between users, psim_U. This is because we only have personality values for users who have participated in group recommendation events. Hence, the global similarity psim_U will be the average of just $\rho_{[0,1]}$, eq_{gender} and rn_diff_{age}. We will designate this recommender by *Std-U* (where it works as per Section 3.1) and by *Soc-U* (where it works as per Section 3.2).

6 Group Recommender Dataset

We need a dataset with which we can evaluate our case-based solution to the cold-start problem in group recommenders. We have built a social group recommender as a Facebook application [9]. But, at the time of writing, it cannot

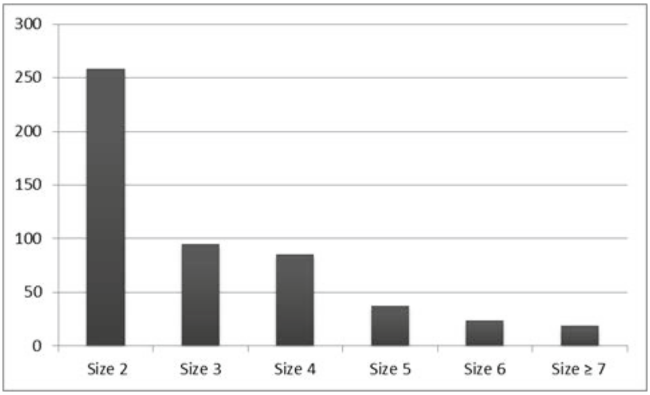


Fig. 1. Group sizes for 525 real movie-going events

provide the volume of data that we need for conducting experiments. Unfortunately, neither are we aware of a public dataset for group recommenders. Hence, we created our own dataset, and we explain how we did this here.

Base Dataset. We have used the MovieLens 1M dataset (`www.grouplens.org`). It gives us around 1 million ratings on a scale of 1 to 5 for around 6040 users for nearly 4000 movies. Each user has at least 20 ratings. The dataset also gives a small amount of demographic information about each user. In particular, we use the user's gender and age range (under 18, $18-24$, $25-34$, and so on).

Groups. We created 100 groups from the MovieLens dataset. Group members are chosen at random from all users in the MovieLens dataset but subject to the following restrictions:

- In a group, users are distinct (but a user may be in more than one group).
- In a group, we ensure that all the users are in the same age range.
- In a group, we ensure that there are at least 15 movies which are co-rated by all members of the group. When we create cases, these 15 movies will be the set T_c. These ratings themselves are withheld from the recommender, because it would not in general know a user's actual ratings for the movies that the group was choosing from.

We conducted a Facebook poll in which we asked respondents to tell us, for the last five times that they went to the cinema in a group, how large the group was. There were 105 respondents and so we learned the group size for 525 events (although we we cannot be certain that they were all distinct events). Figure 1 shows the distribution. We used the frequencies from this distribution to create our 100 groups. Hence, we have 50 groups of size 2, 18 of size 3, 16 of size 4, 7 of size 5, 5 of size 6, and 4 where we took the size to be 7.

Personality Values. We had to impute personality values to the users in the groups. The personality test that we have described in previous work is the

Thomas-Killmann Conflict Mode Instrument (TKI) [15]. Questions on the test reveal the extent to which a person uses each of five modes for dealing with conflict, including "competing", "compromising", "avoiding" and so on. These five modes can be summarized to give scores on two dimensions, "assertiveness" and "cooperativeness", from which we define a single numeric value, $u.pers$, in the range $[0, 1]$, where 0 signals a very cooperative person and 1 signals a very selfish person [10].

To impute personalities to users in our dataset, we make use of the population norms that the TKI Technical Brief provides [14]. We randomly give to each user five scores, one for each mode, based on the distributions given in the Brief. We calculate $u.pers$ from these.

We recognize that this is imperfect. Although the distribution of the five modes among our users will reflect the distribution in the population, the distribution within groups may not reflect reality. Because of the randomness, we might end up with a group of, for example, four very selfish people, where perhaps this rarely occurs in reality. We should be able to take a more informed approach in the future, once our Facebook application generates more data.

Trust Values. As we have discussed, in our Facebook application, trust is computed from Facebook data (distance in the social network, etc.), but that is not available to us for the users in the MovieLens dataset. Rather than simply imputing trust values at random, we have chosen to base them on ratings. For these experiments, the trust between users u and u' is the number of movies on whose ratings they agree as a proportion of the movies that either of them has rated. Agreement here is defined quite loosely: they agree if both have given the movie a rating above the ratings mid-point (which is 3) or if both have given the movie a rating below the ratings mid-point. The formula is as follows:

$$t_{u,u'} \hat{=} \frac{|\{i : (r(u,i) > 3 \land r(u',i) > 3) \lor (r(u,i) < 3 \land r(u',i) < 3)\}|}{|\{i : r(u,i) \neq \bot \lor r(u',i) \neq \bot\}|} \tag{13}$$

Hence, in our dataset, trust is based on the degree of shared taste.

This does not mean that, when psim_{CB} combines $\rho_{[0,1]}$ with $\mathrm{rn\_diff}_{\bar{t}}$ and $\mathrm{rn\_diff}_{\sigma_t}$, it is counting the same shared ratings twice. $\rho_{[0,1]}$ compares ratings between members of different groups (*inter-group*); it aligns a person in one group with someone in the other group who has the same tastes. But $\mathrm{rn\_diff}_{\bar{t}}$ and $\mathrm{rn\_diff}_{\sigma_t}$ compare ratings within groups (*intra-group*) to give trust values, which are then compared between groups; they align a person in one group with someone who has similar trust relationships in the other group.

The Chosen Movie. So far, we have described how we have created 100 groups. As we have explained, we have engineered matters so that, for each group, there is a set of 15 movies that all members of the group have rated (although we withhold the ratings from the recommender), and we are treating these 15 movies as T_c, the set of movies that this group was choosing between. (Remember that T_c can be different for every group.) To create a case, we need to indicate which of these 15 movies the group will actually have chosen to go to see. But we cannot ask random groups of MovieLens users to work out which of their 15 candidate movies they would have gone to see together.

We used four human 'experts' who were given all the information about a group's members G_c and the candidate movies T_c (including the actual ratings by the members of G_c for the items in T_c) and were asked to decide which of the movies the group would be most likely to settle on. Each expert evaluated 50 cases, hence each of the 100 groups was evaluated by two experts (not always the same two experts). Experts were asked to give an ordered list of the three movies from T_c that they thought the members of G_c would agree on.

Since each case is being decided by two experts, we needed a voting scheme to reconcile their judgements. A movie that an expert placed in first position was given three votes; a movie placed in second position was given two votes; and a movie placed in third position was given one vote. By adding up and ranking movies by their votes, we obtain a final ordered list of the movies that G_c would be most likely to see. For example, if both experts placed a movie $i \in T_c$ in first place, then it would receive six votes and would come first in the final combined ordering. But if one expert placed i in first position and $j \neq i$ in second position, but the other expert placed them in the opposite order, then both get five votes. The final ordered set will contain a minimum of three movies (where the experts agreed on the same set of three movies from T_c) and a maximum of six movies (where the two experts disagreed entirely). In fact, the latter never happened; final ordered sets are roughly evenly-split between those of size three and those of size four, plus a handful of size five. We will designate this ordered set by E (for 'Expert') and we will use E_1 to mean movies in the first position in E, E_2 to mean movies in the first and second positions in E, and so on.

7 Evaluation Methodology

The dataset that we have created has 100 movie-going events, in other words 100 cases. We use a leave-one-out cross-validation methodology, where we remove each case in turn from the case base and present it to the recommenders. We compare their recommendations with the judgements of the experts.

We use eight recommenders in these experiments: *Std*, *Soc*, *Std-CB*, *Soc-CB*, *Std-UCB*, *Soc-UCB*, *Std-U* and *Soc-U*. *Soc* (social) indicates that, before aggregation, the recommender uses extra social data to modify individuals' predictions using the *delegation-based* method of Section 3.2, whereas *Std* (standard) indicates that they do not as in Section 3.1. The second part of the name, if there is one, indicates how the recommenders handle cold-start users. The four options here are: they do nothing for cold-start users; they copy ratings from the most similar user in the most similar case (*-CB*, Section 4); they copy ratings from the most similar user from any case (*-UCB*, Section 5); or they copy ratings from the most similar user in the whole dataset (*-U*, also Section 5).

Recall that each recommender recommends the top $k' = 3$ movies from the 15 candidates. Let R be the set of recommendations made by a particular recommender. Then we want to compare R with E from above, the ordered set

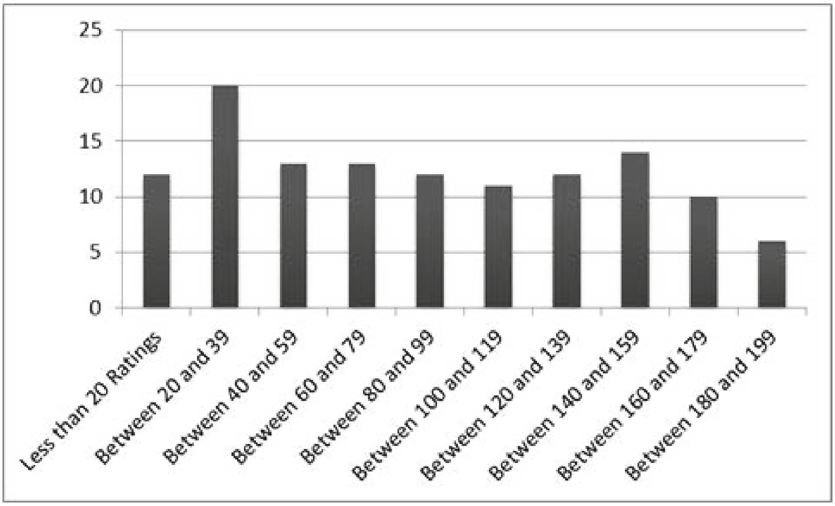


Fig. 2. Number of users in cold-start

of movies that the experts judged to be correct. We computed total *success*@n for $n = 1, 2, 3$, where *success*@$n = 1$ if $\exists i, i \in R \wedge i \in E_n$ and is 0 otherwise. For example, when using *success*@2, we score 1 each time there is at least one recommended movie in the top two positions of E. We also computed total *precision*@n for $n = 1, 2, 3$, where *precision*@$n \mathrel{\hat{=}} |\{i : i \in R \wedge i \in E_n\}|/n$. For example, if no recommended movie is in the top two positions in E, then *precision*@$2 = 0$; if one recommended movie is in the top two positions in E, then *precision*@$2 = 0.5$.

We repeat the experiments with different cold-start thresholds. Figure 2 shows how many users are affected. We see that with $\theta = 20$, just over ten users are in cold-start; with $\theta = 40$, an additional twenty users are in cold-start; and then as θ goes up by 20, the number of users in cold-start grows by about an additional ten each time. (The threshold excludes the 15 ratings for T_a, which are withheld from the recommender.)

8 Results

Figure 3 shows *success*@n for $n = 1, 2, 3$ and *precision*@n for $n = 2, 3$ (*precision*@1 = *success*@1 and is therefore not shown) for cold-start threshold $\theta = 20$.

The first observation about the results is that, as one would expect, as n gets bigger, results improve but differences between systems become less pronounced: with bigger n it is simply easier to make a recommendation that matches an expert judgement. The next observation comes from looking at pairs of bars. The first bar in each pair is a system that does not use social data, and the second is one that does. Consistently throughout all our results, systems that use social data out-perform their counterparts that do not, which shows the value of using

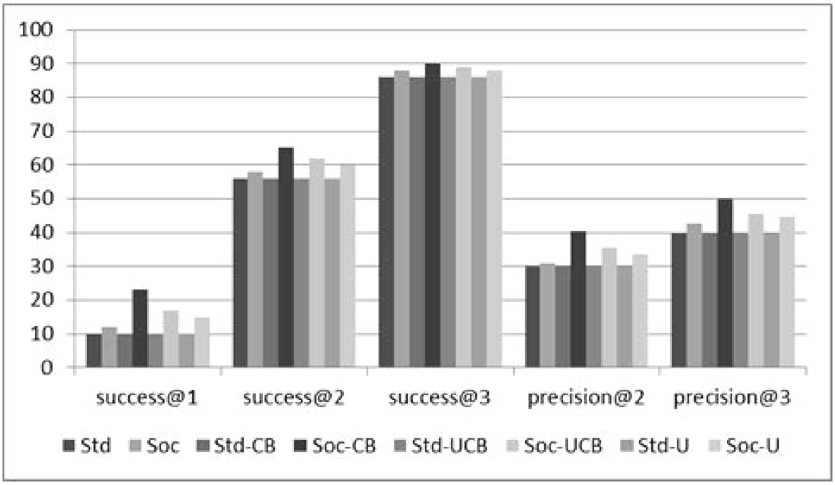


Fig. 3. Results for $\theta = 20$

personality and trust information. This is something we had already established in our previous work (e.g. [10,8]), but it is good to see the result confirmed on our new dataset. A final (and the most important) observation is that the *Soc-CB* system out-performs the *Soc-UCB* system, which out-performs the *Soc-U* system, which out-performs the *Soc* system. In other words, a cold-start strategy that is conditioned on groups (from cases) copies ratings in a more informed and successful way than strategies that copy without regard to groups, and copying ratings is more successful than having no cold-start solution at all.

We tried out a similar cold-start solution in the context of a single-person recommender, where a single active user seeks movie recommendations. If the active user was in cold-start, we copied ratings from a similar user in U. Interestingly, doing so made no or almost no change to the *success*@n and *precision*@n results (not shown here) across several definitions of similarity. We conclude that, for our movie data, conditioning on groups really does seem to be the most effective way to use this cold-start solution.

Figure 4 shows the effects of varying θ from 20 to 200. In other words, more and more users are regarded as being in cold-start and are given ratings from other users. We only show systems that use social data because, as we have already said, they are better. The results for *Soc* itself remain the same for all values of θ because this system has no cold-start strategy. For the other systems, we see that results improve and then fall off as θ increases. For example, for *Soc-CB*, results improve until $\theta = 100$. For this system, 100 is the cut-off point: users with fewer than 100 ratings are ones we should regard as being in cold-start. A higher threshold treats so many users as being in cold-start that the tastes of the active group are swamped by the ratings copied from other users, causing system performance to decrease. The graph is for *precision*@2 but we observed the same pattern of results for all other measures.

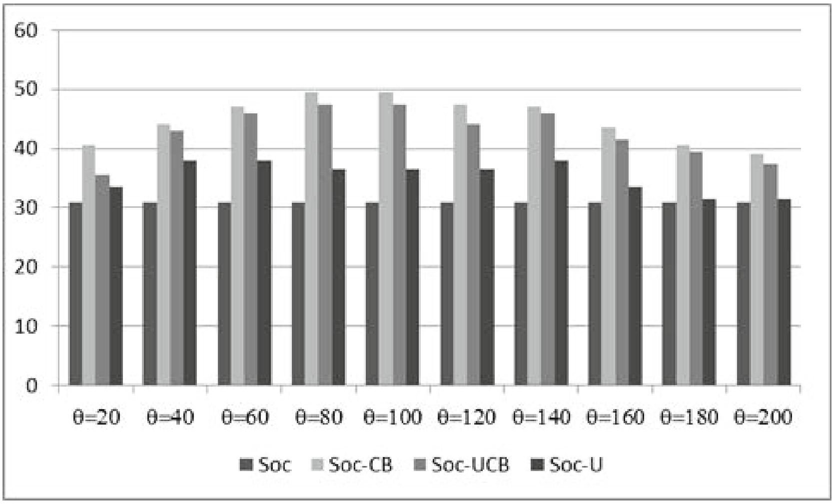


Fig. 4. Results for *precision*@2

9 Conclusions

We have presented a new solution to the cold-start problem in a collaborative group recommender. We use a case base of group recommendation events and copy ratings into the profile of users who are in cold-start from their most similar user in the most similar group in the case base. Our experiments on movie data show that, for users with fewer than 100 ratings, this strategy improves the quality of the group recommendations. The experiments also confirm, using new data, the results of our previous work, viz. a group recommender that uses social data, such as user personality and inter-personal trust, produces higher quality recommendations than one that does not use this data. A side-product of the research has been the construction of a dataset for group recommender research.

There is much that can be done to take this work forward. For us, the next step is to consider a case base in which we more explicitly arrange that there be cases (e.g. movie-going events) that involve groups whose members have a high degree of overlap with the members of the active group, so that we can experiment with the situation where the same group (or nearly the same group) consumes items together on a frequent basis. We also intend to consider richer case representations to take into account such things as timestamps, predicted and actual ratings from group members, and the dynamics of reaching a consensus (e.g. changes in group membership and changes in the selected item). We hope too to gather more data from our Facebook application and use this data as the basis for future experiments.

References

1. Balabanovic, M., Shoham, Y.: Fab: Content-based, collaborative recommendation. Comms. of the Association of Computing Machinery 40(3), 66–72 (1997)
2. Golbandi, N., Koren, Y., Lempel, R.: Adaptive bootstrapping of recommender systems using decision trees. In: Fourth ACM International Conference on Web Search and Data Mining (2011)
3. Herlocker, J.L.: Understanding and Improving Automated Collaborative Filtering Systems. PhD thesis. University of Minnesota (2000)
4. Jameson, A., Smyth, B.: Recommendation to Groups. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 596–627. Springer, Heidelberg (2007)
5. Masthoff, J.: Group modeling: Selecting a sequence of television items to suit a group of viewers. User Modeling and User-Adapted Interaction 14(1), 37–85 (2004)
6. Melville, P., Mooney, R.J., Nagarajan, R.: Content-boosted collaborative filtering for improved recommendations. In: Procs. of the Eighteenth National Conference on Artificial Intelligence (AAAI), pp. 187–192 (2002)
7. Sullivan, D.O., Wilson, D.C., Smyth, B.: Improving Case-Based Recommendation: A Collaborative Filtering Approach. In: Craw, S., Preece, A.D. (eds.) ECCBR 2002. LNCS (LNAI), vol. 2416, pp. 278–291. Springer, Heidelberg (2002)
8. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B.: An architecture for developing group recommender systems enhanced by social elements. International Journal of Human-Computer Studies (in press, 2012)
9. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B., Jiménez-Díaz, G.: Happy movie: A group recommender application in facebook. In: 24th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2011 (2011)
10. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B., Jiménez-Díaz, G.: Social factors in group recommender systems. In: ACM-TIST, TIST-2011-01-0013 (in press, 2011)
11. Rashid, A.M., Albert, I., Cosley, D., Lam, S.K., McNee, S.M., Konstan, J.A., Riedl, J.: Getting to know you: Learning new user preferences in recommender systems. In: Proceedings of the 2002 International Conference on Intelligent User Interfaces, pp. 127–134 (2002)
12. Ricci, F., Arslan, B., Mirzadeh, N., Venturini, A.: ITR: A Case-Based Travel Advisory System. In: Craw, S., Preece, A.D. (eds.) ECCBR 2002. LNCS (LNAI), vol. 2416, pp. 613–641. Springer, Heidelberg (2002)
13. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative Filtering Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 291–324. Springer, Heidelberg (2007)
14. Schaubhut, N.A.: Technical Brief For The Thomas-Kilmann Conflict Mode Instrument. CPP Research Department (2007)
15. Thomas., K.W., Kilmann, R.H.: Thomas-Kilmann Conflict Mode Instrument, Tuxedo, N.Y. (1974)

Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

A Case-Based Solution to the Cold-Start Problem in Group Recommenders*

Lara Quijano-Sánchez
Universidad Complutense de Madrid, Spain
lara.quijano@fdi.ucm.es

Derek Bridge
University College Cork, Ireland
d.bridge@cs.ucc.ie

Belén Díaz-Agudo and **Juan A. Recio-García**
Universidad Complutense de Madrid, Spain
belend@sip.ucm.es, jareciog@fdi.ucm.es

Abstract

In this paper we offer a potential solution to the cold-start problem in group recommender systems. To do so, we use information about previous group recommendation events and copy ratings from a user who played a similar role in some previous group event. We show that copying in this way, i.e. conditioned on groups, is superior to copying nothing and also superior to copying ratings from the most similar user known to the system.

1 Introduction

Groups often dine in restaurants together; visit historic sights, galleries and museums together; attend concerts, the theatre and the cinema together; vacation together; cook and eat together; watch TV together. They must reconcile the different preferences and personalities of the group members when selecting the items they intend to consume together, ranging from restaurants to TV programmes, and may seek the support of a recommender system. While the majority of recommender systems suggest items based on the preferences of an individual consumer, *group recommender systems* suggest items taking into account the preferences and personalities of the members of a group [Jameson and Smyth, 2007].

In this paper, in the context of movie recommendation to groups of friends, we consider a group recommender system that aggregates the results of running a *single-person recommender system* to predict movie ratings for each member of the group. The single-person recommender that we use is a *user-based collaborative recommender system* [Herlocker, 2000], which predicts a user's rating for a candidate movie from the ratings given to that movie by a neighbourhood of users who are similar to that user.

But collaborative recommenders suffer from *cold-start problems* [Herlocker, 2000; Schafer *et al.*, 2007]. In particular, a user-based collaborative recommender finds it difficult to make good predictions for new users, for whom it has few ratings. The group recommender inherits this problem too because it aggregates the predicted ratings from the single-user collaborative recommender. Solutions to the cold-start problem for single-person recommenders are summarized in [Schafer *et al.*, 2007] and include: non-personalized recommendations for cold-start users using population averages; intelligent ways to solicit more ratings (e.g., [Golbandi *et al.*, 2011; Rashid *et al.*, 2002]); and hybrid recommenders that resort to content-based recommendations when there are insufficient ratings to make collaborative recommendations (e.g., [Balabanovic and Shoham, 1997; Melville *et al.*, 2002]).

The contribution of this paper is to introduce and evaluate a *case-based reasoning* (CBR) solution to this problem. We use a case base in which each case records a previous group movie recommendation event. When a group requests a new recommendation but where one or more of the group members is in cold-start, we find a case that describes a previous recommendation event where there are users who are not in cold-start but who play similar roles in their group to the roles the cold-start users play in the active group. We temporarily copy ratings from the users in the case to the corresponding users in the active group and only then proceed to run the single-person recommender and to aggregate its results. It is natural to use a CBR approach because, similar events recur: the same group (perhaps with some small variations) repeats activities together; and some age, gender and personality distributions will tend to recur too (e.g., two adults with two children going to the movies).

CBR has been used in recommender systems before (e.g., [Ricci *et al.*, 2002]) and explicit parallels between CBR and user-based collaborative recommenders have been drawn (e.g., [O'Sullivan *et al.*, 2002]). But we are unaware of any previous use of CBR in group recommenders or in solutions to the cold-start problem.

Section 2 describes our group recommender method; Section 3 describes how we have extended our method to solve the cold-start problem; Section 4 proposes systems against which our system can be compared; Section 5 describes the dataset that we have used in our experiments; Section 6 presents our experimental method; Section 7 contains results; and Section 8 concludes our work.

2 Social Group Recommender Systems

Suppose there are n users, $U = \{u : 1 \ldots n\}$, and m items (e.g., movies), $I = \{i : 1 \ldots m\}$. Let $r_{u,i}$ be the rating that user u assigns to item i. Ratings are on a numeric scale, e.g., 1 = terrible and 5 = excellent, but $r_{u,i} = \perp$ signals that u has

*The paper on which this extended abstract is based was the recipient of the best paper award of the ICCBR 2012 [Quijano-Sánchez et al., 2012].

not yet rated i. Let $G_a \subseteq U$ be an active group of users, in our case a group who intend going to see a movie together. The goal is to recommend to G_a a set of k items drawn from a set of candidate target items $T_a \subseteq I$. For example, T_a could be the set of movies showing this week at the local multiplex.

If none of the users in G_a is in cold-start the group recommender system operates as follows:

Step 1: It uses a user-based collaborative recommender to predict a rating $\hat{r}_{u_a,i}$ for each $u_a \in G_a$ for the various i in T_a. The user-based collaborative recommender that we use works as described in [Herlocker, 2000; Schafer *et al.*, 2007]. In brief, it computes the similarity between u_a and each other user $u \neq u_a$ who has rated i; it retrieves u_a's 20 nearest neighbours, i.e. the 20 users who are most similar to u_a; and its prediction $\hat{r}_{u_a,i}$ is then a weighted average of the neighbours' actual ratings for i.

Step 2: For each i, it aggregates the predicted ratings of each $u_a \in G_a$ yielding a predicted group rating for that item:

$$\hat{r}_{G_a,i} \hat{=} \underset{u_a \in G_a}{\mathrm{F}} \hat{r}_{u_a,i} \tag{1}$$

where F is the aggregation function, which we discuss in more detail below.

Step 3: Finally, it recommends the k items $i \in T_a$ for which the predicted group ratings $\hat{r}_{u_a,i}$ are highest.

Possible aggregation functions F include *least misery* (taking the minimum) and *most pleasure* (taking the maximum) [Masthoff, 2004]. We experimented with both before [Quijano-Sánchez *et al.*, 2011a], and we found *most pleasure* to give better results, and so we adopt that here.

However, our previous work showed an improvement in the accuracy of predicted group ratings by taking into account the *personality* of the users in the group and the strength of their connections, which we refer to as their *trust* [Quijano-Sánchez *et al.*, 2013; 2011a; Quijano-Sánchez *et al.*, 2012]. We refer to our recommender that takes this extra social information into account as being *social* and the method it uses as being *delegation-based*. Specifically then, we have:

$$\hat{r}_{G_a,i} \hat{=} \max_{u_a \in G_a} \mathrm{dbr}(\hat{r}_{u_a,i}, G_a) \tag{2}$$

Here the *most pleasure* principle (maximum) is not applied directly to individual predicted ratings, $\hat{r}_{u_a,i}$. The ratings are modified by the dbr function, which takes into account personality and trust values within the group G_a.

We obtain the personality of each user u by requiring group members complete a personality test on registration with the recommender. The details of the personality test are in [Thomas and Kilmann, 1974]. In a real application, such as the Facebook social group recommender that we have built [Quijano-Sánchez *et al.*, 2011b], trust between two users can be based on distance in the social network, the number of friends in common, relationship duration, and so on. Further details can be found in [Quijano-Sánchez *et al.*, 2013].

We designate by *Soc* the social group recommender system outlined in this section.

3 Using CBR for Cold-Start Users

As explained above, an active user with few ratings is said to be in cold-start. For the recommender systems discussed here, this situation makes it challenging to find a reliable neighbourhood of similar users from which predictions can be made. One solution is to temporarily copy some ratings into the profile of the active cold-start user from a similar user who has additional ratings. Similarity in this case would be measured using demographic information [Schafer *et al.*, 2007] because the active user has insufficient ratings to find a similar user based on co-rated items.

A group recommender can take the same approach when members of the group are in cold-start: prior to predicting individual ratings, it can temporarily augment the ratings profiles of group members who are in cold-start with ratings that are copied from the profiles of similar users. But in a group recommender, we can go further than using just demographic information. In our work, we investigate how to reuse ratings from similar users in similar groups in a case-based fashion.

Assume a case base CB in which each case $c \in CB$ records a previous group movie recommendation event. Each case will have the following structure: $\langle id_c, \langle G_c, T_c \rangle, i_c \rangle$ where id_c is a case identification number. The *problem description* part of the case comprises:

- $G_c \subseteq U$, the group of users who used the recommender previously. For each user $u \in G_c$, we will know u's age and gender; u's ratings, $r_{u,i}$, for some set of items; and u's personality value. For each pair of users $u \in G_c, v \in G_c, u \neq v$, we will know the trust value.
- $T_c \subseteq I$, the set of items that the users were choosing between. In our case, these were the movies that were at the local multiplex on the occasion when this group used the recommender.

And the *solution* part of the case contains just $i_c \in T_c$, the item that the group agreed on. In our case, this is the movie that the group went to see together.

If none of the users in G_a is in cold-start, then the system will work in either of the fashions described in Section 2.

But suppose, on the other hand, that one or more members of G_a are in cold-start. We define this simply using a threshold, θ: a user u_a is in cold-start if and only if the number of items s/he has rated is less than θ. In this case, we need to use the CBR. For each user who is in cold-start, we will copy ratings from the *most similar user in the most similar group* in the case base, as follows.

Case retrieval

We can write the problem statement as $PS = \langle G_a, T_a \rangle$. We will find the *most similar case*, c^*, in the case base:

$$c^* \hat{=} \arg\max_{c \in CB} \mathrm{sim}(PS, c) \tag{3}$$

The similarity between a problem statement $PS = \langle G_a, T_a \rangle$ and a case $c = \langle id_c, \langle G_c, T_c \rangle, i_c \rangle \in CB$, $\mathrm{sim}(PS, c)$, is calculated on the basis of group similarity:

$$\mathrm{sim}(\langle G_a, T_a \rangle, \langle id_c, \langle G_c, T_c \rangle, i_c \rangle) \hat{=} \mathrm{gsim}(G_a, G_c) \tag{4}$$

This means that in our work case similarity takes only the groups, G_a and G_c, into account; it does not take into account the items, T_a and T_c. T_c contains the items that G_c contemplated in the past, but T_a contains items that G_a is contemplating right now, e.g., movies that have just come to town, and these sets need not even overlap.

This process requires a definition of group similarity, gsim. We compute the similarity of any pair of groups, G and G', from the similarity of the users in the two groups, $\mathrm{psim}_{CB}(u, G, u', G'), u \in G, u' \in G'$. Specifically, we define $\mathrm{gsim}(G, G')$ to be the average similarity of each user u in G to his/her most similar user in G'. Note that we do not prevent two or more people from G being associated with the same user $u' \in G'$ (and vice versa). This fact allows us to easily compare groups of different sizes. It does mean that, if two or more users from G_a are in cold-start, they may all copy ratings from the same user $u' \in G$.

We define $\mathrm{psim}_{CB}(u, G, u', G')$, the similarity between two users in groups, as an average similarity over the data that we hold about them: their ratings, gender, ages, personality values and trust values [Quijano-Sánchez *et al.*, 2012].

Case reuse

Next, for each user u_a in G_a who is in cold-start, we find the *most similar user* u^* in case c^* who has rated movies that u_a has not. Let G^* be the group of people described in case c^*:

$$u^* \doteq \operatorname*{arg\,max}_{u \in G^* \wedge \exists i, r_{u_a,i} = \perp \wedge r_{u,i} \neq \perp} \mathrm{psim}_{CB}(u_a, G_a, u, G^*) \quad (5)$$

In the case of more than one such user, we choose the one from whom we can copy the most ratings, i.e. the one who has most ratings for movies that u_a has not rated. Then, temporarily (for the purposes of making u_a's prediction for the items in T_a), we copy into u_a's profile the rating for each item i that u^* has rated ($r_{u^*,i} \neq \perp$) that u_a has not ($r_{u,i} = \perp$).

With each cold-start user's profile augmented in this way, we can then proceed to compute group recommendations in the fashion described in Section 2. But, it should now be less problematic finding neighbourhoods for the users who are in cold-start because they now have augmented user profiles. We designate this system by *Soc-CB*.

4 Other Recommenders for Cold-Start Users

An obvious question is whether it makes a difference that our case-based solution to the cold-start problem in group recommenders works on a group basis at all. Why copy ratings from the most similar user in the most similar group? Why not copy ratings simply from the most similar user in the case base as a whole? Or why not copy ratings from the most similar user known to the system? Systems that work in these different ways will be useful for comparisons in our experiments, hence we define both of these more precisely now.

Consider the set of users who appear in at least one case in the case base:

$$U_{CB} \doteq \{u : \exists c = \langle id_c, \langle G_c, T_c \rangle, i_c \rangle \in CB \wedge u \in G_c\} \quad (6)$$

When trying to predict group G_a's rating for an item $i \in T_a$, then for any user $u \in G_a$ who is in cold-start, we could find, and copy ratings from, the most similar user in U_{CB}:

$$u^* \doteq \operatorname*{arg\,max}_{u \in U_{CB} \wedge \exists i, r_{u_a,i} = \perp \wedge r_{u,i} \neq \perp} \mathrm{psim}_{U_{CB}}(u_a, u) \quad (7)$$

This is different from first finding the most similar group and then, for each active user in cold-start, copying ratings from the most similar user in that group. Our case-based approach is conditioned on the groups; this alternative is not. We designate this recommender by *Soc-UCB*.

The second of our two alternative cold-start recommenders ignores the case base altogether. It finds, and copies ratings from, the most similar user in U (the entire set of users), wholly ignoring whether they have previously participated in group recommendations or not. Hence,

$$u^* \doteq \operatorname*{arg\,max}_{u \in U \wedge \exists i, r_{u_a,i} = \perp \wedge r_{u,i} \neq \perp} \mathrm{psim}_{U}(u_a, u) \quad (8)$$

We designate this recommender by *Soc-U*.

5 Group Recommender Dataset

We need a dataset on which to evaluate our case-based solution to the cold-start problem in group recommenders. We are not aware of a public dataset for group recommenders, hence we created our own. We started from the MovieLens 1M dataset (`www.grouplens.org`). For each user, it records gender, age range, and at least twenty ratings. We impute a personality value to each used based on the population norms in [Schaubhut, 2007].

We created 100 groups, randomly choosing group members from all users in the MovieLens dataset subject to the following restrictions: in a group, users are distinct (but a user may be in more than one group); all users are in the same age range; and we ensure that there are at least 15 movies which are co-rated by all members of the group. When we create cases, these 15 movies will be the set T_c. Their ratings are withheld from the recommender, because it would not in general know a user's actual ratings for the candidate movies.

We conducted a Facebook poll in which we asked respondents to tell us, for the last five times that they went to the cinema in a group, how large the group was. We used the frequencies to create our groups.

As we have discussed, in our Facebook application, trust is computed from Facebook data (distance in the social network, etc.), but that information is not available to us for the users in the MovieLens dataset. Rather than simply imputing trust values at random, we chose to base them on the degree of shared taste as revealed by co-rated items.

To create a case, we need to indicate which of the 15 movies in T_c the group will actually have chosen. But we cannot ask random groups of MovieLens users to work out which of their 15 candidate movies they would have gone to see together. We used four human 'experts' who were given all the information about a group's members G_c and the candidate movies T_c (including the actual ratings by the members of G_c for the items in T_c) and were asked to decide which of the movies the group would be most likely to settle on. Each expert evaluated 50 cases, hence each of the 100 groups was evaluated by two experts (not always the same two). Experts were asked to give an ordered list of the three movies from

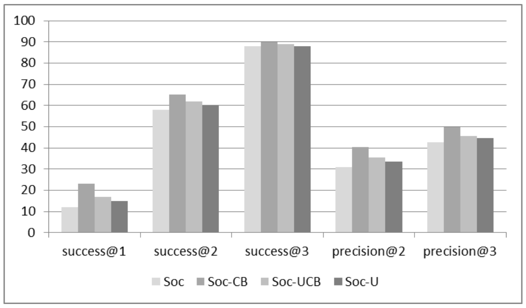


Figure 1: Results for $\theta = 20$.

T_c that they thought the members of G_c would agree on. We combined the experts' judgements into a single final ordered list, denoted by E (for "Expert"). Let E_i be the movie in positions i in E.

6 Evaluation Methodology

The dataset we created has 100 movie-going events. We use a leave-one-out cross-validation methodology, where we remove each case in turn from the case base and present it to the recommenders. We compare their recommendations with the experts' judgements. We report results from four recommenders: *Soc*, *Soc-CB*, *Soc-UCB*, and *Soc-U*. Each recommender recommends the top $k = 3$ movies from the 15 candidates. Let R be the set of recommendations made by a particular recommender. We compare R with E from above. We computed total $success@n$ for $n = 1, 2, 3$, where $success@n$ = 1 if $\exists i, i \in R \wedge i \in E_n$ and is 0 otherwise. For example, when using $success@2$, we score 1 each time there is at least one recommended movie in the top two positions of E. We also computed total $precision@n$ for $n = 1, 2, 3$, where $precision@n \hat{=} |\{i : i \in R \wedge i \in E_n\}|/n$. For example, if no recommended movie is in the top two positions in E, then $precision@2 = 0$; if one recommended movie is in the top two positions in E, then $precision@2 = 0.5$.

We repeat the experiments with different cold-start thresholds (θ). For $\theta = 20$, just over ten users are in cold-start; with $\theta = 40$, an additional twenty users are in cold-start; and then as θ goes up by 20, the number of users in cold-start grows by about an additional ten each time. (The threshold excludes the 15 ratings for T_a withheld from the recommender.)

7 Results

Figure 1 shows $success@n$ for $n = 1, 2, 3$ and $precision@n$ for $n = 2, 3$ ($precision@1 = success@1$ and is therefore not shown) for cold-start threshold $\theta = 20$.

Results show that as n gets bigger, results improve but differences between systems become less pronounced: with bigger n it is simply easier to make a recommendation that matches an expert judgement. Most importantly, we see that *Soc-CB* system out-performs the *Soc-UCB* system, which out-performs the *Soc-U* system, which out-performs the *Soc* system. So, a cold-start strategy that is conditioned on groups copies ratings in a more informed and successful way than strategies that copy without regard to groups, and copying ratings is more successful than having no cold-start solution.

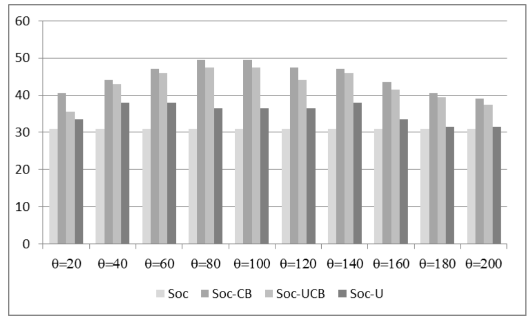


Figure 2: Results for $precision@2$.

We tried out a similar cold-start solution in the context of a single-person recommender, where a single active user seeks movie recommendations. For an active user in cold-start, we copied ratings from a similar user in U. Interestingly, doing so made no or almost no change to the $success@n$ and $precision@n$ results (not shown) across several definitions of similarity. We conclude that, for our movie data, the cold-start solution is most effective on conditioning on groups.

We also studied the impact of varying θ in [20,200], Figure 2. In other words, more and more users are regarded as being in cold-start. The results for *Soc* itself remain the same for all values of θ because this system has no cold-start strategy. For the other systems, we see that results improve and then fall off as θ increases. For example, for *Soc-CB*, results improve until $\theta = 100$. For this system, 100 is the cut-off point: users with fewer than 100 ratings are the ones we should regard as being in cold-start. A higher threshold treats so many users as being in cold-start that the tastes of the active group are swamped by the ratings copied from other users, causing system performance to decrease. The graph is for $precision@2$ but we observed the same pattern of results for all other measures.

8 Conclusions

We introduced a new solution to the cold-start problem in a collaborative group recommender. We use a case base of group recommendation events and copy ratings into the profile of users in cold-start from their most similar user in the most similar group in the case base. Experiments on movie data show that, for users with fewer than 100 ratings, our strategy improves the quality of the group recommendations.

Much can be done to take this work forward. Our next step is to consider a case base in which we more explicitly arrange that there be cases (e.g., movie-going events) involving groups whose members have a high degree of overlap with the members of the active group. We can then experiment with situations where the same group (or nearly the same group) consumes items together on a frequent basis. We also plan to enrich case representations by including timestamps, predicted and actual ratings from group members, and the dynamics of reaching a consensus (e.g., changes in group membership and changes in the selected item). We will gather more data from our Facebook application and use it to overcome the limitations of our current dataset.

References

[Balabanovic and Shoham, 1997] M. Balabanovic and Y. Shoham. Fab: Content-based, collaborative recommendation. *Comms. of the Association of Computing Machinery*, 40(3):66–72, 1997.

[Golbandi *et al.*, 2011] N. Golbandi, Y. Koren, and R. Lempel. Adaptive bootstrapping of recommender systems using decision trees. In *Fourth ACM International Conference on Web Search and Data Mining*, 2011.

[Herlocker, 2000] J. L. Herlocker. *Understanding and Improving Automated Collaborative Filtering Systems*. PhD thesis, University of Minnesota, 2000.

[Jameson and Smyth, 2007] A. Jameson and B. Smyth. Recommendation to groups. In *The Adaptive Web, Methods and Strategies of Web Personalization*, Lecture Notes in Computer Science, 4321, pages 596–627. Springer, 2007.

[Masthoff, 2004] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, 14(1):37–85, 2004.

[Melville *et al.*, 2002] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Procs. of the Eighteenth National Conference on Artificial Intelligence (AAAI)*, pages 187–192, 2002.

[O'Sullivan *et al.*, 2002] D. O'Sullivan, D. C. Wilson, and B. Smyth. Improving case-based recommendation: A collaborative filtering approach. In S. Craw and A. Preece, editors, *Procs. of the 6th European Conference on Case-Based Reasoning*, pages 278–291. Springer, 2002.

[Quijano-Sánchez *et al.*, 2011a] Lara Quijano-Sánchez, Juan A. Recio-García, and Belén Díaz-Agudo. Group recommendation methods for social network environments. *3rd Workshop on Recommender Systems and the Social Web*, 5th ACM International Conference on Recommender Systems, RecSys'11, 2011.

[Quijano-Sánchez *et al.*, 2011b] Lara Quijano-Sánchez, Juan A. Recio-García, Belén Díaz-Agudo, and Guillermo Jiménez-Díaz. Happy movie: A group recommender application in facebook. In *FLAIRS Conference*, 2011.

[Quijano-Sánchez *et al.*, 2012] Lara Quijano-Sánchez, Derek Bridge, Belén Díaz-Agudo, and Juan A. Recio-García. A case-based solution to the cold-start problem in group recommenders. In Belén Díaz-Agudo and Ian Watson, editors, *Case-Based Reasoning Research and Development (Procs. of the 20th International Conference on Case-Based Reasoning)*, LNAI 7466, pages 342–356. Springer, 2012.

[Quijano-Sánchez *et al.*, 2013] Lara Quijano-Sánchez, Juan A. Recio-García, Belén Díaz-Agudo, and Guillermo Jiménez-Díaz. Social factors in group recommender systems. *ACM Transactions on Intelligent Systems and Technology*, 4(1):Article 8, January 2013.

[Rashid *et al.*, 2002] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl. Getting to know you: Learning new user preferences in recommender systems. In *Proceedings of the 2002 International Conference on Intelligent User Interfaces*, pages 127–134, 2002.

[Ricci *et al.*, 2002] F. Ricci, B. Arslan, N. Mirzadeh, and A. Venturini. ITR: A case-based travel advisory system. In S. Craw and A. Preece, editors, *Procs. of the 6th European Conference on Case-Based Reasoning*, pages 613–641. Springer, 2002.

[Schafer *et al.*, 2007] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web, Methods and Strategies of Web Personalization*, Lecture Notes in Computer Science 4321, pages 291–324. Springer, 2007.

[Schaubhut, 2007] N. A. Schaubhut. *Technical Brief for the Thomas-Kilmann Conflict Mode Instrument*. CPP Research Department, 2007.

[Thomas and Kilmann, 1974] K. W. Thomas and R. H. Kilmann. *Thomas-Kilmann Conflict Mode Instrument*. Tuxedo, N.Y., 1974.

Chapter 17

Case-Based Aggregation of Preferences for Group Recommenders

17.1 Citation

Quijano-Sánchez, L., Bridge, D., Díaz-Agudo, B., Recio-García, J. A., Case-Based Aggregation of Preferences for Group Recommenders, in: Procs. of the 20th International Conference on Case Based Reasoning, ICCBR'12, Lyon, France, pages: 327-341, 2012. ISBN 978-3-642-32985-2. ICCBR is ranked as CORE C in the CORE ranking.

17.2 Contributions covered by this paper

In this paper we have studied a new strategy for social group recommendation based on the usage of past recommendations in a CBR. In it we retrieve previous cases of similar groups by analysing groups social components besides the demographic and preferences information. Later, we reproduce users' behaviour in the most similar groups.

Case-Based Aggregation of Preferences for Group Recommenders

Lara Quijano-Sánchez[1], Derek Bridge[2], Belén Díaz-Agudo[1], and Juan A. Recio-García[1]

[1] Department of Software Engineering and Artificial Intelligence, Universidad Complutense de Madrid, Spain
[2] Department of Computer Science, University College Cork, Ireland
{lara.quijano,jareciog}@fdi.ucm.es, d.bridge@cs.ucc.ie, belend@sip.ucm.es

Abstract. We extend a group recommender system with a case base of previous group recommendation events. We show that this offers a new way of aggregating the predicted ratings of the group members. Using user-user similarity, we align individuals from the active group with individuals from the groups in the cases. Then, using item-item similarity, we transfer the preferences of the groups in the cases over to the group that is seeking a recommendation. The advantage of a case-based approach to preference aggregation is that it does not require us to commit to a model of social behaviour, expressed in a set of formulae, that may not be valid across all groups. Rather, the CBR system's aggregation of the predicted ratings will be a lazy and local generalization of the behaviours captured by the neighbouring cases in the case base.

1 Introduction

Groups often holiday together; tour museums and art galleries together; visit historic sights together; attend concerts and other events together; dine in restaurants together; watch movies and TV programmes together; listen to music together; cook and eat together. They must select the items which they intend to consume together, ranging from holiday destinations to recipes, in a way that reconciles the different preferences and personalities of the group members. For this, they may seek the support of a recommender system. But where the majority of recommender systems suggest items based on the preferences of an individual consumer, *group recommender systems* suggest items taking into account the preferences and personalities of the members of a group [4].

Commonly, group recommender systems aggregate predicted ratings for group members [4]: for each group member, a single-person recommender system predicts a set of ratings for the candidate items; then, the group recommender aggregates the ratings. The new group recommender system that we present in this paper takes the same approach, i.e. it aggregates the preferences of the group members, but it uses Case-Based Reasoning (CBR) for the aggregation. Figure 1 is suggestive of its operation. The system has a case base of past group

B. Díaz Agudo and I. Watson (Eds.): ICCBR 2012, LNCS 7466, pp. 327–341, 2012.

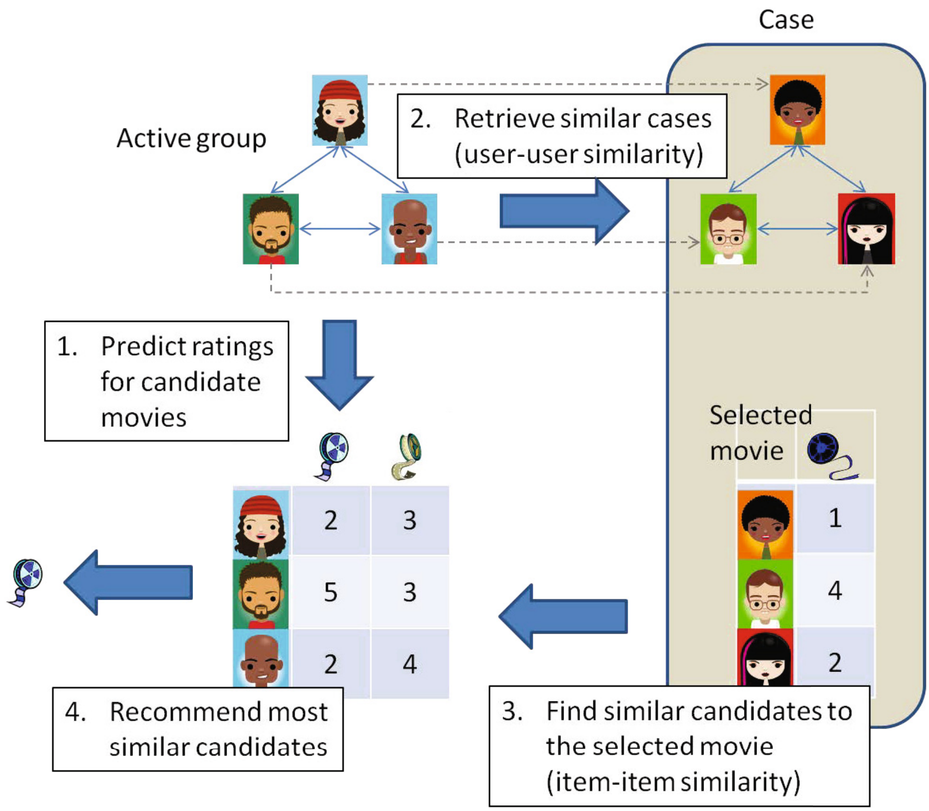


Fig. 1. Overview of the case-based recommender

recommendation events. Each case (right-hand side in the diagram) records the members of the group; the candidate items; the item that the group chose to consume together, which we will call the *selected item*; and the ratings that each group member gave to the selected item after consuming it. To make a recommendation to a new active group (top-left in the diagram), the CBR system deploys a unique combination of user-user and item-item similarity, as follows:

Step 1: First, it uses a user-based collaborative recommender to predict a rating for each candidate item by each group member.

Step 2: Next, it retrieves cases, i.e. past group recommendation events, that involve groups that are similar to the active group. Case retrieval uses the user-user similarity measure, and, as a by-product, it aligns each member of the active group with a member of the group in the case (the dashed lines in Figure 1). The similarity measure compares group members on their age, gender, personality and ratings and the degrees of trust between members of each group (the solid lines between group members in the diagram).

Step 3: Then, it reuses each case that is retrieved: the contributions that each group member made in choosing the selected item are transferred to the corresponding member of the active group. This is done by scoring the new candidate items by their item-item similarity to the selected item. In this way, the retrieved cases act as implicit models of group decision-making, which are transferred to the decision-making in the active group.

Step 4: Finally, it recommends the candidate items that have obtained the highest scores.

The paper explains this more fully. Section 2 gives some background exposition that we need for later sections; Section 3 describes an existing group recommender system, which we will use for comparison purposes; Section 4 describes the new case-based group recommender; Section 5 describes an experiment that compares the new recommender with the one we developed previously; and Section 6 concludes and presents some ideas for future work.

2 User-User and Item-Item Similarity

Suppose there are n users, $U = \{u : 1 \dots n\}$, and m items (e.g. movies), $I = \{i : 1 \dots m\}$. Let r be a ratings matrix and $r_{u,i}$ be the rating that user u assigns to item i. Ratings are on a numeric scale, e.g. 1 = terrible and 5 = excellent, but $r_{u,i} = \perp$ signals that u has not yet rated i.

The similarity between one user and another, $u \in U, u' \in U, u \neq u'$, can be computed using Pearson Correlation [3], ρ. In effect, this computes the similarity between two *rows* in a ratings matrix like the one in the table in the lower left-hand part of Figure 1. The *user-user similarity* is:

$$\rho_{u,u'} \hat{=} \frac{\sum_{i \in I \wedge r_{u,i} \neq \perp \wedge r_{u',i} \neq \perp} (r_{u,i} - \bar{r}_u)(r_{u',i} - \bar{r}_{u'})}{\sqrt{\sum_{i \in I \wedge r_{u,i} \neq \perp \wedge r_{u',i} \neq \perp} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I \wedge r_{u,i} \neq \perp \wedge r_{u',i} \neq \perp} (r_{u',i} - \bar{r}_{u'})^2}} \tag{1}$$

$\bar{r}$ denotes a mean value and σ denotes a standard deviation, and these are computed over the *co-rated items* only ($i \in I \wedge r_{u,i} \neq \perp \wedge r_{u',i} \neq \perp$).

Suppose we want to recommend to active user u_a one or more of a set of candidate items $T_a \subseteq I$. For example, T_a could be the set of movies showing this week at u_a's local multiplex. Using user-user similarity, $\rho_{u,u'}$, we can build a *user-based collaborative recommender* [3,13]. For each $i \in T_a$, it will predict active user u_a's rating for i, $\hat{r}_{u_a,i}$. It can do this using nearest-neighbour methods: from the users for whom $\rho_{u_a,u'}$ is greater than zero, it finds the k users $u' \in U$ who have rated i and who are most similar to u_a. The predicted rating is a weighted average of the neighbours' ratings for i [12]. The recommender suggests to the user the k' items $i \in T_a$ for which the predicted ratings $\hat{r}_{u_a,i}$ are highest.

But, given a ratings matrix we can equally well compute the similarity between one item and another, $i \in I, i' \in I, i \neq i'$, the *item-item similarity*, again using

Pearson correlation. In effect, this computes the similarity between two *columns* in a ratings matrix such as the one in the lower-left of Figure 1:

$$\rho_{i,i'} \hat{=} \frac{\sum_{u \in U \wedge r_{u,i} \neq \perp \wedge r_{u,i'} \neq \perp} (r_{u,i} - \bar{r}_i)(r_{u,i'} - \bar{r}_{i'})}{\sqrt{\sum_{u \in U \wedge r_{u,i} \neq \perp \wedge r_{u,i'} \neq \perp} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U \wedge r_{u,i} \neq \perp \wedge r_{u,i'} \neq \perp} (r_{u,i'} - \bar{r}_{i'})^2}} \quad (2)$$

In this case, the means ($\bar{r}$) and standard deviations (σ) are computed over the *users who have rated both items* ($u \in U \wedge r_{u,i} \neq \perp \wedge r_{u,i'} \neq \perp$).

Using item-item similarity, $\rho_{i,i'}$, it is possible to build an *item-based collaborative recommender* [6,13], although we use it for a different purpose in this paper. Before presenting the case-based group recommender in detail, we present the group recommender system against whose performance we will be comparing the new recommender.

3 Social Recommendations to Groups

For the comparison, we use a group recommender that we developed previously [11,8]. With real data and, in more recent work, with a larger dataset of artificial data, we showed that, relative to simpler approaches, our group recommender improves the accuracy of predicted group ratings and the precision of group recommendations, and that is why we use it here.

Let $G_a \subseteq U$ be an active group of users, in our case a group which intends to see a movie together. The goal is to recommend k' items from a set of T_a items. As Section 1 has mentioned, the system works by aggregation of ratings, as follows:

- For each $i \in T_a$ taken in turn, the recommender does the following:
  - It predicts a rating for item i, $\hat{r}_{u_a,i}$, for each individual group member $u_a \in G_a$. It does this using the user-based collaborative technique that we described in Section 2, i.e. it averages the ratings of i given by u_a's k most similar neighbours who have rated i.
  - It applies a function, designated dbr (which stands for *delegation-based rating*), to each predicted rating. The dbr function modifies $\hat{r}_{u_a,i}$ to take into account the *personality* of the user and the strength of connections between this person and other members of the group, which we refer to as their *trust*. In this way, not all the predicted individual ratings will contribute equally in the aggregation. We explain it in detail below.
  - It aggregates the individual predicted ratings into a single group rating $\hat{r}_{G_a,i}$. Possible aggregation functions include *least misery* (where the minimum is taken), and *most pleasure* (where the maximum is taken) [7]. We experimented with both before [9], and we found *most pleasure* to give better results, and so we adopt that here:

$$\hat{r}_{G_a,i} \hat{=} \max_{u \in G_a} \mathrm{dbr}(\hat{r}_{u,i}, G_a) \quad (3)$$

- It recommends the k' items in $i \in T_a$ for which the predicted group ratings $\hat{r}_{G_a,i}$ are highest.

The delegation-based method recognizes that a person's opinions may be based in part on the opinions of other members of the group. The formula, which we explain below, is as follows:

$$\mathrm{dbr}(\hat{r}_{u,i}, G_a) \mathrel{\hat{=}} \frac{\sum_{v \in G_a \wedge v \neq u} t_{u,v} \times (r_{v,i} + \theta_{r_{v,i}} \times (v.pers - u.per))}{\sum_{v \in G_a \wedge v \neq u} t_{u,v}} \qquad (4)$$

In Equation 4, $t_{u,v}$ denotes the trust between u and v, which is a real number between 0.0 (no connection) and 1.0 (strong connection). In a real application, such as the Facebook movie group recommender that we have built [10], $t_{u,v}$ can be based on distance in the social network, the number of friends in common, relationship duration, and so on. As you can see, for user u in group G_a, we take into account the predicted ratings, $r_{v,i}$, for each other member of the group, $v \in G_a, v \neq u$, weighted by the trust between the two users, $t_{u,v}$. This follows [2], where a method for group recommendations using trust is proposed.

In Equation 4, $u.pers$ denotes user u's personality, also a real number between 0.0 (very cooperative) and 1.0 (very selfish). In our Facebook group movie recommender, users complete a personality test on registration. The details of the test are in [15]. In Equation 4, the rating given by another group member $r_{v,i}$ is increased or decreased depending on the difference in personality, $v.pers - u.pers$. This way, users with stronger personalities will contribute more to the final score. A user v with a positive opinion of i, i.e. where $r_{v,i}$ is greater than the mid-point of the ratings scale, will want to increase u's opinion of i; but if v has a negative opinion, i.e. where $r_{v,i}$ is less than the mid-point of the scale, then v will want to decrease u's opinion. We model this through a function θ:

$$\theta_{r_{v,i}} \mathrel{\hat{=}} \begin{cases} 5 & \text{if } r_{v,i} \geq mid \\ -5 & \text{otherwise} \end{cases}$$

where *mid* is the mid-point of the ratings scale, e.g. 3 on a five-point scale. We chose the constants (5 and -5) because the mean difference in personality values is 0.2 and therefore the impact of $\theta_{r_{v,i}}$ in Equation 4 will typically be 1 or -1.

4 A Case-Based Group Recommender System

Our new group recommender takes a case-based reasoning approach. There are two motivations for a case-based approach to group recommender systems.

- Firstly, groups tend to recur: the same group (with few variations) repeats activities together. Furthermore, group structures tend to recur: in the case of movies, for example, family outings comprising two adults and two children are common, as are parties of friends in the same age range.
- Secondly, group recommenders such as the one described in Section 3, have a 'one-size-fits-all' approach to the way they combine the predicted individual ratings. This ignores the possibility that different groups might have very different dynamics, not captured by a single theory expressed in a set of

formulae that apply globally. A case-based approach does not require us to commit to a model of social behaviour and to find a way to express that model in a set of formulae. Rather, aggregation of predicted ratings will be a lazy and local generalization (in the spirit of CBR) of the behaviours captured by the neighbouring cases in the case base.

4.1 Case Representation

Assume a case base CB in which each case $c \in CB$ records a previous group recommendation event. Each case will have the following structure:

$$\langle id_c, \langle G_c, T_c \rangle, i_c, \{r_{u,i_c} : u \in G_c\} \rangle$$

- id_c is a case identification number, used to distinguish the case from others.
- The *problem description* part of the case comprises:
  - $G_c \subseteq U$, the group of users who used the recommender previously. For each user $u \in G_c$, we will know demographic information such as u's age ($u.age$) and gender ($u.gender$); u's ratings, $r_{u,i}$ for some set of items; and u's personality value, $u.pers$. And, for each pair of users $u \in G_c, v \in G_c, u \neq v$, we will know the trust value, $t_{u,v}$.
  - $T_c \subseteq I$, the set of items that the users were choosing between. In our cases, these were the movies that were at the local multiplex on the occasion when this group used the recommender.
- The *solution* part of the case contains just $i_c \in T_c$, the selected item, i.e. the item that the group agreed on. In our cases, this is the movie that the group went to see together.
- The *outcome* part of the case [1,5] is a set of ratings. These are the actual ratings r_{u,i_c} that the members of the group $u \in G_c$ gave to item i_c: for example, after a group has gone to see their selected movie, group members return and rate the movie. In practice, some members of the group will not do this. In these cases, we can use $\hat{r}_{u,i_c}$ instead, i.e. the rating that a user-based collaborative recommender (Section 2) predicts the user $u \in G_c$ will assign to i_c. However, we have not so far evaluated empirically the consequences of using predicted ratings in place of actual ratings.

We now explain how this recommender makes its recommendations.

Step 1: Predict Individual Ratings

As usual, the goal is to recommend k' items from a set of items, $T_a \subseteq I$, to an active group of users, $G_a \subseteq U$. We can write the problem statement as $PS = \langle G_a, T_a \rangle$. The first step is to predict individual ratings $\hat{r}_{u,i}$ for each candidate item $i \in T_a$ for each member of the active group $u \in G_a$. We do this using a standard user-based collaborative recommender, as described in Section 2.

Later in the process, it may be necessary to insert virtual users into G_a, i.e. ones that are not real people. We explain when and why this happens at the appropriate time. But it simplifies the later exposition if we say now how we predict the ratings of items by virtual users. Since virtual users have no actual ratings, we cannot use the user-based collaborative recommender, as we do for real users. Instead, if u is a virtual user, its predicted rating for item i, $\hat{r}_{u,i}$, is the population average rating for i: $\frac{\sum_{u \in U \wedge r_{u,i} \neq \bot} r_{u,i}}{|u \in U \wedge r_{u,i} \neq \bot|}$.

Step 2: Retrieve Cases

The next step is to find the k'' *most similar cases*. We use $k'' = 3$. The similarity between a problem statement $PS = \langle G_a, T_a \rangle$ and a case $c = \langle id_c, \langle G_c, T_c \rangle, i_c, \{r_{u,i_c} : u \in G_c\} \rangle \in CB$, $\text{sim}(PS, c)$, is calculated on the basis of group similarity:

$$\text{sim}(\langle G_a, T_a \rangle, \langle id_c, \langle G_c, T_c \rangle, i_c, \{r_{u,i_c} : u \in G_c\} \rangle) \hat{=} \text{gsim}_{cbr}(G_a, G_c) \qquad (5)$$

This means that in our work case similarity only takes the groups, G_a and G_c, into account; it does not take into account the items, T_a and T_c. T_c contains the items that G_c contemplated in the past, but T_a contains items that G_a are contemplating right now, e.g. movies that have just come to town. These sets may or may not overlap. If they do, we have the basis for a refinement to the similarity we could use in case retrieval. We leave this to future work.

We denote the group similarity by gsim_{cbr}, and we emphasize that this is a new definition, richer than definitions that we have used in other work [8]. In effect, it is a form of graph similarity: users are nodes; trust relationships are weighted edges.

In our definition of group similarity, we pair each user from the active group G_a with exactly one user from the group in the case G_c and vice versa. In other words, we will be finding a *bijection* from G_a to G_c. This raises a problem when comparing groups of different sizes, where a bijection is not possible. In such situations, we could simply say that $\text{gsim}_{cbr}(G_a, G_c) = 0$. However, we did not want to do this. It might force the system to retrieve unsuitable cases. Consider a case base that just happens to contain many families of four (two adults, two children), no families of five, but many parties of five friends. If the active group is a family of five (two adults, three children), it is surely not appropriate to prevent retrieval of families of four and only retrieve parties of five friends.

To enable comparisons, this is the point, prior to computing similarity, that we insert additional virtual users into either G_a or G_c, whichever is the smaller, in order to make the groups the same size.

Now, we can define the group similarity measure. Consider any pair of equal-sized groups, G and G' and a bijection, f, from G to G'. The function f will map members of G to G', and so for any $u \in G$, we can compute the similarity, psim_{cbr}, to his/her partner $f(u) \in G'$. We will do this for each user and his/her partner, and take the average:

$$\text{gpsim}_{cbr}(G, G', f) \hat{=} \frac{\sum_{u \in G} \text{psim}_{cbr}(u, f(u))}{|G|} \qquad (6)$$

But, we also have trust values for each pair of users in G, and we can compute the similarities between each of these and the trust values for the corresponding pair of users in G'. Again we take the average (dividing by the number of pairs):

$$\text{gtsim}_{cbr}(G, G', f) \hat{=} \frac{\sum_{u \in G, v \in G, u \neq v} \text{tsim}_{cbr}(t_{u,v}, t_{f(u),f(v)})}{|G|^2 - |G|} \tag{7}$$

We combine gpsim_{cbr} and gtsim_{cbr} in a weighted average to obtain the following definition of the similarity between any pair of equal-sized groups, G and G', given a bijection f from G to G':

$$\text{gsim}_{cbr}(G, G', f) \hat{=} \alpha \times \text{gpsim}_{cbr}(G, G', f) + (1 - \alpha) \times \text{gtsim}_{cbr}(G, G', f) \tag{8}$$

We currently use $\alpha = 0.5$.

This definition of gsim_{cbr} (Equation 8) uses gtsim_{cbr} (Equation 7), which uses tsim_{cbr}, the similarity between two trust values, which we have not yet defined. We use their range-normalized difference:

$$\text{tsim}_{cbr}(x, y) \hat{=} \text{rn\_diff}_t(x, y) \tag{9}$$

where

$$\text{rn\_diff}_{attr}(x, y) \hat{=} 1 - \frac{|x - y|}{range_{attr}} \tag{10}$$

There is a problem, however. If one or both of u or v (Equation 7) is a virtual user, we will not have a trust value; similarly, if one or both of $f(u)$ or $f(v)$ is virtual. In these situations, we impute an average trust value between that pair of users, which empirically we found to be 0.05.

Equally, the definition of gsim_{cbr} (Equation 8) uses gpsim_{cbr} (Equation 6), which uses psim_{cbr}, the similarity between a person u in one group G and a person v in another group G', which we have not yet defined. We make use of their ratings, age, gender and personality values. Specifically, we combine local similarities into a global similarity. The local similarities are as follows. For the users' ratings, we use the Pearson correlation (Equation 1) but normalized to $[0, 1]$, denoted here by $\rho_{[0,1]}$. For gender, we use an equality metric:

$$\text{eq}(x, y) \hat{=} \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

For ages and personalities, we use the range-normalized difference. Finally, the global similarity, psim_{cbr}, is simply an average of $\rho_{[0,1]}$, eq_{gender}, rn_diff_{age} and rn_diff_{pers}.

Again we have the problem of virtual users, who do not have ages, genders, personalities, or ratings. If either user is a virtual user, we simply take psim_{cbr} to be the mid-point of the similarity range. Empirically, this is 0.6. This means that there is neither an advantage nor a disadvantage to being matched with a virtual user and, since everyone must be paired with someone, this seems appropriate.

While this completes the definition of $\text{gsim}_{cbr}(G, G', f)$, it assumes that we give it a particular bijection, f, which pairs members of G with members of G'.

But, for the similarity, we want to consider *every* such bijection and settle on the *best one*, the one that gives the best alignment between the group members (their ages, genders, personalities, ratings) and the trust values. We must compute $\text{gsim}_{cbr}(G, G', f)$ for each bijection.

Let $\mathcal{B}(A, B)$ denote all bijections between equal-sized sets A and B. For example, if A is $\{a, b, c\}$ and B is $\{x, y, z\}$, then one bijection is $\{a \mapsto x, b \mapsto y, c \mapsto z\}$, another is $\{a \mapsto y, b \mapsto x, c \mapsto z\}$, and so on. Our definition of the similarity of group G and G' is based on finding the bijection, out of all the possible bijections, that maximizes $\text{gsim}_{cbr}(G, G', f)$:

$$\text{gsim}_{cbr}(G, G') \hat{=} \max_{f \in \mathcal{B}(G, G')} \text{gsim}_{cbr}(G, G', f) \tag{12}$$

Think of this as finding the pairing that maximizes total similarity. It does mean that a person in G might not be paired with the person who is most similar in G': it optimizes total similarity (over all group members and all trust values).

If G (and G') are of size n, then there are $n!$ bijections, and all must be considered. There is cause to be concerned whenever a computation requires consideration of $n!$ objects, because of the way that factorial grows with n. But, fortunately, the groups that most recommenders will deal with will be small enough to keep this manageable. For example, of 525 movie-going events reported to us through a Facebook poll, 21 were of size seven or a little above seven. Those that were of size seven would require consideration of $7! = 5040$ bijections, which remains manageable. If there are group recommenders where the number of bijections becomes too large, then some sort of sampling or greedy heuristic can be used, with the cost that the optimal bijection might be missed.

Step 3: Reuse Cases

At this point, we have explained our similarity measure, which is used to retrieve the k'' most similar cases. We must now explain how we reuse the cases that we have retrieved. To simplify the explanation, we will first consider the reuse of a single retrieved case, denoted $c = \langle id_c, \langle G_c, T_c \rangle, i_c, \{r_{u,i_c} : u \in G_c\} \rangle$.

Immediately, there is an issue that we must resolve. We want to predict G_a's ratings for each $i \in T_a$. But in case c, the selected item (e.g. the movie which the members of G_c went to see), was chosen from among T_c, which in most cases will not be equal to T_a: group G_a is going to the movies this week, whereas group G_c describes a previous outing to the movies, when it is probable that a different set of movies were on show. How can we transfer the contributions that the members of G_c made to the selection of $i_c \in T_c$ to the new situation where members of G_a must select an item from T_a?

The key to this is item-item similarity, which we described in Section 2. With item-item similarity, we can find the item $i \in T_a$ that is, for these users, most similar to $i_c \in T_c$. But there remains a problem. The Pearson correlation between two items i and i' is computed over the users who have rated both i and i' (Equation 2). There is no guarantee that there will be any user in either G_a or G_c who has rated both $i \in T_a$ and $i_c \in T_c$. But this is where the bijection f found

	Predicted ratings for candidate movies	
G_a	**Shrek**	**Hulk**
Ann	2	3
Ben	5	3
Col	2	4

	Actual ratings for the selected movie
G_c	**Twilight**
Dee	1
Edd	4
Flo	2

(a) No users in common

	Predicted & actual ratings		
Aligned users	**Shrek**	**Hulk**	**Twilight**
Ann+Dee	2	3	1
Ben+Edd	5	3	4
Col+Flo	2	4	2

(b) Using the bijection

Fig. 2. How item-item similarity is used

in Equation 12 can be used again. When comparing a rating from a user $u \in G_a$ for an item $i \in T_a$, we can use the rating $r_{f^(u),i_c}$ made by the corresponding user $f^(u) \in G_c$ for the item $i_c \in T_c$. It is by this means that we transfer the contributions that users in c made in their group decision to the group decision for $\langle G_a, T_a \rangle$.

But there is still a problem. The users $u \in G_a$ are unlikely to have a rating $r_{u,i}$ for the items $i \in T_a$, because T_a contains the candidate items that the group is choosing between. Instead, we use their predicted ratings $\hat{r}_{u,i}$, which we computed previously (Section 4.1) or, in the case of virtual users, the population average rating for the item.

Figure 2 contains an example. Suppose Ann, Ben and Col are in active group G_a, and that Dee, Edd and Flo are in case G_c. Figure 2a shows that we are unable to compute the item-item similarity between the selected movie from the case, Twilight, with the candidate movies, Shrek and Hulk. The movies have no users in common. For the active group, we have the predicted ratings for the candidate items; for the group in the case, we have the actual ratings for the selected movie. But suppose that, by the bijection, Ann maps to Dee, Ben maps to Edd and Col maps to Flo. Then, we can compute the item-item similarity between Shrek and Twilight by comparing Ann's predicted rating for Shrek with Dee's actual rating for Twilight, and Ben's predicted rating for Shrek with Edd's actual rating for Twilight, and so on. In effect, while there may be no users in these two groups who have rated both Shrek and Twilight, we are treating Ann & Dee as a 'single person' who has a rating for both Shrek (Ann's predicted rating) and Twilight (Dee's actual rating); see the Figure 2b.

We use Equation 2 to do this, but there are some changes. First, instead of computing the correlation over all users U, we compute it only over the users $u \in G_a$. Secondly, wherever the formula uses $r_{u,i}$, we now use u's predicted rating, $\hat{r}_{u,i}$; and wherever the formula uses $r_{u,i'}$, we now use the rating given by the user in G_c who corresponds to u, i.e. $r_{f^(u),i'}$.

We must still decide what to do if the groups are not of the same size. Consider the situation first where G_a is smaller than G_c . When we computed group similarity gsim_{cbr} earlier, we will have inserted extra virtual users into G_a. In this situation, we would not use G_a in place of U in Equation 2; rather, we would use the augmented version of G_a in place of U. That way, we can properly transfer the decision of the larger group to the smaller group: each person's contribution in the larger group is transferred to someone, either a real person from the smaller group or a virtual person who was inserted into the smaller group.

In the situation where G_a is larger than G_c, we will have earlier inserted virtual users into $G_c$in order to compute gsim_{cbr}. This time, however, we do use G_a in place of U. In other words, we compute the item-item similarity only on the ratings of the real people in G_a and their real counterparts in G_c. The virtual users were obviously not in reality present when G_c made its decision to consume i_c, so it makes no sense to transfer their contributions (i.e. none) to the decision-making of the smaller group G_a. This is achieved by simply computing item-item similarity over the real users and their counterparts, which is what Equation 2 will do if we use G_a in place of U. This does mean that, in these situations, there will be users in G_a whose opinions will be ignored (because they have no real counterparts in the smaller group, G_c).

So, we have explained how, given a retrieved case c, we can compute the similarity between i_c from c and each $i \in T_a$. We repeat this for each of the k'' retrieved cases. We can accumulate the item-item similarities and weight them by the group similarities. Formally, if C is the set of k'' cases, then the score for a candidate item $i \in T_a$ is $\sum_{c \in C} \text{gsim}_{cbr}(G_a, G_c) \times \rho_{i,i_c}$.

Step 4: Recommend Items

All the items in T_a have now received a score based on cumulating the similarities to the selected items in similar cases, weighted by the degree of similarity to those cases. So, finally, we recommend the k' items that have the highest scores.

5 Experiment

5.1 Group Recommender Dataset

We need a dataset with which we can evaluate our new system. We have built a social group recommender as a Facebook application [10]. But, at the time of writing, it cannot provide the volume of data that we need for conducting experiments. Unfortunately, neither are we aware of a public dataset for group recommenders. Hence, we created our own dataset. We have explained its construction elsewhere [8], and so we only summarize here.

We created our dataset from the MovieLens 1M dataset (`www.grouplens.org`), which gives us around 1 million ratings on a scale of 1 to 5 for around 6040 users for nearly 4000 movies. We created 100 groups from the MovieLens users, selecting group members at random but in such a way that everyone in a group falls into the same age range, and we ensured that there were at least 15 movies which are co-rated by all members of the group. When we create cases, these 15 movies will be the set T_c. We created 50 groups of size 2, 18 of size 3, 16 of size 4, 7 of size 5, 5 of size 6, and 4 where we took the size to be 7, this distribution being based on respondents to a Facebook poll that we administered.

The MovieLens dataset gives us the age, gender and ratings of each user. We had to impute personality values, which we did using the population distributions given in [15,14]. Similarly, we had to impute trust values between pairs of users in the same group. We took the trust between users u and u' to be the number of movies on whose ratings they agree as a proportion of the movies that either of them has rated. We take it that users agree if both have given the movie a rating above the ratings mid-point (which is 3) or if both have given the movie a rating below the ratings mid-point.

As we have explained, we have engineered matters so that, for each group, there is a set of 15 movies that all members of the group have rated, and we are treating these 15 movies as T_c, the set of movies that this group was choosing between. (Remember that T_c can be different for every group.) To create a case, we need to indicate which of these 15 movies the group will actually have chosen to go to see. For this, we got the opinion of four 'experts', two for each group. The experts voted on which three movies in T_c the group was most likely to select, placing movies into first, second and third position. Depending on the level of agreement between the experts, there might be ties for, e.g., first place, and so, although there were only three positions, the sets contained between three and five movies. We will designate this ordered set by E (for 'Expert') and we will use E_1 to mean movies in the first position in E, E_2 to mean movies in the first and second positions in E, and so on.

5.2 Evaluation Methodology

The dataset that we have created has 100 movie-going events, in other words 100 cases. We use a leave-one-out cross-validation methodology, where we remove each case in turn from the case base and present it to the recommenders.

We use three recommenders in these experiments: *Std*, *Soc* and *CBR*. *Std* is a simple group recommender: it uses the user-based collaborative recommender to predict the ratings each group member would give to the candidate items, and combines the ratings using the principle of *most pleasure*. *Soc* does the same but, before aggregation, it uses extra social data to modify individuals' predictions using the *delegation-based* method of Section 3. *CBR* is the new recommender, which uses cases to aggregate predicted ratings, which we described in Section 4.

Recall that each recommender recommends the top $k' = 3$ movies from the 15 candidates. Let R be the set of recommendations made by a particular recommender. Then we want to compare R with E from above. We computed total $success@n$ for $n = 1, 2, 3$, where $success@n = 1$ if $\exists i, i \in R \wedge i \in E_n$ and is 0 otherwise. For example, when using $success@2$, we score 1 each time there is at least one recommended movie in the top two positions of E. We also computed total $precision@n$ for $n = 1, 2, 3$, where $precision@n \hat{=} |\{i : i \in R \wedge i \in E_n\}|/n$.

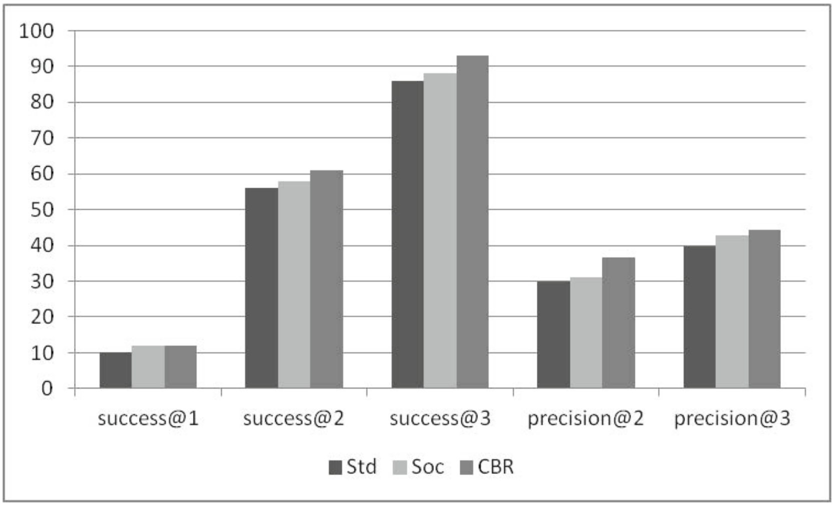


Fig. 3. Results of the experiment

For example, if no recommended movie is in the top two positions in E, then *precision*@2 = 0; if one recommended movie is in the top two positions in E, then *precision*@2 = 0.5.

5.3 Results

Figure 3 shows *success*@n for $n = 1, 2, 3$ and *precision*@n for $n = 2, 3$ (*precision*@1 = *success*@1 and is therefore not shown).

The first observation about the results is that, as n gets bigger, the results get better, e.g. *success*@2 results are better than *success*@1 results. This is not surprising: with bigger n, it is simply easier to make a recommendation that matches an expert judgement. The second observation is that results with *Soc* are better than results with *Std*: the use of the social information improves the quality of the recommendations. This is not a new result [11,8]. But what is new, our third observation, is the performance of the CBR system. It is never worse, and usually better, than both of the other systems. In detail, *CBR* has the same total *success*@1 (and *precision*@1) as *Soc*, just 12: it is very difficult for the systems to recommend the movie(s) the experts place in first position. But in all other cases, the CBR does better. For example, *Soc*'s *success*@2 = 58 but *CBR*'s *success*@2 = 61; and *Soc*'s *precision*@2 = 31 but *CBR*'s is 36.5. This shows the value of abandoning *Soc*'s single model of social behaviour, in favour of the lazy and local generalization that we obtain from the Case-Based Reasoning. We suspect that the differences would be even more marked in real datasets with more variability in the make-up of the groups.

6 Conclusions

We have described a new case-based group recommender system. It aggregates the predicted ratings of members of the active group but with reference to ratings of users in similar cases. A user-user similarity measure aligns members of the active group with members of the group in the case. The system uses an item-item similarity measure to transfer the contributions made to the group decision from the case to the corresponding users in the active group. One of its advantages is that preferences will be aggregated in different ways depending on how they played out in neighbouring groups, rather than according to a global, hypothesized theory of social interaction. This is borne out by our experiment, in which the CBR system is never worse, and is usually better, than a system that has a global model of group behaviour, expressed as a set of equations.

In our experiment, the selected item(s) in the cases are chosen by experts with knowledge of the actual ratings. So they are, in some sense, the absolutely best item(s). Therefore, it makes sense to run an experiment in which we see the extent to which the systems recommend such items.

But, matters are more complicated in practice. Suppose the recommender has recommended a movie to a group, and the group members have come back and rated that movie. We cannot simply retain this as a case in the case base. It may be suboptimal; it may not have been the best movie for this group. If we retain it, we will replay it in any future recommendation where it gets retrieved as a neighbour, where it may contribute to suboptimal decisions in the future.

In fact, this is not just a problem with CBR in group recommenders. It is a more general problem for the evaluation of group recommenders. It is very difficult to know whether they make good recommendations or not. If a user watches a recommended movie in a group and later gives it a low rating, this does not mean that the group recommender has done a poor job. It may even be that the group recommender predicted that this user would give a low rating. But the movie was recommended nonetheless, as it was judged to be the one that best reconciled the different tastes and personalities of the group members.

The implication is that, when group recommenders seek feedback from group members after recommended items have been consumed, they may need to solicit two types of feedback: the opinion of each individual user about whether the item satisfied him/her or not, but also the opinion of each individual user about whether the item satisfied the group as a whole or not.

Even if we get this more nuanced kind of feedback, it is not clear at this stage how to use it in evaluation of recommenders or in building case-based recommenders, not least because different group members may disagree on whether the recommendation satisfied the group or not. In case-based recommenders, the *outcome* part of the case might need to become much richer, to capture the opinions of the group members after they have consumed the item together, implying additional complexity in the kind of case-based recommender that we have described. This is a major issue for future work.

Other future work includes the use of datasets in which we explicitly arrange for the same group (or nearly the same group) to consume items together on a frequent basis, which can lead to a case base with more directly relevant cases in it. We hope too to gather more data from our Facebook group recommender and use this in future experiments.

References

1. Bridge, D.G.: The Virtue of Reward: Performance, Reinforcement and Discovery in Case-Based Reasoning. In: Muñoz-Ávila, H., Ricci, F. (eds.) ICCBR 2005. LNCS (LNAI), vol. 3620, pp. 1–1. Springer, Heidelberg (2005)
2. Golbeck, J.: Generating Predictive Movie Recommendations from Trust in Social Networks. In: Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) iTrust 2006. LNCS, vol. 3986, pp. 93–104. Springer, Heidelberg (2006)
3. Herlocker, J.L.: Understanding and Improving Automated Collaborative Filtering Systems. PhD thesis. University of Minnesota (2000)
4. Jameson, A., Smyth, B.: Recommendation to Groups. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 596–627. Springer, Heidelberg (2007)
5. Kolodner, J.L.: Case-Based Reasoning. Morgan Kaufmann (1993)
6. Linden, G., Smith, B., York, J.: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing 7(1), 76–80 (2003)
7. Masthoff, J.: Group modeling: Selecting a sequence of television items to suit a group of viewers. User Modeling and User-Adapted Interaction 14(1), 37–85 (2004)
8. Quijano-Sánchez, L., Bridge, D., Díaz-Agudo, B., Recio-García, J.A.: A case-based solution to the cold-start problem in group recommenders. In: Díaz Agudo, B., Watson, I. (eds.) ICCBR 2012. LNCS, vol. 7466, pp. 342–356. Springer, Heidelberg (2012)
9. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B.: An architecture for developing group recommender systems enhanced by social elements. International Journal of Human-Computer Studies (in press, 2012)
10. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B., Jiménez-Díaz, G.: Happy movie: A group recommender application in facebook. In: 24th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2011 (2011)
11. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B., Jiménez-Díaz, G.: Social factors in group recommender systems. In: ACM-TIST, TIST-2011-01-0013 (in press, 2011)
12. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An open architecture for collaborative filtering of netnews. In: Procs. of the Conference on Computer Supported Collaborative Work, pp. 175–186 (1994)
13. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative Filtering Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 291–324. Springer, Heidelberg (2007)
14. Schaubhut, N.A.: Technical Brief for the Thomas-Kilmann Conflict Mode Instrument. CPP Research Department (2007)
15. Thomas, K.W., Kilmann, R.H.: Thomas-Kilmann Conflict Mode Instrument. Tuxedo, N.Y. (1974)

Chapter 18

Social Factors in Group Recommender Systems

18.1 Citation

Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo,B, Jimenez-Díaz, G. Social Factors in Group Recommender Systems. Journal Transactions on Intelligent Systems and Technology. Special Issue: Social Recommender Systems volume 4, issue 1, pages: 8, ACM Transactions on Intelligent Systems and Technology, 2013. ISSN 2157-6904. JCR (2013): 9.39 (Computer Science, Artificial Intelligence, 1 out of 121, Q1. Computer Science, Information Systems 1 out of 135, Q1).

18.2 Contributions covered by this paper

In this paper we have detailed our techniques of making social group recommendations by including social factors in the recommendations processes. In it we detail different techniques to use the social factors in the group recommendations processes and we present and experimental analysis with both synthetic and real data where we measure the improvement in the accuracy of the recommenders that use a social approach.

Social Factors in Group Recommender Systems

LARA QUIJANO-SANCHEZ, JUAN A. RECIO-GARCIA, BELEN DIAZ-AGUDO, and GUILLERMO JIMENEZ-DIAZ, Universidad Complutense de Madrid, Spain

In this article we review the existing techniques in group recommender systems and we propose some improvement based on the study of the different individual behaviors when carrying out a decision-making process. Our method includes an analysis of group personality composition and trust between each group member to improve the accuracy of group recommenders. This way we simulate the argumentation process followed by groups of people when agreeing on a common activity in a more realistic way. Moreover, we reflect how they expect the system to behave in a long term recommendation process. This is achieved by including a memory of past recommendations that increases the satisfaction of users whose preferences have not been taken into account in previous recommendations.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Systems and Software—*social networking*; H.3.4 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

General Terms: Algorithms, Human Factors, Performance

Additional Key Words and Phrases: Memory, personality, recommender systems, social networks, trust

ACM Reference Format:
Quijano-Sanchez, L., Recio-Garcia, J. A., Diaz-Agudo, B., and Jimenez-Diaz, G. 2013. Social factors in group recommender systems. ACM Trans. Intell. Syst. Technol. 4, 1, Article 8 (January 2013), 30 pages.
DOI = 10.1145/2414425.2414433 http://doi.acm.org/10.1145/2414425.2414433

1. INTRODUCTION

Most of the work in recommender systems provides recommendations for individual users. However, there are many different activities that can be performed by groups of people, like watching a movie, going to a restaurant, listening to a radio station or traveling with friends. For these activities, recommender systems have to suggest items to groups based on the individual preferences of their members.

In recent years, the number of recommender systems that deal with the challenge of making recommendations for groups of people has increased. These recommenders are based on the aggregation of the individual preferences of every member or on the generation of a model based on the group itself for providing recommendations to the group. However, most of these systems deal with every individual preference in the same way, ignoring the personality of each member and the relationships among them within the group.

In this article we describe an approach to making recommendations for groups of people connected by social network structures. Our approach proposes making

Supported by Spanish Ministry of Economy and Competitiveness under grants TIN2009-13692-C03-03 and IPT-2011-1890-430000.
Author's address: Facultad de Informatica. Universidad Complutense de Madrid. C/ Prof. Jose Garcia Santesmases s/n. 28040. Madrid, Spain; email: gjimenez@fdi.ucm.es.

DOI 10.1145/2414425.2414433 http://doi.acm.org/10.1145/2414425.2414433

recommendations to groups by using existing techniques of collaborative filtering [Schafer et al. 2007], while taking into account several social factors that improve the accuracy of the system: the composition of the group personality and the social connections among the individuals.

The set of methods proposed can be integrated into any social network to provide recommendation capabilities to groups of users for activities such as going to the cinema, choosing a restaurant, planning trips, etc. For example, it is common to create events on Facebook where friends are invited to group activities. Users join the event and later decide its details. Quite often this decision consists of choosing an item or service to be consumed (movie, pub, restaurant, trip, and others). This decision is usually very tedious because it requires continuous communication among users. Our approach consists of automating the decision-making step by modeling the preferences of the user (as in collaborative filtering) and later aggregating these preferences to obtain a group recommendation. However, we can improve the aggregation strategies that generate the group recommendation because we are running over a social network and we can infer some social aspects about the users. These social aspects allow us to perform a more realistic simulation of the real discussion that would take place when deciding the item to consume. In our approach, personality is used to measure the users degree of permissiveness when his/her preferred items are not selected by the group. And social trust is exploited to calculate how the preferences of close friends may bias user choice.

Moreover, after we evaluate the possibilities that the personality composition of the group and the evaluation of the trust among group members offer, we study the option of introducing a system with memory to ensure a certain degree of fairness when the group recommendation process is repeated several times.

Our experiment is run with real users connected through social networks where we study how these features affect the group decision-making process. Our experiment in the movie recommendation domain compares the results of a standard group recommendation based on collaborative filtering and average aggregation with our improved recommenders, which use only personality, only social trust, a combination of both factors and the inclusion of a memory process.

The next section relates existing works in group recommender systems that include the features proposed in this article. Personality and social trust factors used to improve the recommendations are detailed in Section 3. Section 4 explains our proposal about a recommender system with memory, which assures fairness in group decisions. Our case study is explained in Section 5, followed by the experimental results obtained, shown in Section 6. In Section 7 we summarize the main conclusions that we obtained throughout this research. Finally, Section 8 details some of our ongoing and future work.

2. RELATED WORK

Group recommendation approaches are typically based on generating an aggregated preference by using the users individual preferences. As stated in Jameson and Smyth [2007] the main approaches to generating the preference aggregation are (a) merging the recommendations made for individuals, (b) aggreging ratings for individuals and (c) constructing a group preference model. A wide set of aggregation functions has been devised for combining individual preferences [Masthoff 2004], the average and least misery strategies being the most commonly used.

There is a proliferation of recommender systems that cope with the challenge of addressing recommendations for groups of users in different domains. MusicFX [McCarthy and Anagnost 1998] provides recommendations about background music at a fitness centre based on the preferences provided by the users in different musical genres. Polylens [O'Connor et al. 2001] is an extension of Movielens for generating recommendations to groups. Users create groups and the system recommends movies for these groups while trying to satisfy, at least the less satisfied members.

Another interesting content-based recommender system is Pocket Restaurant Finder [McCarthy 2002], which recommends restaurants for groups of people based on user location and the culinary characteristics of the restaurant. YuTV [Yu et al. 2006] is a TV program recommender for groups of viewers that uses a vector space model about features of TV programs (such as genre or actors) to find relevant recommendations for the groups.

Other examples are LET'S BROWSE [Lieberman et al. 1999], which recommends Web pages to a group of two or more people who are browsing the Web together, or FlyTrap [Crossen et al. 2002], a group recommender that selects music to be played in a public room. What all these recommenders have in common is that the group recommendations take the personal preferences obtained from their users into account but they consider each user equal to the others. The recommendation is not influenced by their personality or the way each one behaves in a group when joining a decision-making process. In our approach we propose to study how people interact depending on their personality or their closeness in order to improve group recommendations.

Several works have focused on organizing trips and visits for groups of tourists. Travel Forum Decision [Jameson 2004] helps a group of users to plan their vacations together. The system not only takes into account the preferences of each member in order to make recommendations but interaction among members is also reflected in the recommendation. The system provides a solution and allows group members to discuss. It acts as a mediator until they reach a solution. What we propose is to simulate this discussion in order to arrive at a solution. This way, our approach requires less interaction from users and it presents an immediate solution.

Other systems take into account the attitudes and behavior of other group members. CATS [McCarthy et al. 2006] is a conversational recommender for planning skiing holidays. The main feature of this system is that the recommendation is defined as an incremental process where users collaboratively refine the suggested recommendation by critiquing its features or discarding it. They consider that the preference of the current member partially depends on the preferences and/or anticipated behavior of other members. During the process of choosing a recommendation, users can see what other members have voted for, so they are conditioned by the opinions of other members. Our approach simulates this conditioning more thoroughly. CATS users need to read the information from other users in order to alter their initial opinion. Obviously this is only possible for users who vote later. However, our approach can simulate these alterations beforehand, by taking into account the strength of the relationships between group members.

Intrigue [Ardissono et al. 2003] plans visits for groups of tourists by weighting the preferences of different subgroups with special needs (like children or disabled people). The weight is computed using the subgroup size and its relevance within the whole group. Chen et al. [2008] propose the use of genetic algorithms to learn group preferences based on the known preferences of the subgroups within a group. Although the results seem to be significant, they suppose that the groups are fixed and they have previously rated some items together.

Other works focus on the integration of the group disagreements in the recommendation process. One of the most recent systems is GRec-OC [Kim et al. 2010], a book recommender system for online communities. GRec-OC provides recommendations based on the books that other similar groups have purchased and tries to reduce the dissatisfaction of individual members. Amer-Yahia et al. [2009] propose a recommender that aggregates prior member group preferences to create the recommendation. Then, preference disagreements between pairs of individuals are collected and employed to score and rank the recommended items. Finally, Masthoff and Gatt [2006] use individual satisfaction and emotional contagion in order to recommend a sequence of video clips for a group. They think that a member changes the selection of her best clip according to the clip selected during the previous selection step. This change can be reflected in the recommendation algorithm as an individual satisfaction function that computes the individual affective state. This state influences the affective state of the other members, producing an emotional contagion that should be taken into account during the recommendation process. Additionally, they point out the tendency of social status to influence the selection process.

There is agreement about the need to adapt the recommendation process to group composition [Jameson and Smyth 2007; Masthoff 2004]. Recent works have focused their studies on analysing the effectiveness of group recommendations according to different aspects, such as group size and inner group similarity [Baltrunas et al. 2010], or on studying different weighting models to combine the preferences of group members according to their role within the group or their activity [Berkovsky and Freyne 2010]. Additionally, it is also known that user preferences can be affected by other members of the group [Masthoff 2004; Chen et al. 2008]. However, most of the aggregation strategies employed in previous works combine user preferences without taking into account either the relationships among group members or the relevancy of each members preferences. The work dealing with these issues is limited. We observed that there was a need to modify those existing strategies that consider each user of the group as equal to the others. So we focus our line of work on reflecting the individual aspects of each user in the group and reflecting how they interact with each other.

Our recent work [Recio-García et al. 2009] involves the improvement of current group recommendation techniques by introducing a novel factor: the personality of every individual in the group when dealing with conflict situations. We have used a personality test to obtain the different roles that people play when interacting in a decision-making process. Besides the individual characterization of the people in the group, we are also studying other factors like the structure of the group itself and how users interact with each other.

Current research has pointed out that people tend to rely more on recommendations from people they trust (friends) than on recommendations based on anonymous ratings [Sinha and Swearingen 2001]. This factor is even more important when we are performing a group recommendation where users have to decide on an item for the whole group. This kind of recommendations usually follows an argumentation process, where each user defends her preferences and rebutts other's opinions. Here, when users must change their mind to reach a common decision, trust among users is the major factor.

There is a huge body of work about the generation of trust models. However, the raising of the current collaborative web (Web 2.0) has boosted the idea of Web Of Trust (WOT) [Golbeck 2006b; O'Donovan and Smyth 2005; Victor et al. 2008]. The WOT represents trust among users, modeled on an online network. There are specific approaches that use a custom trust network to recommend items. One example is FilmTrust [Golbeck 2006b], which exploits a custom network of trust among users according to movie preferences. However, these specific trust networks are quite difficult to generate because they require explicit feedback from users, and that can generate rejection.

Another promising approach is to infer knowledge of trust from existing social networks like Facebook or Twitter. These networks contain implicit information that can be exploited in order to improve the recommendation process. This option has the main advantage of being completely transparent to users. Users are not required to provide explicit information about their trust to other users because this knowledge is extracted implicitly from their daily interaction in the social network. However, it has the obvious drawback that every user involved in the group recommendation process must belong to the social network. Nevertheless, the rising popularity of this kind of Web applications minimizes this risk. Even more, it is becoming usual to organize events (like going to the cinema) through social networks, so group recommendation techniques could be integrated into these Web sites.

Recent works provide evidence that users prefer the sort of system that relies on trust in social networks because users tend to prefer recommendations from people they know and trust [Sinha and Swearingen 2001]. Golbeck [2006a] presents a study of how to infer trust relations within social networks. The computational problem of trust is to determine how much one person in the network should trust another. Certainly, trust inferences will not be as accurate as a direct rating. But in this work, Golbeck presents an algorithm for inferring trust in networks with continuous rating systems, named TidalTrust, that improves the accuracy by 10%.

Another important matter that should be taken into account is the special importance that some works give to avoid repeated recommendations or recommendations that tend to be detrimental to the same group members repeatedly. MusicFX employs a weighted random policy for selecting one of the top radio stations selected by the recommender, instead of always selecting the top category. Another solution is taking into account the history of the results produced by the recommender. For example, in FlyTrap previous selections are taken in consideration. This way, when they choose the next song to be played, abrupt changes of genre do not appear. Another system that takes previous selections into account is PoolCasting [Baccigalupo and Plaza 2007]. It uses a Case-Based Reasoning system to generate a sequence of songs customized for a community of listeners. To select each song in the sequence, first a subset of songs musically associated with the last song of the sequence is retrieved from a music pool; then the preferences of the audience expressed as cases are reused to customize the selection for the group of listeners; finally, listeners can revise their satisfaction (or lack of) for the songs they have heard.

The works presented rely on the different factors involved in our proposal: personality, trust and memory. However, we have not found any work that integrates and evaluates these three factors in group recommendation processes. Therefore we consider that our approach improves them by making a more exact representation of how group argumentations take place in real life. This is the main contribution of this article. Next section describes our approach for including personality and trust in group recommenders, whereas Section 4 explains the inclusion of a long-term memory of past recommendations.

3. INCLUDING PERSONALITY AND SOCIAL TRUST IN GROUP RECOMMENDATION

Most of the previous works in group recommendation consider the preferences of every member of the group with the same degree of importance and try to satisfy the preferences of every individual. However, groups of people can have very different characteristics, like size, the relationships among their members or the distribution of people with similar or antagonistic personal preferences. Our approach presumes that the general satisfaction of the group is not always the aggregation of the satisfaction of its members.

It is a fact that when we face a situation in which peoples concerns appear to be incompatible, a *conflict situation* arises. Different people have different expectations and behavior in conflict situations that should be taken into account. When we started our research to improve the group recommendation process, we decided to study the different behaviors that people have in conflict situations according to their personality. In Recio-García et al. [2009] we presented a method for recommendation to groups that distinguishes among the different types of individuals according to their personality.

After evaluating the different behaviors that people have when carrying out a decision-making process according to their personality, we decided to study how trust among group members will affect the recommendation process for a group. To do so, we analyzed the social factors that could reflect the trust among users.

Once we can characterize the personality of group members in conflict situations and the trust among these members, we can integrate these factors in the recommendation process. Our group recommendation method is based on the preference aggregation approaches most commonly used [Masthoff and Gatt 2006; O'Connor et al. 2001], which aggregate the users individual predicted ratings to satisfy the greatest number of members. Therefore, the basic building block of our group recommender is an individual recommender that predicts the preferences for a given user. However, we bias the individual predictions with the personality and trust features of the user.

The individual recommender implements the collaborative filtering algorithm described in Kelleher and Bridge [2004]. We have chosen this algorithm because it is broadly used to recommend items when the modeling of user preferences is not a valid option (as in most of real scenarios [Linden et al. 2003; Schafer et al. 2007; Sarwar et al. 2001]). This algorithm requires users to rate an initial set of items. Then, those ratings are used to estimate the predicted rating for an unrated item. The algorithm runs as follows: First, it weights all users with respect to the similarity with the active user by computing the Pearson Correlation coefficient.[1] Next, it selects a set of the most similar users as a set of predictors. Finally, it normalizes the ratings and computes a prediction from a weighted combination of selected predictors' ratings. This individual recommender returns an estimated prediction $pred(u, i)$ for a user u and a given item i.

In our approach, the ratings predicted by the individual recommender are combined to obtain the predicted scoring of the item i for the group. A common way to combine these individual predictions $pred(u, i)$ into a prediction for the group, $gpred(G, i)$, is the average function [Masthoff 2004]:

$$gpred(G, i) = \frac{1}{|G|} \sum_{u \in G} pred(u, i). \tag{1}$$

Here G is a group of users, which user u belongs to. This function provides an aggregated value that predicts the group preference for a given item i. Using this average function, our group recommender proposes the set of k items with the highest group predicted scoring. The average aggregation strategy has been successfully applied in many recommenders like Intrigue [Ardissono et al. 2003], Travel Decision Forum [Jameson 2004] or YuTv [Yu et al. 2006], so we have chosen it as our baseline.

Our group recommendation strategy employs the basic average function using modified individual predictions that include personality and trust factors.

$$gpred'(G, i) = \frac{1}{|G|} \sum_{u \in G} pred'(u, i). \tag{2}$$

The improvement in the accuracy of the group recommender against the baseline recommender mainly depends on the way we compute the new $pred'(u, i)$ values using the personality and trust values. Following sections will describe how we compute the personality value (Section 3.1), the trust value (Section 3.2) and how to combine the individual ratings with these personality and trust factors (Section 3.3). Finally, the evaluation of that improvement is detailed in Section 5.

[1]Pearson Product Moment Correlation coefficient (called Pearson's Correlation for short) reflects the degree of linear relationship between two statistical variables. In this case these are the ratings given by two users.

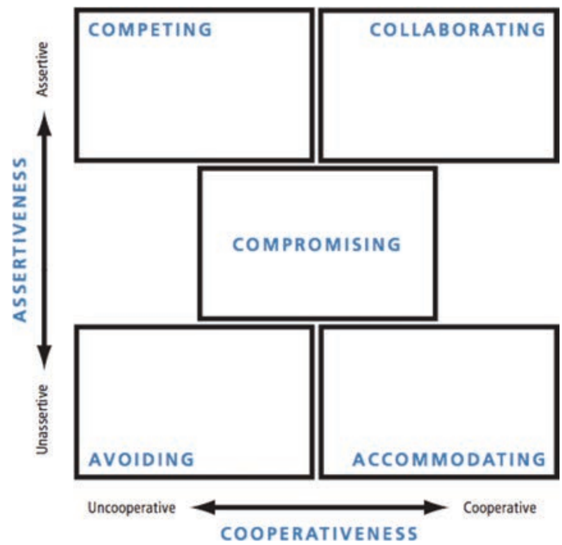


Fig. 1. TKI personality modes.

3.1. Personality Values

Our proposal characterizes people using the Thomas-Kilmann Conflict Mode Instrument (TKI) [Thomas and Kilmann 1974]. TKI is a test designed to measure the behavior of people in such situations. It is a leading instrument in conflict resolution assessment that is often used by Human Resources and Organizational Development consultants to facilitate learning about how conflict-handling styles affect personal and group dynamics. This test describes a person's behavior in conflict situations along two basic dimensions: assertiveness and cooperativeness. These two dimensions of behavior can be used to define five personality modes of dealing with conflicts: competing, collaborating, avoiding, accommodating and compromising (see Figure 1).

In general terms, the inclusion of personality in the group recommendation process runs as follows: assertive behaviors penalize the differences with the best choice of other members (the other choices do not satisfy her personal concerns), while cooperative behaviors reward the differences with the best choice of other members (it is not her preference choice but it will be good enough for other members and for the group).

To determine the personality, our users fill out the TKI test. It consists of 30 different situations with two possible answers. Depending on the answers, a score is assigned for each personality mode. If a score is below or above the 25 or 75 percentile according to the population, then that user is classified as having a low or high personality mode. This way the test indicates if a user has high or low competitive, collaborative, avoiding, accommodative, and compromising modes. Following the schema shown in Figure 1, if a user has a high competing and collaborative mode she is assigned a high assertiveness value. High avoiding and accommodating personality modes are considered as low assertiveness. Following the second dimension, a high cooperativeness value is given to a user if it has a high accommodating and collaborating mode. Therefore, the *Assertiveness* and *Cooperativeness* values are a weighted sum of the five personality modes. Details about the calculation of these values are explained in Recio-García et al. [2009].

Once each user u completes it, we calculate a value that represents how selfish or cooperative she is. This value is the difference between the assertiveness and cooperativeness of the user. We call this value the *Conflict Mode Weight* and it represents the predominant behavior for that particular user according to her TKI evaluation. For the sake of simplicity, we will use p_u to refer the conflict mode weight of user u, as this value summarizes her personality.

The p_u factor is computed using the following equation:

$$p_u = \frac{1 + Assertiveness(u) - Cooperativeness(u)}{2}. \quad (3)$$

We note that people with strong personalities have a high p_u value, while a low p_u value represents an easygoing personality. The p_u value fits in a range of [0,1], 0 being the reflection of a very cooperative person and 1 the reflection of a very selfish one.

3.2. Social Trust Factors

Social network users post on their profiles a huge amount of personal information that can be analyzed to compute tie strength with other users: likes and interests, personal information, pictures, games, etc. [Golbeck 2006a; Gilbert and Karahalios 2009]. These factors are characteristic of personal social networks and they cannot be extrapolated to other kinds of networks [Wu et al. 2010]. However, previous works have reported that trust and tie strength are conceptually different but that there is a correlation between them [Levin et al. 2004]. For this reason, we decided to employ a set of ten different factors for computing the tie strength, which we will employ as a measure of trust between two users u and v in a given group. [Granovetter 1973] defines the strength of a tie as a (probably linear) combination of the amount of time, the emotional intensity, the intimacy (mutual confiding), and the reciprocal services that characterize the tie. The literature reviewed identifies these four factors as some of the major dimensions of predictive variables. With these dimensions as a guide, Gilbert and Karahalios [2009] identified 74 Facebook variables as potential predictors of tie strength. From those 74 we used 10, the ones that were most representative of each of the major dimensions and could also be extracted from any user profile on a social network (for example, inbox messages cannot be read from outside the user account). These factors are the following.

— $f_1(u, v)$: Distance in the social network. We check if the two users are friends; in case they are not we look to see if they have any friends in common.
— $f_2(u, v)$: Number of friends in common. We count the number of mutual friends they have.
— $f_3(u, v)$: Intensity of the relationship. We count how often their name appears on each other's walls. This is the number of times that they have posted a comment, they have discussed a publication or they have commented on a picture.
— $f_4(u, v)$: Intimacy of the relationship. We classify relationships by finding keywords in their wall interactions. These keywords represent intimacy levels like very affectionate, familiar/caring, friendly, work-related, or casual.
— $f_5(u, v)$: Duration. We calculate how long they have known each other. To do so we read their own age and contrast it with the information related to schools, universities, work, family relations, etc.
— $f_6(u, v)$: Reciprocal services. We count the number of videos/songs/webs posted on each others walls and we also count the games/applications shared (like Pet Society, Mafia wars, Music challenge, among others).
— $f_7(u, v)$: Structural variable. We count the common interests described in their profile, like movies, books, or general interests. We also count how many groups they have both joined or become fans of.
— $f_8(u, v)$: Social distance. We count how many of the following sections in their personal information are shared: political beliefs, school/universities, religious beliefs and demographical situation.

—$f_9(u, v)$: Status. We read the information about the relationships between users, like "family" or "relationship status."
—$f_{10}(u, v)$: Pictures. We compute the percentage of pictures where they appear together.

Some of these factors can be easily obtained from social networks. For example, Facebook directly provides the information required by factors 2, 9, or 10. However, other factors require extra analysis of the social network. More specifically, there are some factors, like privacy, that require the processing of the messages exchanged. Here, common Information Extraction techniques can be applied to extract the keywords that identify the nature of the text.

Once we choose the different factors involved in the computation of social trust, we must combine them to obtain a final value. Previously described factors may have a different impact on the recommendation process, so we have decided to combine them using a weighted average:

$$trust(u, v) = t_{u,v} = \sum_{k=1}^{10} w_k \cdot f_k(u, v). \tag{4}$$

Note that $t_{u,v} \in [0, 1]$. We have measured the importance of every factor w_k by using an experimental approach. The case study presented in Section 5 shows how these factors are weighted to maximize the accuracy of our recommender.

3.3. Integration of Personality and Social Trust in the Group Recommendation Process

The next step refers to the inclusion of personality and trust factors in our group recommendation process. As described in Section 3, our aggregation function combines the modified individual rating predictions $pred'(u, i)$ of each group member (see Equation (2)). We propose three different approaches to computing $pred'(u, i)$. These formulas combine individual ratings (predicted by the individual recommender) with the personality and trust factors.

The following methods (more concretely, delegation-based and influence-based prediction methods) take several ideas from the social sciences. One is *emotional contagion*: the influence of an individual's affective state on that of others in the group [Masthoff and Gatt 2006; Hatfield et al. 1994; Barsade 2002]. Another important social aspect is *conformity* [Deutsch and Gerard 1955], whereby judgements are influenced by those of others. It means that individuals may probably give in because they trust other people's judgement more that their own. Therefore, conformity may cause individuals to change both their public and private opinions. Next, we will detail our proposed methods.

3.3.1. Personality-Based Rating Prediction. The Personality based rating approach takes into account the differences in personality between pairs of individuals in the group. It is based on the modified average satisfaction employed in our previous work [Recio-García et al. 2009]. This strategy reflects that assertive characters will have more influence in the aggregated scoring than cooperative characters. This factor is computed as the distance of the personality values. Our approach uses the type of personality to weight the influence of user ratings during the recommendation process. If we consider $pred(u, i)$ as the individual rating predicted by the collaborative recommender for a given user u and a certain product i, the following equation computes *personality-based rating prediction* ($pbr(u, i)$):

$$pred'(u, i) = pbr(u, i) = \frac{1}{|G| - 1} \cdot \sum_{v \in G \wedge v \neq u} (\, pred(u, i) + (p_u - p_v)\,). \tag{5}$$

In this equation, $|G|$ represents the number of components in the group and this value is used to normalize the result. p_u and p_v are the conflict mode weights of users u and v. Note that the difference between p_u and p_v will boost the $pred(u, i)$ prediction if the personality of user u is stronger than user v. Otherwise, the predicted value is decreased.

3.3.2. Delegation-Based Rating Prediction. Delegation-based rating prediction is inspired by an approach previously described in Golbeck [2006a], where individual prediction is based on the estimated rating of other users. The idea behind our method is that users create their opinion based on the opinions of their friends. The average of these opinions is weighted depending on the level of trust with every friend. Additionally, the personality of every friend is also taken into account, thus modifying the base opinion. The delegation-based rating approach tries to simulate the following behavior: when we are deciding which item to choose within a group of users we ask the people who we trust. Moreover, we also take into account their personality to give a certain importance to their opinions (for example, because we know that a selfish person may get angry if we do not choose her preferred item).

Delegation-based rating prediction ($dbr(u, i)$) given an user u and an item i is computed in the following way:

$$pred'(u, i) = dbr(u, i) = \frac{1}{|\sum_{v \in G} t_{u,v}|} \sum_{v \in G \wedge v \neq u} t_{u,v}(\ pred(v, i) + p_v\ ) \tag{6}$$

In this formula, we take into account the estimated rating $pred(v, i)$ for every individual v for the i item. This prediction is increased or decreased depending on her personality (p_v), and finally it is weighted according to the level of trust ($t_{u,v}$). Note that this formula is not normalized by group size and it uses the accumulated personality. This is the way it was originally described in Golbeck [2006a]. Therefore, this formula (and the one proposed next) could return a value out of the ratings range. This is simply managed by the recommender by choosing the closest value within the valid range.[2]

3.3.3. Influence-Based Rating Prediction. Influence-based rating prediction simulates the influence that each friend has in a given person. This approach supposes that the user may modify her preference for an item depending on the rating given by her friends to the same item. For example, if our rating for an item is 3 and our friend has a 5 rating for the same item, we could think of modifying our rating to 4. Depending on the trust with this friend, we decide the level of variation for our rating (i.e. 3.5 if the trust is low, and 4.5 if trust is high). Furthermore, the variation of our rating also depends on our personality. If we have a strong personality we will not be willing to change our rating, but if we have a weak personality we could be easily influenced by other users. By combining both factors we get the following formula:

$$pred'(u, i) = ibr(u, i) = \ pred(u, i)\ + (1 - p_u) \frac{\sum_{v \in G \wedge v \neq u} t_{u,v} \left(\ pred(v, i) - pred(u, i)\ \right)}{|G| - 1}. \tag{7}$$

In this formula, the individual rating prediction for the item ($pred(u, i)$) is modified according to its difference with the ratings predicted for other users

[2]For example, if valid range is [0..5] and the estimated rating is 5.1, the returned value will be rounded to the interval (5.0). Although this may theoretically affect accuracy, in practice it is not usual to get an out-of-range result.

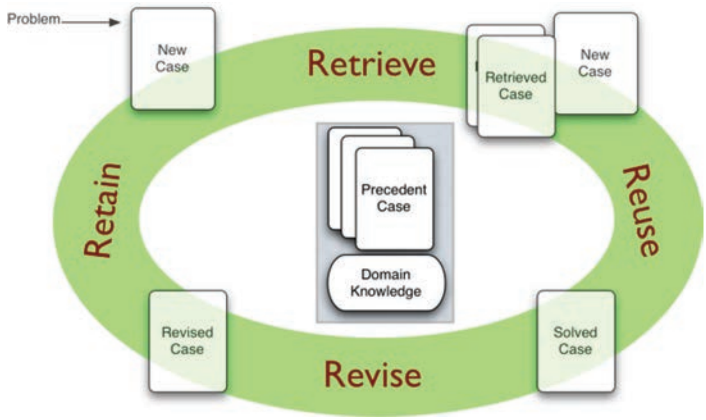


Fig. 2. Case-Based Reasoning Cycle.

$(pred(v, i) - pred(u, i))$. This difference takes into account the trust between users $(t_{u,v})$. Finally, the accumulated difference is weighted according to our personality in a complementary way $(1 - p_u)$.

4. RECOMMENDATIONS WITH MEMORY

Until now we have focused on the specific situation where the recommender makes a recommendation just once. But frequently, a group will expect to use the system several times, thereby getting a bigger sample of recommendations. However, these novel recommendation techniques will always tend to favor the same users (because they have stronger personalities or because they are closer to each other). Therefore, we could end up with a situation where we have some dissatisfied users because we take their opinions less into account for the group's sake. In order to avoid a situation of high dissatisfaction levels in the group, we should take a certain degree of fairness into account. For example, we could face a situation where a certain recommendation is very promising for the group in general, but one of the users could end up very dissatisfied with this recommendation. It would be desirable that future recommendations would favor this component of the group so that she could reach a proper level of satisfaction.

To address this issue, we propose the use of a memory of past recommendations. Having recommendations with memory means that we are able to create a system that remembers all the previous recommendations for a given group. We believe that this is a necessary step when providing a whole set of fair recommendations. This way, if one member accepts a proposal that she was not interested in, next time her preferences will be prioritized in the recommendation process. This means that her opinion will have a higher weight next time. These weights will also be influenced by the different personalities of each group member as we applied in Section 3.1. For example, someone with a strong personality that has been negatively affected would be immediately compensated next time; however someone with a mild personality would not have problems giving in several times.

The process that we have followed is very similar to the Case Base Reasoning (CBR) cycle [Aamodt and Plaza 1994], sketched in Figure 2. CBR is a successful and established methodology in Artificial Intelligence that has inspired us in the implementation of our recommender with memory. In our system each recommendation provided by the group recommender will be stored as a new case that can be used later to improve the next recommendation. This fact corresponds to the *retain phase* in the CBR cycle. This way, we acquire the experiences that will be useful for the resolution of future recommendations because we will know which products have been recommended to a group. We also store how satisfied each member of the group is with this recommendation, so we are able to adjust the satisfaction factor in future recommendations. Before making the following recommendation we will check the previous situation, which in the CBR cycle would be the *retrieve phase*. Once we obtain that information we can perform a new recommendation, while taking into account what we have retained (the products that we have already recommended and how satisfied each of the member of the group is). This is equivalent to the *reuse phase* in the CBR cycle. Last but not the least, we will modify the recommendation so that the products proposed are not repeated and so that we can assure a certain degree of fairness when we benefit the preferences of each user. This last phase, the *revise* one, closes the CBR cycle.

The following formulas reflect these concepts, the first one applied to the delegation-based rating prediction method and the second one to the influence-based rating prediction method. These formulas include a factor s_v that reflects the level of satisfaction of the user v with the previous recommendations. Note that here we always take into account both the personality and trust factors because they return better results than including them separately (as we will show in Section 6).

$$dbrm(u,i) = \frac{1}{|\sum_{v\in G} t_{u,v}|} \sum_{v\in G \wedge v\neq u} t_{u,v}(\, pred(v,i) + p_v\,) + \alpha(\mathbf{1} - \mathbf{s_v})\mathbf{p_v} \tag{8}$$

$$ibrm(u,i) = pred(u,i) + (1 - p_u)\frac{\sum_{v\in G \wedge v\neq u} t_{u,v}\left(\, pred(v,i) - pred(u,i)\,\right) + \alpha(\mathbf{1} - \mathbf{s_v})\mathbf{p_v}}{|G| - 1}. \tag{9}$$

The satisfaction value s_v is the level of satisfaction of a user v. A user who is extremely happy with the recommendations will have this satisfaction measure value close to 1. However, the more upset with the recommendations she is, the more that this value will decrease, reaching 0 in the worst case. The computation of this value is detailed in Section 5.5, but in general terms it sorts all the items according to user preferences and checks the position of the item selected for the group. This position is inversely correlated to the s_v value. Parameter α has been experimentally selected, and it is used to modify the impact of the use of memory in the modified rating. Moreover, the satisfaction value is weighted depending on the users personality to reflect the importance of satisfying that concrete user. Once the recommendation process has finished the s_v value is updated for every user. The computation of the s_v value will be described in Section 5.5, although in general terms it is updated every time a user gets a recommendation according to her preference for the item selected by the group.

It is important to note that the methods described in Equations (8) and (9) use the satisfaction of other users s_v instead of the target user u to obtain a predicted rating. The use of group satisfaction is based on some results from organizational behavior and social psychology that have highlighted the concept of *emotional contagion* [Masthoff and Gatt 2006]. This social aspect states that the satisfaction of an individual is likely to depend on that of other individuals of the group [Barsade 2002; Hatfield et al. 1994].

Next we explain how these factors would be taken into account in a real group. For example, Peter, John, and Mary go together to the cinema for the first time. Peter and Mary are both very stubborn and they have strong personalities (for example, they have a $p_{Peter} = p_{Mary} = 1$), whereas John is very accommodating and has a mild personality (being $p_{John} = 0.2$). Initially they all have the same level of satisfaction; this means that

$s_v = 1$ for all of them. As this is the first time that they go to the cinema together, they do not have a history together so the satisfaction levels will not be taken into account. As they all have a satisfaction value of 1, the second part of the equations explained before (Equations (8) and (9)) is canceled. It means that $\alpha \cdot (1 - s_v) \cdot p_v = 0$. Therefore the recommendation process is the same one that we have applied previously when we did not take into account either the satisfaction or the history; these equations correspond to the original processes based on delegation or influence (Equations (6) and (7)).

As a result of this process our users will be provided with a recommendation. For simplicity, we are going to focus on the best three movies that the recommender has found for the group (*Amovie, Bmovie, and Cmovie*). Let's suppose they choose to watch the movie *Amovie*. Once we have their feedback we store the movie selected and we update the values that measure the level of satisfaction s_v. To recalculate these satisfaction values we check the rating given by the each users individual recommender to the actual movie selected by the group. If this movie would never be recommended individually to the user, the value of satisfaction will be very low. On the other hand, if the movie would indeed be recommended individually to the user, the value s_v will still be near 1.

Lets say that Peter, who has a very strong personality, is especially dissatisfied and John, the one with the mild personality, is also dissatisfied with the recommendation, so their discontent values are updated to $s_{Peter} = 0.2$ and $s_{John} = 0.4$, respectively. The calculation of the discontent values after each recommendation is based on how many movies preferred by the user appear in the movie list proposed by the group recommender. This process will be detailed in Section 5.5. This way, Mary will be very happy because she has managed to watch the movie that she wanted, so her satisfaction value is still $s_{Mary} = 1$.

The next time that they go to the cinema together, Mary's opinion of the movie will only be influenced by the trust she has in her friends and by her own personality, which for example is reflected in the delegation-based method with $t_{uv}(\,pred(v, i) + p_v\,)$. The same fact happens to the opinions of Peter and John. However, as they were not satisfied last time, their opinion will have an increment because their level of discontent is being added in proportion to their personalities (through the $\alpha \cdot (1 - s_v) \cdot p_v$ factor in the equations). In the particular case of Peter the increment in his rating will be higher because $p_{Peter} \gg p_{John}$ –as we said, p is higher when the personality is more conflictive and lower when it is more cooperative. Peter is also more dissatisfied than John, so his opinion will have a higher impact (at this moment $s_{Peter} = 0.2$ and $s_{John} = 0.4$).

Memory-based algorithms provide a new set of item score predictions by taking into account these factors. However, the memory of past recommendations is also used in another, different way. Once the users have decided on the movie they are going to watch together, it is included in the memory of the recommender. Therefore, if the recommender proposes a movie that they have already seen (because it has a high predicted score), it is automatically rejected and the next best movie will be the recommended one.

The levels of discontent and the database of the movies watched are updated each time we get the feedback from our users. The next time that this group wants to go to the cinema together the process will start all over again with the satisfaction levels and the set of movies watched stored in the system. We can see the diagram of this process of recommendation with memory in Figure 3.

Another important issue about this approach is the scope of the memory of user satisfaction. We can update the s_v value to reflect the satisfaction according to the last immediate group recommendation or to take into account previous ones. Therefore, the satisfaction value for an execution e of the recommender may depend on the satisfaction of the user with the items recommended in e –$satisfaction(v, G)$– but it also depends

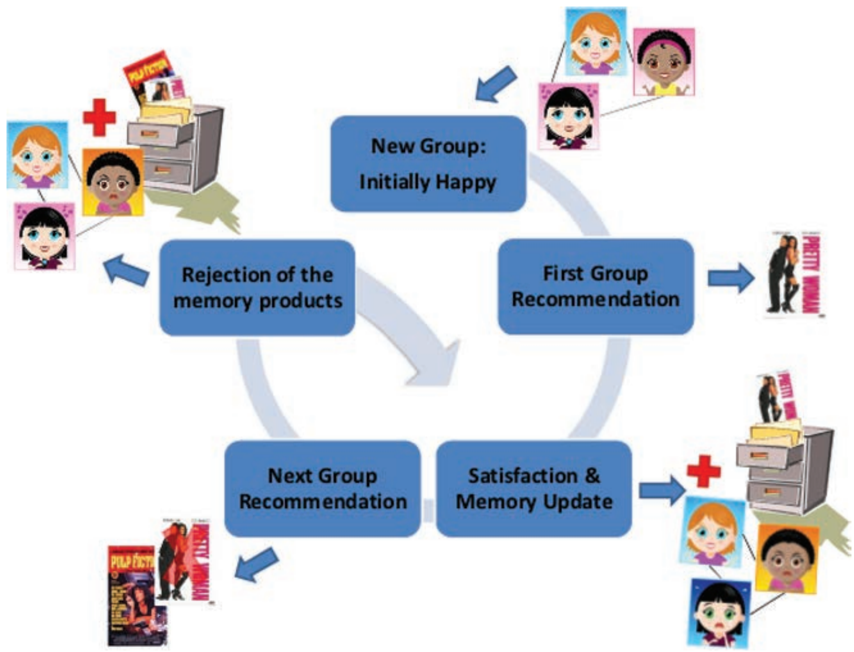


Fig. 3. Example of the recommendations with memory process.

on her satisfaction in the previous recommendation $e-1$, as reflected in the following equation:

$$s_v(e) = (1-\delta) \cdot satisfaction(v, G) + \delta \cdot s_v(e-1). \tag{10}$$

In this equation we use the $\delta \in [0..1]$ parameter to adjust the impact of previous satisfaction when updating that value. The details about this process are provided in Section 5.5.

Now that we have finished the exposition of the theoretical algorithms we will next present a case study that validates our proposals.

5. CASE STUDY: MOVIE RECOMMENDATION

Now that we have described our proposal, we will evaluate our algorithms using the movie recommendation domain. We chose this domain because it is a very accessible field with several datasets available and, more importantly, easy to query about to users. The specific goals of our experiments are:

G1. To study the influence of personality and trust factors in the recommendation by comparing the results obtained by the proposed algorithms: personality-based rating prediction (PBR), delegation-based rating prediction (DBR) and influence based rating prediction (IBR).

G2. To measure the robustness of the recommenders based on the proposed algorithms in comparison with the group size.

G3. To determine if the recommenders based on the proposed algorithms are biased by group homogeneity, according to the individual personalities.

G4. To determine if the recommenders based on the proposed algorithms are biased by the trust strength of the group, according to the trust between group members.

G5. To study the improvements obtained by including memory capabilities in the proposed algorithms.

5.1. Experimental Setup

In order to perform our experiment in the movie recommendation domain, we created two events in two different social networks, Facebook[3] and Tuenti.[4] Fifty-eight real users participated in our experiment.

During the experiment we employed the following instruments.

—*TKI*. It is a personality test [Thomas and Kilmann 1974] with 30 items. This test is used to obtain a personality profile about the way a person deals with conflict.
—*Movie listing*. A list of 15 recent movies (of year 2009), which represents a movie listing from a cinema. This movie listing was chosen heterogeneously from movies from the MovieLens dataset [Bobadilla et al. 2009].
—*Movies to rate*. A list of 50 movies selected from the MovieLens dataset.

The experiment was carried out as follows.

(1) Every user completed three different questionnaires[5] The answers to these questionnaires, received directly into our email, were analyzed to define each participant's user profile, which contains information about their personality and their individual movie preferences.
   —The first questionnaire consists of completing the *TKI* test described earlier. This way, we obtain the personality profile p_u for every user.
   —The second questionnaire consists of rating –with a score between 0.0 and 5.0– the movies from the *Movies to rate* list that the user has previously seen. This questionnaire compiles every users individual preferences about cinema. On average, the users rated 30 movies from the *Movies to rate* list. These preferences are employed in the collaborative filtering algorithm to compute individual predictions $pred(u, i)$.
   —The third questionnaire asks users to choose their 3 favorite movies from the *Movie Listing*. These movies are the ones they enjoyed best or would actually like to watch. The goal of this questionnaire is to obtain the users individual preferences and, therefore, to be able to measure the accuracy of the individual recommender.
(2) We have computed the trust value $t_{u,v}$ for every pair of users with the factors detailed in Section 3.2. The computation of the trust value using the information extracted from the social network is detailed in Section 5.3.
(3) We have managed to form groups of users. Afterwards, we asked each group to choose which 3 movies from the *Movie Listing* they would actually watch together. The movies chosen by each group G are stored as the *real group favorites* list rgf_G. We compiled this list for 15 groups with 3 (5 groups), 5 (6 groups) and 9 (4 groups) members.
   The first two questionnaires, the *TKI* test and the movie rating questionnaire, are the most time-consuming tasks for the user before taking part in the group recommendation process. However, the TKI test was selected from among other personality tests due to its brevity—it takes no more than 10 minutes—and the movie rating questionnaire is similar to the one proposed when a user starts to use Movielens, which can be completed in a few minutes. Once these questionnaires are completed, the use of the recommender is really straightforward, thus demonstrating its feasibility in the real world.

[3] http://www.facebook.com.
[4] http://www.tuenti.com. Tuenti is the most popular social network in Spain for people in their twenties.
[5] Questionnaires are accessible at http://www.lara.warhalla.com/ (in Spanish).

It is important to detail the implications of having a list rgf_G with only 3 movies. It limits the kind of evaluation functions that we can use to evaluate the accuracy of our recommender. If we had a real group favorites list with the 15 movies from the movie listing ordered according to the preferences of the group, we could apply several evaluation measures based on comparing both rankings: the predicted ranking of movies returned by the recommender and the real ranking (Mean Absolute Error, normalized Discounted Cumulative Gain are examples of such metrics [Herlocker et al. 2004; Baltrunas et al. 2010]). However, when we asked users to rank the movies they would actually watch together given a movie listing with 15 items, we realized that they only care about the 2 or 3 movies they really like. Then they order the remaining movies almost randomly because they are not interested in these movies at all. Therefore, we decided to ask them to choose only their 3 favorite movies as the ordering of the remaining ones would be noisy (or at least irrelevant) information for our experiments.

(4) We have implemented a set of group recommender systems that follow the previously described algorithms and employ the user profiles obtained in previous steps. The implementation details are presented in Section 5.4.
(5) We have employed the recommenders to provide a list of recommended movies for every group. The recommendation list is extracted from the *Movie listing*. The top 3 recommended movies are stored as the *group favorite* list gf_G.
(6) The *real group favorites* lists rgf_G and the *group favorite* lists gf_G have been compared to measure the accuracy of the group recommendation. The evaluation metrics employed to measure the performance of the proposed algorithms are described in Section 5.2.

Finally, we performed an additional experiment with a recommender system based on memory. The details of this experiment are described in Section 5.5.

5.2. Evaluation Metrics

The aim of the evaluation is to compare the results of our recommender system to the real preferences of the users (that is, what would happen in a real life situation). This evaluation has some particular features that must be taken into account. First, we are not interested in a long list of ordered movies when estimating the movies a user or group should watch. As we have previously explained, real users are only interested in a few movies they really want to watch. This fact discards several evaluation metrics that compare the ordering of the items in the real list of favorite movies and the estimated one. On the other hand, the number of relevant and retrieved items in our system is fixed. Therefore, we cannot use general measures like recall or precision. However, there are some metrics used in the Information Extraction field that limit the set retrieved. This is the case of the *precision@n* measure, which computes the *precision* after n items have been retrieved. In our case, we can use the *precision@3* to evaluate how many of the movies in gf_G are in the rgf_G set (note that $|rgf_G| = 3$). This kind of evaluation can be seen from a different point of view: we are usually interested in having at least one of the movies from gf_G in the rgf_G set. This measure is called *success@n* and returns 1 if there is at least one hit in the first n positions. Therefore, we could use *success*@3 to evaluate our system by computing the rate of recommendations where we have at least “one-hit” in the real group favorites list. For example, a 90% accuracy using *one-hit* means that the recommender suggests at least one correct movie for 90% of the groups evaluated. In fact, *success*@3 is equivalent to having *precision*@3 $> 1/3$. We can also define a “two-hits” metric (equivalent to *precision*@3 $> 2/3$), which represents how many times the estimated favorites list gf_G

contains at least two movies from rgf_G. Obviously, it is much more difficult to achieve high results using this second measure.

5.3. Trust Values

Next we compute the trust factor $t_{u,v}$ using the factors detailed in Section 3.2. More specifically, we use the following concrete trust factors. We must note that the thresholds and specific values have been experimentally obtained by analyzing the profiles of the users that took part in our experiment.[6]

—$f_1(u, v)$. Distance in the social network: 1.0 if direct friends, 0.5 if friend of a friend and 0.1, otherwise.

—$f_2(u, v)$. Number of common friends:

$$f_2(u,v) = \begin{cases} 1.0, & \text{if} >25 \text{ common friends} \\ 0.7, & \text{if} >15 \text{ common friends} \\ 0.5, & \text{if} >10 \text{ common friends} \\ 0.3, & \text{if} >5 \text{ common friends} \\ 0.1, & \text{if} <5 \text{ common friends.} \end{cases}$$

—$f_3(u, v)$. Intensity of the relationship: how often they write each other on their walls.

$$f_3(u,v) = \begin{cases} 1.0, & \text{everyday} \\ 0.7, & \text{5 times per week} \\ 0.5, & \text{every week} \\ 0.3, & \text{every month} \\ 0.1, & \text{never.} \end{cases}$$

—$f_4(u, v)$. Intimacy of the relationship: We classify relationships by finding keywords that represent the following intimacy levels: very affectionate, familiar/caring, friendly, work-related, casual, or none.

$$f_4(u,v) = \begin{cases} 1.0, & \text{Very affective} \\ 0.7, & \text{Carefree} \\ 0.5, & \text{Friendly} \\ 0.3, & \text{Professional} \\ 0.1, & \text{Distant.} \end{cases}$$

—$f_5(u, v)$. Duration: how long they have known each other.

$$f_5(u,v) = \begin{cases} 1.0, & >10 \text{ years} \\ 0.7, & >5 \text{ years} \\ 0.5, & >3 \text{ years} \\ 0.3, & >1 \text{ year} \\ 0.1, & <1 \text{ year.} \end{cases}$$

—$f_6(u, v)$. Reciprocal services: number of videos/songs/webs posted. In Facebook we also count the games/applications (like Pet Society, Mafia wars, Music challenge, among others) shared.

—$f_7(u, v)$. Structural variable: common interests described in the profile like movies, books, or general interests. From Tuenti we also count the places they go partying, and from Facebook how many groups they have joined or become a fan of.

[6]For example, if every pair of users with real $t_{u,v} = 1$ has more than 25 common friends, we choose that threshold for $f_2(u, v) = 1.0$.

—$f_8(u, v)$. Social distance: how many of the following properties are shared: political, educational, religious and demographical information.

$$f_8(u, v) = \begin{cases} 1.0, & >4 \text{ shared properties} \\ 0.7, & 3 \text{ shared properties} \\ 0.5, & 2 \text{ shared properties} \\ 0.3, & 1 \text{ shared properties} \\ 0.1, & \text{No shared properties.} \end{cases}$$

—$f_9(u, v)$. Status: this value depends on the kind of status: couple (1.0), family (0.7), best friends (0.5), friends (0.3), barely know (0.1).
—$f_{10}(u, v)$. Pictures: Percentage of pictures where they appear together.

The final trust value $t_{u,v}$ is a weighted average of the previous factors, as described in, Equation (4). These weights have been experimentally obtained using our group recommendation strategies by means of a genetic algorithm (GA). Our GA manages a population of vectors of weights (w_k). These vectors are combined and mutated in order to maximize the fitness function. We have used two different fitness functions according to the strategies described in Section 3.3: *delegation-based rating* prediction and *influence-based rating* prediction. Therefore, the individuals of the GA population (vectors of weights) are used to compute the trust factor $t_{u,v}$ required by these approaches. This fitness function is evaluated using the *one-hit* measure. This evaluation approach lets us explore a huge number of weights and obtain the best combination for each group recommendation algorithm. The optimal combination of weights found by the GA is shown in Section 6.

5.4. Implementation of the Recommender System

The construction of the recommender runs as follows. First, we build an individual movie recommender using the jCOLIBRI framework [Recio-García et al. 2008]. jCOLIBRI is currently a reference platform in the Case-Based Reasoning (CBR) community that facilities the design of different types of CBR applications, and it has a specific extension for developing recommender systems. The individual recommender follows the collaborative filtering approach described in Kelleher and Bridge [2004] based on the Pearson Correlation. It uses ratings extracted from the MovieLens database plus the ratings obtained from the second questionnaire completed by our volunteers. We select those MovieLens users who have rated at least 15 movies from the *Movies to rate* list and more than 3 from the *Movie listing*. The individual recommender returns a sorted list of movies that a user should watch individually. This list cif is a complete estimation of the users preferences for a given movie listing. However, in practice, only the first elements of this list should be taken into account because they are supposed to reflect the movies a user wants to watch (for example, we are not interested in the 12th movie the user prefers). Therefore, we select the three movies with the highest scoring and refer to this list as the individual favorites if_u of a user u. This if_u list can also be evaluated with the one-hit and two-hits measures previously detailed as, again, we are not interested in the exact order of these movies but are concerned with the inclusion of these movies in the set of movies selected for the group.

The next step is building the group recommender system. Each version of this recommender applies a different algorithm proposed in this article.

—*Base*. This is a standard group recommender using the average satisfaction aggregating function (Equation (1)). It is the baseline to measure the accuracy of our algorithms.
—*Personality*. It only uses the personality values and it implements the *Personality-based rating* prediction (PBR) approach (Equation (5)).
—*Trust-DBR*. It implements the *delegation-based rating* prediction (DBR) algorithm proposed in Equation (6), but it only takes into account the trust values $t_{u,v}$ (p_u values are set to 0).
—*Trust-IBR*. It implements the *influence-based rating* prediction (IBR) algorithm proposed in Equation (7). Again, this recommender only takes into account the trust values $t_{u,v}$ ($p_u = 0$).
—*TP-DBR*. This is the full Trust+Personality DBR algorithm (Equation (6)).
—*TP-IBR*. This is the full Trust+Personality IBR algorithm (Equation (7)).

The Personality, Trust-DBR and Trust-IBR recommenders let us explore the impact of both factors –personality and trust– independently. Then, the TP-DBR and TP-IBR are supposed to improve their results (as it is described in Section 6).

5.5. Experimenting with Memory

We also performed an experiment with a recommender system based on memory. We implemented this recommender by using the *dbrm* and the *ibrm* methods, described in Section 4. In these experiments, we assigned $\alpha = 1.0$ to simplify the comprehension of the results.

The evolution of user satisfaction over time is obtained by using the formula shown in Equation (10). We configured $\delta = 0$ in our case study to get a clearer picture of the impact of satisfaction. However, we plan to perform further experiments to analyze the consequences of this parameter.

To calculate user satisfaction we compare the gf_G list to the estimated individual favorite movies for that user. That is, on one hand we have the list of predicted movies that a user wants to watch individually, and on the other hand we have the list of movies recommended to the group. Therefore, we can compare both lists to estimate the users satisfaction.

Regarding the list of predicted movies that a user wants to watch individually we could use the if_u list. However this list only contains the first 3 movies with the highest scoring. If we compare this list to gf_G we get very low values of satisfaction (because the intersection between the two lists will probably be empty). To avoid this problem, we use the complete list of movies returned by the recommender (cif). This list contains the estimated scoring of every movie for a concrete user. As we have used a movie listing of 15 movies, it is the size of cif and it reflects the user preferences in order. We used this complete list because it enables us to evaluate the approach by means of the *Expected Search Length* (ESL) measure [Cooper 1968]. This measure assumes that the ordering of the target items is not relevant (it is a "weak ordered set") and it counts the number of nonrelevant items that a user would find until she finds a target number of relevant items. This is the case of a user when she observes the list of three movies proposed to the group (gf_G). Users expect that the movies recommended to the group will be their individual favorites; that is, they are as high as possible in her individual favorites list (cif). The user does not mind the order they appear in her individual favorites list but will be more satisfied when those movies are in the first positions of her list. And this behavior is captured by the ESL measure. Moreover, this measure has the advantage of combining precision and recall, which cannot be applied here as we have a fixed retrieval set.

The satisfaction value is obtained through Equation (11):

$$satisfaction(u, G) = \begin{cases} 1.0, & \text{if } esl(gf_G, cif_u) = 3; \\ 0.9, & \text{if } esl(gf_G, cif_u) \leq 4; \\ 0.8, & \text{if } esl(gf_G, cif_u) \leq 6; \\ 0.6, & \text{if } esl(gf_G, cif_u) \leq 8; \\ 0.4, & \text{if } esl(gf_G, cif_u) \leq 10; \\ 0.2, & \text{if } esl(gf_G, cif_u) \leq 12; \\ 0.0, & \text{if } esl(gf_G, cif_u) > 12, \end{cases} \tag{11}$$

where $esl(gf_G, cif_u)$ returns the position of the last element of gf_G that appears in cif_u. It is worth noting that in our experiments we consider that each user begins with the highest level of satisfaction ($s_u = 1.0$) and also that this satisfaction measure is different for each user with each of the groups that she belongs to. We store these measures separately as a user can be very pleased with the decisions of one group and, on the other hand, dissatisfied with the decisions of another group.

Our goal is to maximize average satisfaction ($\bar{S}$) while minimizing the differences in satisfaction levels within the group. These differences are computed by using the standard deviation (σ_S), which measures the uniformity of the satisfaction levels.

Therefore, the formula used to obtain the *global satisfaction* of the group ($gs(G)$) runs as follows:

$$gs(G) = \alpha \cdot \bar{S} - \beta \cdot \sigma_S, \tag{12}$$

where S represents the set of satisfaction levels of every user in G.

This equation includes two parameters (α and β) that could be used to weight the impact of both factors (average value and uniformity). Experiments are configured with these parameters set to 1, giving the same importance to the average level of satisfaction and the uniformity of these values. Our goal is to obtain very satisfied users (the best possible situation being an average of 1) and users who are equally satisfied (being the best situation a standard deviation of 0).

Finally, note that we store in our memory each movie chosen by the group from the proposed list (gf_G). If the recommender proposes a movie that has been previously chosen it is automatically rejected.

Once we have described the case study, the next section details the results obtained.

6. EXPERIMENTAL RESULTS

Group recommendation approaches presented in this article are based on an individual recommender. This individual recommender is only used as a basic building block for group recommenders so we are not interested in improving its performance. However, we must remark that any improvement in this system will lead to an improvement of the whole system, as the group recommender system is based on each users individual preferences. The individual recommender obtains a hit rate of 58,6% using the one-hit evaluation metric and 13,79% for the two-hits metric.

Figure 4 shows the performance of our group recommenders using the one-hit and two-hits evaluation metric. A chi-square test (p=0.05) was employed to test the accuracy of fit for each recommender. This test verifies whether the results obtained by our recommenders differ significantly from a randomly generated recommendation. It shows that the Base ($\chi^2 = 4.7115$, df = 1, p-value = 0.02996), Personality ($\chi^2 = 4.7115$, df = 1, p-value = 0.02996) and TP-DBR ($\chi^2 = 10.2671$, df = 1, p-value = 0.001354) recommenders obtain significant results. The results obtained by the TP-IBR recommender are weakly significant ($\chi^2 = 2.735$, df = 1, p-value = 0.09817, with p=0.1). However, the hit-rate obtained by both recommenders based only on trust –Trust-DBR

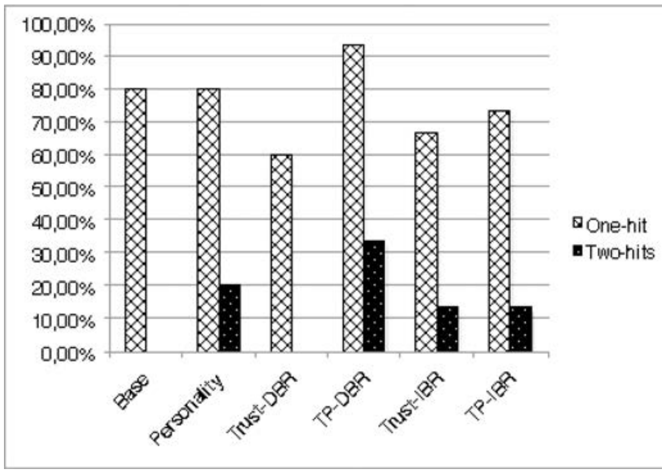


Fig. 4. Rate of hits of our six group recommender systems. The figure displays the hit rates for both one-hit and two-hits metrics.

($\chi^2 = 0.3846$, df = 1, p-value = 0.5351) and Trust-IBR ($\chi^2 = 1.2927$, df = 1, p-value = 0.2555)– does not differ significantly from a random recommendation.

The most promising results are achieved by the TP-DBR recommender. It obtains the highest hit rate not only with the one-hit metric but also with the two-hits. The recommender based on Personality gets similar results to the Base recommender with the one-hit metric but it obtains up to 20% with the two-hits metric. The worst results are obtained when considering only the trust factors during the recommendation (Trust-DBR and Trust-IBR). None of these recommenders beat the Base recommender in the one-hit metric, although the Trust-IBR approach has slightly better results with the two-hits metric. As we explained in Section 5.2, the two-hits evaluation metric is more difficult to achieve than the one-hit, and that is why we can see more abrupt changes in the two-hits than in the one-hit. The better the recommendation technique is, the higher the value of the two-hits measure is. The base recommender gets a 0% for the two-hits metric. The Trust-DBR algorithm does not improve the Base recommender. However, the Personality algorithm improves the Base, obtaining a 20%. The best recommender, TP-DBR, obtains 30% using the two-hits metric. The results obtained using Delegation-based approach differ from those obtained using the Influence-based approach. We can see that the Influence-based version works better than the Delegation-based one when taking into account just trust (Trust-IBR & Trust-DBR). However, when dealing with personality and trust it means a great improvement in the Delegation-based method, but not so much in the Influence-based method (TP-DBR & TP-IBR).

As stated in Section 5, we deal with groups of different size in order to evaluate the influence of the group size on the recommender. Figure 5 summarizes the results according to group size. Most of the recommenders achieve their best results with small groups (size=3) as it is easier to find good items for such kinds of groups. The worst results are obtained by the Trust-IBR recommender, with only a 60% hit rate using the one-hit metric. The results get worse as the group size increases for the Base and Trust-DBR recommenders. These three recommenders follow a monotonic decrease that is explained because they have proven to be the worst recommendation strategies. Therefore, as group size grows, and it gets harder to find a good recommendation, their behavior gets worse. However, we do not notice this trend in other strategies. The Personality recommender decreases the hit rate with medium-sized groups (size=5) but

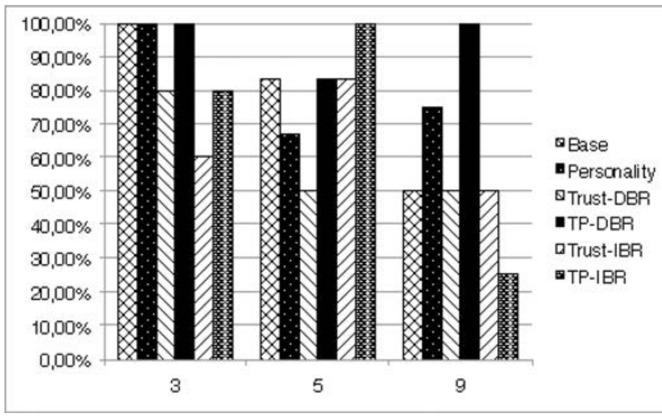


Fig. 5. Rate of hits for each recommender using the one-hit metric according to group size.

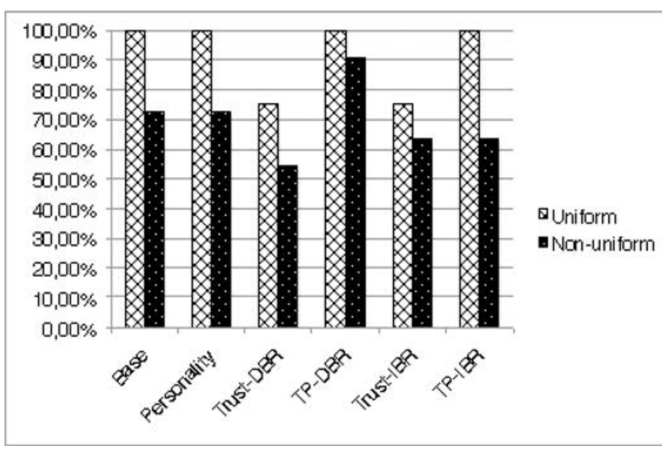


Fig. 6. Rate of hits for each recommender using the one-hit metric according to group homogeneity based on personalities.

it obtains similar results with large groups (size=9), around 70% accuracy. The TP-IBR achieves the best results for medium-sized groups but it goes down sharply with large groups, with only a 25% hit rate with the one-hit metric. These irregular behaviors should be studied in further experiments. However the most feasible explanation for the TP-IBR strategy is that taking too many influencers into account adds a lot of noise to the predicted ratings, although with a medium number of them this influence is a positive factor. On the other hand, the TP-DBR recommender is the most robust to changes in group size. It is one of the best recommenders for medium-sized groups (excluding the irregular Trust-IBR) and it remains the best with large groups, obtaining a 100% hit rate. As this strategy modifies personal predictions according to other users' preferences, we think it maximizes the average satisfaction of every member of the group, thus explaining its excellent performance.

The analysis of group homogeneity according to individual personalities is displayed in Figure 6. Homogeneity is evaluated by using the variance of personality (p_u) in the group. To study the impact of group homogeneity we have classified user groups into two categories: *uniform* and *nonuniform*. A uniform group has low variance, whereas a group with high variance is considered to be a nonuniform group. The threshold to

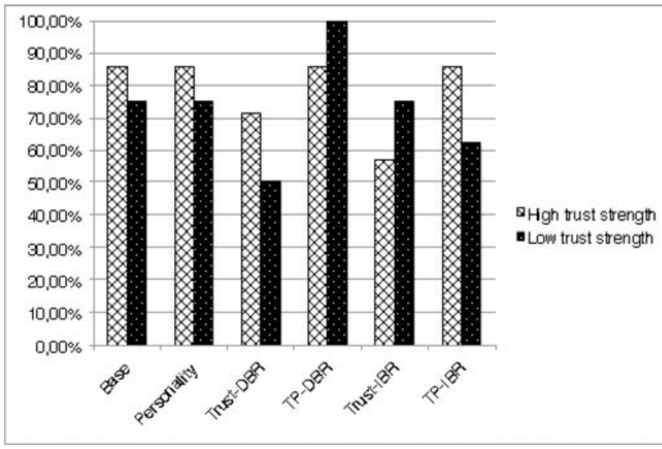


Fig. 7. Rate of hits for each recommender using the one-hit metric according to the group trust strength.

determine when variance is low or high was computed by using the median of variance for all the groups in the experiment.

The results stress that most of the recommenders work well with uniform groups. Again, the worst results are obtained by the recommenders based only on trust, with a 75% hit rate with the one-hit metric. It is worth highlighting that the best results with nonuniform groups are obtained by the TP-DBR recommender with a 90% hit rate. This is almost 20 points more than the Base and Personality recommenders and more than 25 points when compared to the recommenders using IBR. Finally, note that in uniform groups $p_u \approx p_v$; therefore, Equation (5) collapses to Base, which explains the high performance of the personality *pbr* method.

These results align with the ones described in Baltrunas et al. [2010], where the authors study the effects of group size and homogeneity on group recommendations. The authors state that the effectiveness of group recommendations does not necessarily decrease when the group size grows. Moreover, they confirm that the more alike the users in the group are, the more satisfied they are with the group recommendations.

Additionally, we analyzed the recommender hit rate in comparison with the global trust strength of the group. This trust strength is the average of the trust between each pair of group members. As we did with personalities, we employed the median of trust strength for every group that participated in the experiment as the threshold to split global trust strength into high trust strength and low trust strength. Figure 7 summarizes this analysis. Surprisingly, the worst results in groups with high strength are obtained with the recommenders based only on trust (Trust-DBR and Trust-IBR). It leads us to think that trust must be combined with personality to obtain a general improvement in accuracy and is not a significant factor by itself. Both DBR and IBR strategies are rooted in the idea that users may modify their preferences to please close friends. If so, Trust-DBR and Trust-IBR should achieve higher performance. These results show that this premise is not completely realistic, leading us to think that users could modify their preferences if and only if they do not have a selfish personality. That is the reason why methods mixing personality and trust report better results. When groups have low trust strength results get worse for most of the recommenders except for the TP-DBR and the Trust-IBR. The former again shows the best performance with a 100% hit rate in groups with low trust strength. With these results we conclude that the TP-DBR recommender shows the most robust behavior because it adapts to

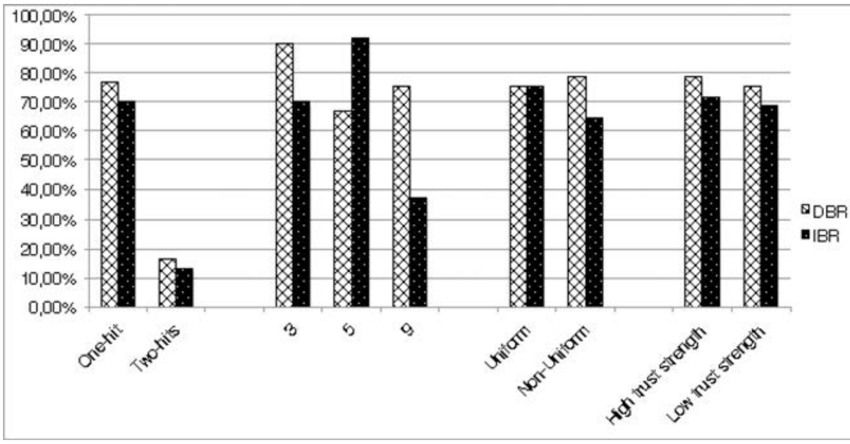


Fig. 8. Results obtained by DBR vs. IBR recommenders: Global rate of hits using one-hit and two-hits metrics, and rate of hits using the one-hit metric according to group size, homogeneity of personalities and trust strength.

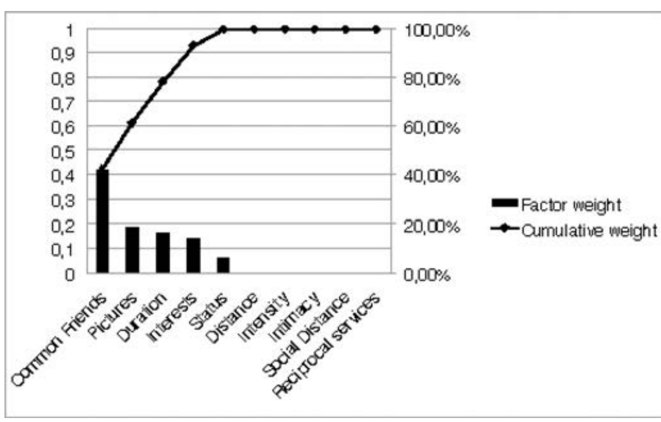


Fig. 9. Trust factor weights computed by the GA using the DBR method.

different group configurations very well, getting high hit rates in all scenarios. However controversial results require further future work.

Finally, we have compared the results obtained by the two algorithms proposed in Section 3.3 that combine the personality values and the trust factors of group members: DBR and IBR. Figure 8 shows that recommenders implemented using DBR obtain higher hit rates than recommenders based on the IBR approach. Only with medium-sized groups do the IBR recommenders get better results. Again, we think this occurs because a medium number of influences is a positive factor, but having too many people influencing other's preferences is counterproductive. DBR does not show this irregular behavior as it tries to maximize the average satisfaction within the group. In conclusion, we can assume that the DBR approach is more robust for different group configurations than IBR.

Regarding the importance of the factors that conform the trust value, we can see in Figure 9 how they are taken into account in order to maximize the performance of our recommender. These weights (w_k) are obtained by using the Genetic Algorithm (GA) described in Section 5.3. The most relevant factor is the number of friends in

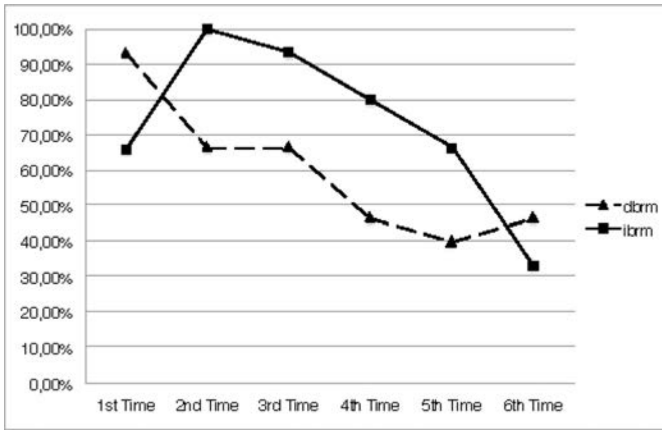


Fig. 10. Evolution of the rate of hits using the one-hit metric for the group recommender implementing the *ibrm* and *dbrm* algorithms.

common, followed by the pictures, duration, common interests and status. These five factors almost accumulate 100% of weight (see the cumulative weight in Figure 9) when computing trust whereas the remaining factors are mostly insignificant. This conclusion is supported by the diagram of the top predictive variables that was presented by Gilbert and Karahalios [2009] where they show the predictive power of the different tie strength dimensions and the top three predictive variables for each dimension. The percentages that they achieve are: 32.8% for the Privacy dimension (the number of mutual friends was one of the top three predictive variables for this dimension), 19.7 % for the Intensity dimension (the number of pictures in common represents this dimension) and in third position, duration with 16.5%.

These weights are the optimal values found for the DBR method. On the other hand, when we perform the study of the optimal weights for the IBR method, we discover that the best one is f_{10} (common pictures) with 0.97% significance. We have repeated the experiment after modifying the parameters of the GA (population size, repetitions, etc) to confirm this surprising result and this value always came up high. We can explain it because this factor (the percentage of pictures in common) can summarize many other factors: if two people appear together in a lot of pictures it usually means that they have had a long relationship (duration), they are very close (status), they go together to the same places (interests), etc.

6.1. Results of the Recommenders with Memory

In this section we detail the results obtained when including memory capabilities in the recommender systems. During these experiments we execute the proposed algorithms several times and we not only measure accuracy using the *one-hit* metric but also the global satisfaction of the groups as described in Section 5.5.

In the experiments with memory-boosted recommenders we are not interested in studying the personality and trust factors independently. The results of the recommenders without memory show that the combination of both factors achieves the best performance. Therefore, it makes no sense to study the impact of personality and trust separately in recommenders with memory. Evaluation is again performed with the one-hit measure.

Figure 10 shows the accuracy results for both algorithms, *ibrm* and *dbrm*, using the one-hit metric. The *ibmr* algorithm has a very good initial performance. However, as long as we repeat its execution, the proposed recommendations are worse; that is, the

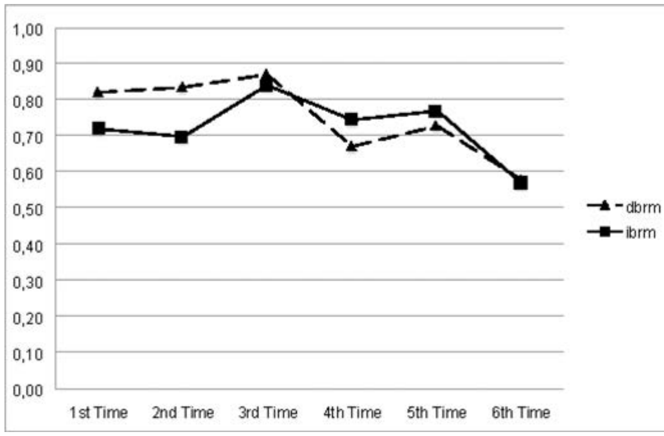


Fig. 11. Evolution of group satisfaction *gs* with recommendations provided by the group recommender implementing the *ibrm* and *dbrm* algorithms.

hit rate of the recommender goes down. The *dbmr* follows a similar behavior. This situation is coherent because we are giving priority to the long-term satisfaction of everyone in the group when proposing a movie, instead of maximizing the individual preferences for that particular movie. Here, the algorithm tries to take into account the opinions of dissatisfied users, although their preferred movies are not a good recommendation for the group as a whole.

According to the global satisfaction of the group, we can observe in Figure 11 the evolution of the *gs* function (Equation (12)). In both algorithms global satisfaction tends to rise (or, at least, does not decrease) when we perform the few initial recommendation to the group. Therefore, this experimental result prove our hypothesis that a memory of past recommendations can improve user satisfaction. We observe that satisfaction tends to decrease after the 4th recommendation. The reason is that during the experiments we do not change the movie listing. This way, the more recommendations we make, the less interesting movies we leave for the next recommendation (individually or in groups). This problem could be fixed by changing the movie listing after several recommendations, an experiment that we intent to perform soon.

We also evaluated the option of penalizing those users who were especially satisfied with one recommendation, so that their opinion would have less priority. To achieve this, we simply modified the $satisfaction(u, G)$ formula (Equation (11)) to return a value higher than 1 when the individual favorites of the user if_u were in the very first positions of gf_G (for example, returning 1.2 if gf_G are within the first 3 movies in if_u). The results obtained are very similar to the previous ones, so we can conclude that penalizing the users that were especially satisfied with one recommendation does not suppose a relevant change.

7. CONCLUSIONS

In this article we have introduced a novel method for making recommendations for groups based on existing techniques of collaborative filtering, and taking into account group personality composition and the structure of the group itself. Once we showed that personality profiles can improve a recommendation for a group of people [Recio-García et al. 2009], we extended this approach by reflecting the trust relationships between the users in the group in a more realistic way. We propose two approaches, named Influence-Based Rating and Delegation-Based Rating that improve existing techniques by including both the personality and social trust factors in the group recommendation process.

The trust level is measured by using a set of social factors extracted from groups of people connected through social networks. Examples of these social factors are distance wihin the social network; the number of mutual friends, the intensity, intimacy, and duration of the relationship; social distance; and the percentage of pictures where they appear together, among others.

The trust value is computed for every pair of group members by using a weighted average of the social factors. These weights have been experimentally obtained for every group recommendation strategy by means of a genetic algorithm. It gives us a collateral result: the identification of the most relevant social factor for recommendation to groups. The most relevant factor is the number of friends in common, followed by pictures, duration, common interests and status.

Finally, we propose the inclusion of a memory of past recommendations to increase user satisfaction. The use of this memory lets the system increase the importance of the preferences of the most dissatisfied users in previous recommendations. This way, we keep a uniform level of satisfaction for all the users during several recommendations.

We performed an experiment in the movie recommendation domain, using data from MovieLens, and we studied the accuracy of different recommendation approaches for different group compositions. In our experiment we employed 58 real users who participated in two events in two different social networks, Facebook and Tuenti.

We conclude that including only the personality factor in the recommendation process improves accuracy, but using only trust values between the group members does not yield a significant improvement. However, when combining both personality and trust we achieve the best results. We think that by using just the personality factor or just the trust factor we do not have a complete representation of human behavior; we do not have enough information, and thus, we do not get the results that take place in the real world. That is why the recommenders that use these factors in an isolated way do not render a representative difference when comparing their results with the base recommender. However when we combine both factors we manage to represent how discussions take place in real life and how some opinions have more weight than others; that is why the approaches with the combination of the personality and trust improve the base recommender. Delegation-Based Rating has yielded the best approach. With the one-hit evaluation metric we get a rate of hits that is 13.3% higher than the hit rate for the basic group recommender. Moreover, the hardest evaluation measure, the two-hits metric, returns an accuracy of 35% whereas the basic recommender has 0% success in this measure.

We have also studied several features of group composition to measure their impact on the accuracy of the group recommender. According to the influence of group size, the conclusion is that all the approaches work better with smaller groups. It is an understandable fact because with more people there are more diverse preferences and personalities, thus it is more difficult to arrive at a consensual solution. The homogeneity of group personalities is also an important factor as uniform groups get better results. Finally, we studied the impact of trust strength within the group. These results vary depending on the algorithm chosen, although the Delegation-Based Rating method again achieves one of the best results. In general, we can conclude that the Delegation-Based Rating method is the best option for different group compositions as it gets good results for different group sizes and homogeneity levels.

Regarding the inclusion of a memory of past recommendations, the results prove that this feature improves the global satisfaction of users after several recommendations. However, this memory has a negative impact on the performance of the recommender because it suggests items that fit the requirements of very dissatisfied users, instead of proposing the best items for the group.

In summary, we propose several algorithms to recommend items to groups with different sizes and personal preferences. We have proven that by introducing the trust factor, personality awareness and a memory of past recommendations we can improve system performance. Although our approach has been tested in the movie field, it is not specific to this domain and the proposed algorithms could be directly applied to other domains, like choosing restaurants or planning trips.

Our approach can be exploited to develop recommender applications for groups in any social network. According to Facebook's latest statistics,[7] there are 500 million active users and the average user is connected to 80 community pages, groups and events. Therefore, it represents an impressive number of potential users and applications that could take advantage of the methods detailed in the article.

8. FUTURE WORK

In this work we have employed average satisfaction as the aggregation function to generate recommendations for groups. However there are other aggregation functions that can be employed in our recommender. We plan to evaluate the impact of these aggregation functions on the accuracy of our approach.

We studied the behavior of the recommender according to the group size. Moreover, we include an evaluation of the recommendation process in comparison with the homogeneity of the personalities within the group. However, we could also evaluate the system against group homogeneity in terms of their movie preferences. To do so, we need an estimation of the similarity between users that we could obtain from the Pearson correlation. Additionally, further evaluations are needed in order to corroborate the preliminary results obtained in this work.

Once we analyzed the behavior or recommender according to group size and personalty/preference homogeneity, we were able to combine our results with the ideas proposed in Amer-Yahia et al. [2009] to propose adaptive recommenders, where the recommendation algorithm adapts itself to the personality distribution in the group, its trust strength and its size.

In addition to the trust and personality of group members, there are another factors that could be taken into account during the recommendation process. Cultural aspects or the context where the recommendation is performed may have an impact on the recommendation process. For this reason, we plan to work on how these aspects can influence the rating prediction similarly to the way that we used trust and personality factors, and to evaluate the impact of these new aspects on the group recommendation.

Although the memory of previous recommendations improves user satisfaction, the accuracy of our experiments decreases due to the use of a fixed movie listing. Experiments should be repeated with movie listings that are updated for every recommendation. Additionally we plan to perform new studies to analyze the impact of the "memory parameter" δ from Equation (10) and reflect several degrees of previous satisfaction.

ACKNOWLEDGMENT

We also thank the friends and students that participated in the experiment.

REFERENCES

Aamodt, A. and Plaza, E. 1994. Case-based reasoning: Foundational issues, methodological variants, and system approaches. *Artif. Intell. Commun.* 7, 1, 39–59.

[7] http://www.facebook.com/press/info.php?statistics.

AMER-YAHIA, S., ROY, S. B., CHAWLAT, A., DAS, G., AND YU, C. 2009. Group recommendation: semantics and efficiency. *Proc. VLDB Endow. 2*, 1, 754–765.

ARDISSONO, L., GOY, A., PETRONE, G., SEGNAN, M., AND TORASSO, P. 2003. Intrigue: Personalized recommendation of tourist attractions for desktop and handset devices. *Appl. Artif. Intell. 17*, 8, 687–714.

BACCIGALUPO, C. AND PLAZA, E. 2007. A case-based song scheduler for group customised radio. In *Proceedings of the 7th International Conference on Case-Based Reasoning (ICCBR '07)*. R. Weber and M. M. Richter, Eds., Lecture Notes in Computer Science, vol. 4626, Springer, 433–448.

BALTRUNAS, L., MAKCINSKAS, T., AND RICCI, F. 2010. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, 119–126.

BARSADE, S. G. 2002. The ripple effect: Emotional contagion and its influence on group behavior. *Admin. Sci. Quart. 47*, 4, 644–675.

BERKOVSKY, S. AND FREYNE, J. 2010. Group-based recipe recommendations: analysis of data aggregation strategies. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, 111–118.

BOBADILLA, J., SERRADILLA, F., AND HERNANDO, A. 2009. Collaborative filtering adapted to recommender systems of e-learning. *Knowl.-Based Syst. 22*, 4, 261–265.

CHEN, Y.-L., CHENG, L.-C., AND CHUANG, C.-N. 2008. A group recommendation system with consideration of interactions among group members. *Expert Syst. Appl. 34*, 3, 2082–2090.

COOPER, W. S. 1968. Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *Amer. Document. 19*, 1, 30–41.

CROSSEN, A., BUDZIK, J., AND HAMMOND, K. J. 2002. Flytrap: intelligent group music recommendation. In *Proceedings of the 7th International Conference on Intelligent User Interfaces (IUI '02)*. ACM, 184–185.

DEUTSCH, M. AND GERARD, H. B. 1955. A study of normative and informational social influences upon individual judgement. *J. Abnormal Social Psych. 51*, 3, 629–36.

GILBERT, E. AND KARAHALIOS, K. 2009. Predicting tie strength with social media. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI '09)*. ACM, 211–220.

GOLBECK, J. 2006a. Combining provenance with trust in social networks for semantic web content filtering. In *Provenance and Annotation of Data, International Provenance and Annotation Workshop, Revised Selected Papers*, L. Moreau and I. T. Foster, Eds., Lecture Notes in Computer Science, vol. 4145, Springer, 101–108.

GOLBECK, J. 2006b. Generating predictive movie recommendations from trust in social networks. In *Proceedings of the 4th International Conference on Trust Management*. 93–104.

GRANOVETTER, M. 1973. The strength of weak ties. *Amer. J. Sociology 78*, 6, 1360–1380.

HATFIELD, E., CACIOPPO, J., AND RAPSON, R. 1994. *Emotional Contagion. Studies in Emotion and Social Interaction*. Cambridge University Press.

HERLOCKER, J. L., KONSTAN, J. A., TERVEEN, L. G., AND RIEDL, J. T. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst. 22*, 1, 5–53.

JAMESON, A. 2004. More than the sum of its members: challenges for group recommender systems. In *Proceedings of the Working Conference on Advanced visual interfaces (AVI '04)*. ACM, 48–54.

JAMESON, A. AND SMYTH, B. 2007. Recommendation to groups. In *The Adaptive Web, Methods and Strategies of Web Personalization*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., Lecture Notes in Computer Science, vol. 4321, Springer, 596–627.

KELLEHER, J. AND BRIDGE, D. G. 2004. An accurate and scalable collaborative recommender. *Artif. Intell. Rev. 21*, 3-4, 193–213.

KIM, J. K., KIM, H. K., OH, H. Y., AND RYU, Y. U. 2010. A group recommendation system for online communities. *Int. J. Inf. Manage. 30*, 3, 212–219.

LEVIN, D. Z., CROSS, R., AND ABRAMS, L. C. 2004. The strength of weak ties you can trust: the mediating role of trust in effective knowledge transfer. *Manage. Sci. 50*, 1477–1490.

LIEBERMAN, H., DYKE, N. W. V., AND VIVACQUA, A. S. 1999. Let's browse: A collaborative web browsing agent. In *Proceedings of the International Conference on Intelligent User Interfaces.* 65–68.

LINDEN, G., SMITH, B., AND YORK, J. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput. 7*, 76–80.

MASTHOFF, J. 2004. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Model. User-Adapt. Interact. 14*, 1, 37–85.

MASTHOFF, J. AND GATT, A. 2006. In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. *User Model. User-Adapt. Interact. 16*, 3–4, 281–319.

McCarthy, J. F. 2002. Pocket restaurant finder: A situated recommender systems for groups. In *Proceedings of Workshop on Mobile Ad-Hoc Communication at the ACM Conference on Human Factors in Computer Systems*.

McCarthy, J. F. and Anagnost, T. D. 1998. MusicFX: An arbiter of group preferences for computer aupported collaborative workouts. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW '98)*. ACM, 363–372.

McCarthy, K., McGinty, L., Smyth, B., and Salamó, M. 2006. The needs of the many: A case-based group recommender system. *Adv. Case-Based Reas. 4106*, 196–210.

O'Connor, M., Cosley, D., Konstan, J. A., and Riedl, J. 2001. Polylens: a recommender system for groups of users. In *Proceedings of the European Conference on Computer Supported Cooperative Work (ECSCW'01)*. Kluwer Academic Publishers, Norwell, MA, 199–218.

O'Donovan, J. and Smyth, B. 2005. Trust in recommender systems. In *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI '05)*. ACM, 167–174.

Recio-García, J. A., Díaz-Agudo, B., and González-Calero, P. A. 2008. Prototyping Recommender Systems in jCOLIBRI. In *Proceedings of the ACM Conference on Recommender syStems (RecSys '08)*. ACM, New York, NY, 243–250.

Recio-García, J. A., Jimenez-Diaz, G., Sánchez-Ruiz, A. A., and Díaz-Agudo, B. 2009. Personality aware recommendations to groups. In *Proceedings of the ACM Conference on Recommender Systems*. L. D. Bergman, A. Tuzhilin, R. D. Burke, A. Felfernig, and L. Schmidt-Thieme, Eds., ACM, 325–328.

Sarwar, B., Karypis, G., Konstan, J., and Reidl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. ACM, New York, NY, 285–295.

Schafer, J., Frankowski, D., Herlocker, J., and Sen, S. 2007. Collaborative filtering recommender systems. In *The Adaptive Web*, Lecture Notes in Computer Science, vol. 4321, Springer, 291–324.

Sinha, R. R. and Swearingen, K. 2001. Comparing recommendations made by online systems and friends. In *Proceedings of the DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*.

Thomas, K. and Kilmann, R. 1974. *Thomas-Kilmann Conflict Mode Instrument*. Tuxedo, NY.

Victor, P., Cornelis, C., Cock, M. D., and Teredesai, A. 2008. Key figure impact in trust-enhanced recommender systems. *AI Com. 21*, 2-3, 127–143.

Wu, A., DiMicco, J. M., and Millen, D. R. 2010. Detecting professional versus personal closeness using an enterprise social network site. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, 1955–1964.

Yu, Z., Zhou, X., Hao, Y., and Gu, J. 2006. TV program recommendation for multiple viewers based on user profile merging. *User Model. User-Adapt. Interact. 16*, 1, 63–82.

Received January 2011; revised April 2011, July 2011; accepted July 2011

Chapter 19

Including Social Factors in an Argumentative Model for Group Decision Support Systems

19.1 Citation

Recio-García, J. A., Quijano-Sánchez, L., Díaz-Agudo, B. Including Social Factors in an Argumentative Model for Group Decision Support Systems. Decision Support Systems, DSS, 56, pages: 48-55, Elsevier, 2013. ISSN: 0167-9236. JCR (2013): 2.036 (Computer Science, Artificial Intelligence, 27 out of 121, Q1. Computer Science, Information Systems, 20 out of 135, Q1. Operations Research and Management, 9 of 79, Q1).

19.2 Contributions covered by this paper

In this paper we have proposed a new technique of group recommendation based in social factors using a system where each user delegates to an agent that represents her preferences and argues with other agents to reach a consensus. This technique includes the analysis of users' social factors, personality and trust, along with a negotiation system, which is based in a multi-agent architecture that represents the social connections within the group.

Decision Support Systems 56 (2013) 48–55

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Including social factors in an argumentative model for Group Decision Support Systems☆

Juan A. Recio-García *, Lara Quijano, Belen Díaz-Agudo

School of Computing, Complutense University of Madrid, Spain

ARTICLE INFO

Article history:
Received 20 January 2012
Received in revised form 28 November 2012
Accepted 9 May 2013
Available online 27 May 2013

Keywords:
Decision Support Systems
Negotiation Support Systems
Personality
Social networks
Trust
Multi-agent systems

ABSTRACT

In this paper we propose a Decision Support System for groups of people where each user delegates to an agent that represents her preferences and argues with other agents to obtain the best alternative for the whole group. The novelties of our approach are the inclusion of users' social factors, personality and trust, in the argumentation process and the negotiation system, plus a multi-agent architecture that represents the social connections within the group. Therefore, our model simulates the argumentations made by real users to agree on a concrete product in a very accurate way. As a case study, we have tested our theories in the movie recommendation domain with real social networks. We have concluded that distributed models and argumentation techniques including personality and social trust improve the satisfaction of users involved in a group decision making process.

1. Introduction

A decision is a choice among alternatives based on estimated values for these alternatives. Supporting a decision means helping people to work alone or in groups to gather information, generate alternatives and make decisions. A decision making process also involves the estimation, evaluation and comparison of alternatives. Our work presented in this paper consists of providing a new method to evaluate users' estimation of different products and supporting decision making processes by providing a group recommendation for these products.

Our goal is to get an accurate reproduction of the decision making processes run by real groups of people when deciding leisure activities. There are some types of items, like restaurants, movies or trips that people tend to enjoy together. These kinds of items have a very relevant commercial interest, so it is kind of a natural thing to make the most accurate recommendations to groups of people. Existing approaches on *Group Decision Support Systems* (GDSS) are typically based on the aggregation of the preferences of group members, where every person in the group is considered equal to the others [7,18]. Other group decision approaches have solved the conflict by trying to maximize the preferences of the greatest number of group members [20]. But none of these approaches have into account that different groups of people have very different characteristics that strongly affect the decision process: size, social strength and influence between group members, personal preferences, personality of the group members, etc. It is a fact that when we face a situation where people's concerns don't match, conflict arises. Therefore, the general satisfaction of the group is not always the aggregation of the satisfaction of its members, as different people have different expectations and behavior in conflict situations. This fact is taken into account in recent works that agree on the need to adapt the decision making process to the group composition [15,19]. Furthermore, it is also well-known that user preferences can be affected by the rest of the group [6,19].

Our recent work [24] involves the improvement of current group recommendation techniques by introducing two novel factors: the *personality* of each individual and the *trust* among users. We have also presented some experiments where we test our theories for recommending products to groups of people connected through social network structures. In our model, we support the process of decision making by taking into account the group personality composition and the social connections among the individuals of the group. Once the relevance of these factors has been validated, in this paper we propose integrating them into a novel approach for group decision making based on a multi-agent system that accurately reproduces real argumentation processes made by real users. In the network of agents every agent should be able to define the trustworthiness regarding the connected agents [12,14] and to reflect the personality of the user it is representing. Our model is based on the idea of taking into account the social connections of the collaborative agents, including the level of trust of the agent they collaborate with [11,21,32].

Therefore, this paper presents a software architecture where each user delegates to an agent that represents her in the argumentation process.

☆ Supported by the Spanish Ministry of Science & Education (TIN2009-13692-C03-03), Madrid Education Council UCM (Group 910494), and the Spanish Ministry of Economy and Competitiveness (IPT-2011-1890-430000).

* Corresponding author. +34 91 3947560.
E-mail addresses: jareciog@fdi.ucm.es (J.A. Recio-García), lara.quijano@fdi.ucm.es (L. Quijano), belend@sip.ucm.es (B. Díaz-Agudo).

0167-9236/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.dss.2013.05.007

This way, users are freed from holding the typical annoying discussions to agree on a common decision for the group [30]. Another relevant advantage of this architecture is its perfect integration into existing social networks like Facebook or Google+.

Finally, we describe a case study for a collaborative movie recommender system and we present the results of an experiment where we measure the accuracy of the system results using argumentation protocols and a network topology based on a real social network.

Summarizing, this paper presents our research on GDSS by reproducing the real social organization of the group and including deliberation capabilities. Our main contribution is to improve group decisions by moving to a distributed model with social network topologies, introducing social factors, like personality and trust, plus an argumentation process that enables users to argue and defend their opinions by means of delegation to agents.

The paper runs as follows: Section 2 presents our approach for distributed GDSS and how to obtain the personality and trust factors to be integrated in this kind of systems. The distributed models and argumentation processes are explained in Section 3. Section 4 describes a case study of our method in the movie recommendation domain using data extracted from real social networks. Section 5 presents the results of our experiments. Finally, the conclusions and main lines of future work are explained in Section 6.

2. Distributed Group Decision Support Systems

Our approach to solving decision support problems is rooted in the Case-Based Reasoning field [1]. Case-Based Reasoning (CBR) is based on the intuition that similar problems tend to recur. It means that new problems are often similar to previously encountered problems and, therefore, that past solutions may be of use in the current situation [17]. When a CBR system faces the resolution of a new problem, it will search in its case base for problems similar to the current one. Once it finds them, these previous cases will be adapted to the current problem in order to provide a valid answer. The analogies between CBR and *Decision Support Systems* (DSS) are manifold. In DSS users manage a memory of preferences that must be similar to the alternative chosen by the group. Once the best alternative is obtained it is proposed directly to the user without requiring adaptation. Moreover, both techniques pay significant attention to the learning processes that improve system performance by taking into account user knowledge (the preferences or experiences of the users).

Brehmer [5] describes a distributed decision making system as an environment that (a) enables cooperation from a number of decision makers, where (b) each decision maker owns part of the resources needed to solve the problem; and (c) no decision maker has a complete overview of the problem as a whole. This schema fits perfectly in several works in distributed CBR that assume multi-case-base architectures involving multiple processing agents differing in their working memory [22]. In this kind of systems, each agent manages its own memory of preferences that make up its partial view of the world. To solve a given problem, this knowledge must be shared to obtain a solution for the whole group. This way we can reuse the existing research on distributed CBR in the GDSS field. CBR literature proposes several ways to combine different experiences to obtain improved solutions in distributed architectures. One important methodology is the *ensemble effect*, explained in [23], which proves that the argumentation of two agents improves the results obtained by one only agent working with the same experiences. This conclusion was the precursor of a research line focused on finding the best argumentation protocols to allow CBR agents to discuss a common problem. In [23] they came up with the AMAL protocol, which enables several CBR agents to argue about a common problem by means of arguments and counterarguments. This protocol and its adaptation to our model are presented in Section 3.

Setting aside the distributed architecture, when moving from individual to GDSS, the main issue that arises is how to find an alternative that satisfies the greatest number of group members, while taking into account the preferences of the decision makers. Several GDSS propose the generation of an aggregated preference built with individual user preferences [4,31]. However, our approach to group decision support is completely different because it simulates the argumentation process of a group of users by using a distributed architecture instead of providing an aggregated estimation. This way, we try to reproduce – in an accurate way – the real argumentation process carried out by decision makers when reaching an agreement. Moreover, to reproduce these argumentations accurately in our model we include two factors that reflect the real (or *human*) behavior of users. These factors – described in following subsections – are personality and social trust.

2.1. Personality estimation for Group Decision Support Systems

Usually, works in GDSS consider the preferences of every member of the group to have the same degree of importance and try to satisfy the preferences of every group member. However, groups of people can have very different characteristics and can be made of people with similar or antagonistic personal preferences. It is a fact that when we face a situation in which the concerns of people appear to be incompatible a *conflict situation* arises.

Our approach determines that the general satisfaction of the group is not always the aggregation of the satisfaction of its members, as different people have different expectations and behavior in conflict situations that should be taken into account. In [24] we presented a method for group decision support where we distinguish between different types of individuals in a group. Our research characterizes people using the Thomas–Kilmann Conflict Mode Instrument (TKI) [29]. From the answers to the TKI test we compute a value $p_u \in [0,1]$ that represents the personality as user u; 0 being the reflection of a very cooperative person and 1 the reflection of a very selfish one. Our method takes this value into account by studying how group personality composition influences the decision making process for the group, and how performance is improved for certain types of groups when compared to different simple group preference aggregation algorithms.

In this paper we present some experiments where we include the impact that personality will have on the argumentation process when two users u and v are arguing. This factor is computed as the personality difference:

$$\Delta p_{u,v} = p_u - p_v$$

where p_u and p_v are the values that reflect the personality of users u and v respectively. Note that $\Delta p_{u,v} \in [-1,1]$.

As we detail in Section 3, we propose to use the personality difference value to configure the behavior of each agent in the distributed architecture. This factor will be integrated into the group decision making process together with another feature: trust among agents. This second factor is detailed next.

2.2. Social trust and network topologies in GDSS

In today's networked worlds, uncertainty and anonymity are important factors that have strong implications in decision-making. Several researchers have therefore proposed to incorporate the concept of interpersonal trust in Group Decision Support Systems [27,28,32]. This factor is even more important when we are performing a group decision making process where users have to agree on an alternative for the whole group. This kind of process usually follows an argumentation schema where each user defends her preferences and rebuts others' opinions. Here, trust among users is the major factor when users must change their mind to reach a common decision.

A promising approach is to collect trust knowledge from existing social networks like Facebook, Twitter, Google+, among others.

The use of social networks and trust when building a DSS system is not new. Generally, trust is employed as a way to give more weight to some users, to compute users' similarity, or to sort and filter the alternatives by giving priority to trusted sources. It has been employed in different domains like movie recommendation, e-mail filtering [9] or ski mountaineering [2].

In our approach, we propose the use of social network topologies to reflect the interactions of the users within the group. We think that, by reproducing this structure, the decision making process will be a realistic reproduction of the argumentations that take place in a group and, consequently, the results will be more accurate. This organization of the users can easily be obtained from current social networks or built ad-hoc for a GDSS application. However, a very promising option is to include the system in the social network as it will easily exploit the information in the network [26]. That is one of the reasons for the use of a multi-agent architecture where each agent is linked to the user profile.

Our working hypothesis is that this new organization of the structure of the group will affect and improve the result of the decision making process, mainly because with the social network topology we give a more realistic structure and organization to the group, which is closer to how the argumentations would take place in a real group when they argue about which alternative to choose. For a given group of users, we sketch a network where each node represents a person and each connection represents that the particular person has a relation with the one he is connected to. If two nodes are not connected, it means that the people they are representing don't know each other or that they are not close. When each node generates its preferences, it will consider only the information provided by the nodes that it is connected to, representing those that could have influenced its decision in real life. In this way we use social network topologies to evaluate social trust, as it is reflected in our trust function, explained in Section 4.

Moreover, we have studied what the most important factors are in the social networks that must be taken into account when computing trust between users. Examples of these factors are: the number of shared messages, common pictures, direct friends, etc. To perform this task we have reviewed several existing works [8,10] and selected the most relevant and feasible factors. We have chosen 10 factors that are combined to get a final trust value. Furthermore, we have evaluated which factors have the highest impact in the decision making process. The specific trust factors are:

- $f_1(u,v)$: Distance in the social network.
- $f_2(u,v)$: Number of common friends.
- $f_3(u,v)$: Intensity of the relationship: how often they write each other on their walls.
- $f_4(u,v)$: Intimacy of the relationship: We classify relationships by finding keywords that represent different intimacy levels.
- $f_5(u,v)$: Duration: how long they know each other.
- $f_6(u,v)$: Reciprocal services: number of posted videos/songs/webs, shared games/applications.
- $f_7(u,v)$: Structural variable: common interests described in the users' profile like movies, books, or general interests.
- $f_8(u,v)$: Social distance: how many of the following properties are shared: political, educational, religious and demographical information.
- $f_9(u,v)$: Status: value depending on the kind of status: couple, family, best friends, etc.
- $f_{10}(u,v)$: Pictures: percentage of pictures where they appear together.

The final trust value $t_{u,v}$ is a weighted average of the previously described factors:

$$t_{u,v} = \sum_{i=1}^{10} \alpha_i \cdot f_i(u,v). \quad (1)$$

Note that $t_{u,v} \in [0,1]$. We have measured the importance of every factor α_i by using an experimental approach. Results are reported in Section 4.

In the following section we describe the distributed model to carry out our group decision making theories.

3. Distributed argumentative model for Group Decision Making processes

Up to this point we have described the two social factors of our approach, the one related to personality and the one related to social networks and trust among group members. Now we are going to introduce the distributed architecture that imitates the social network connections for the group decision making process. The main goal is to improve decision making by taking into account the friendship topology (who is whose friend), the group personality composition and trust between group members. To include these three features in our model we define a multi-agent architecture following a social network topology, where each agent represents a member of the group. One of the main advantages of using distributed models is that the agents representing members of the group do not necessarily have to be stored in the same machine. This way each user can have one agent representing her preferences and arguing for her best interests in her computer.

In our architecture each agent discusses with all its neighbors in the network and the final decision will depend on the personality and trust the users being represented have with each other. When two agents are not connected, it means that the users they are representing don't know each other or they're not close. Therefore there won't be any kind of argumentation between them. However if two agents are connected, associated users that have some level of interpersonal trust, our model simulates a face to face discussion. This model improves the typical "fully connected" network topologies where every agent debates with every agent. We think that the "fully connected" topology is an artificial representation of the group and does not reflect real interactions among users. Fig. 1 shows the differences between the two alternatives in the topology of the network, the "fully connected" network topology and the social network topology.

Once we have described the distributed model to be built we need the infrastructure to implement it. We use D^2*ISCO* to implement the process of argumentation and the distributed architecture. This framework is described next in Section 3.1. It uses a reasoning protocol (fully detailed in Section 3.2) that begins with an agent issuing a query to the agents that it is linked to. Each agent will provide its individual local solution for the problem, this is, its favorite alternative for a given query. At each round the agents can rebut the solution made by its neighbor agents. These different counterexamples will have different weights depending on the personality and trust among the people who are being represented by the agents.

Finally they provide a consensual solution which will be the individual local solution of the agent that threw the query to them. This process keeps going backwards until it reaches the agent that threw the initial query. Next we will detail the argumentation processes between agents in D^2*ISCO*.

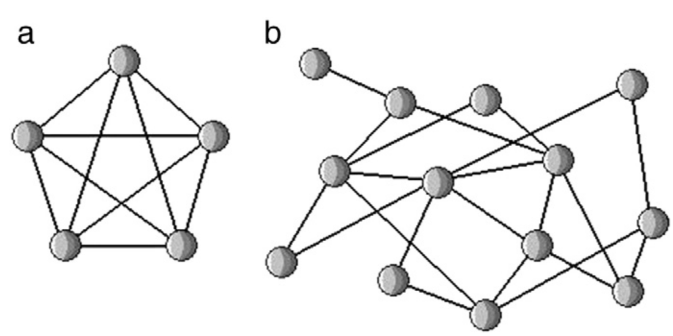


Fig. 1. a) "Fully connected" network topology. b) Social network topology.

3.1. D^2ISCO: distributed reasoning for collective experiences

D^2ISCO[1] is a platform for the design and implementation of deliberative and collaborative CBR applications. Using D^2ISCO [13] we can develop distributed decision making systems where each agent collaborates, argues and counterargues its local results with other agents in order to improve the global response of the system.

Essentially, this platform consists of the creation of a practical framework for the characterization of distributed argumentative systems. It implements a modification of the AMAL argumentation protocol that can be applied to different types of distributed systems. This protocol defines the way of exchanging arguments and counterarguments between agents. In the original AMAL protocol [23] these arguments are generated using a Description Logics system. However we follow a different approach to generate the arguments: a fuzzy reasoning system. Besides, it includes learning mechanisms to infer the trust models of the distributed CBR systems. D^2ISCO performs a practical demonstration of the following aspects (discussed in [13]): (a) The network topology significantly influences the results for the distributed CBR systems. (b) Social networks provide a network topology and a trust model that improves the results obtained. (c) The use of argumentations in the different reasoning processes improves the accuracy of the system.

Next, we will describe the argumentation protocol based on AMAL and the fuzzy decision system that generates the argumentations in our proposal.

3.2. Argumentation model for simulating decision making processes

Our proposal allows a group of agents $\{A_1, \ldots, A_n\}$ to deliberate about the correct solution to a problem Q by means of an argumentation process, refined with a fuzzy decision system that takes into account the personality of each user in the system and the trust they have with each other.

When the argumentation process starts, each agent uses its own internal memory to find a solution for a given query Q and then they begin a deliberation process with other agents by means of counterarguments. This also allows the agents to learn from the counterexamples received from other agents. The reasoning protocol begins with an agent (A_q) issuing a query to the agents that it is linked to. Each one of these agents retrieves k solutions from its own memory. Then, an argumentation process consisting of k cycles is performed to defend and discard the proposed solutions by means of counterexamples. When the process finishes A_q receives at most k trusted solutions.[2]

Our argumentation and solution retrieval process is hierarchical. When solving a problem Q, the agent that issues the query A_q becomes the root of the whole hierarchy of agents, defined by the structure of the social network. Then, the query is propagated to the leaves of the tree and the retrieval of the solution follows an inverse path. The leaves of the tree deliberate with their immediate parent node A_p, which organizes the reasoning. When this intermediate deliberation finishes, A_p participates in the deliberation organized by its parent node but this time it takes on the role of a child node A_c. This behavior is repeated until reaching the root A_q. Therefore, the distributed reasoning process finishes with agent A_q obtaining a list of agreed-on solutions ranked with a goodness value.

The method consists of a series of rounds. In the initial round, each agent states what its individual local solution for the problem Q is. Then, during each round an agent can try to rebut the solution or prediction made by any of the other agents by giving a counterexample. When an agent receives a counterargument or counterexample, it informs the other agents if it accepts the counterargument or not. Moreover, agents also have the opportunity to answer counterarguments by trying to generate a defense from the counterargument.

More specifically, the argumentation process is directed by the parent node/agent A_p, which issues the query and organizes the deliberation of its child nodes (A_c). The agent that leads the argumentation (A_p) plays an important role in the decision making process because it defines the confidence in the agents involved in the argumentation, and decides if a counterexample is accepted or rejected based on (a) its confidence in the agents involved, (b) the goodness value of their items, (c) her own personality and (d) the personality of the other agents involved. This process follows a peer-to-peer schema where the lead agent A_p deliberates with each child agent A_c individually. After every peer-to-peer deliberation A_p keeps the best item that has agreed with A_c. Then it will deliberate again with the following child agent and so on.

When a child agent A_c representing user u proposes a solution i to the parent agent A_p (representing user v), this solution is rated with a value $g_{u,i}$ that represents its goodness value according to the preferences and partial knowledge of user u. In DSS this measure of goodness is usually based on the estimation of an agent's/user's preferences. This estimation can be obtained by applying similarity metrics developed in the Case-Based Reasoning domain. In our case, this preference is a rating that denotes the preference of the user u for a given item i and is referred to as $rating(u,i)$. Next we will study how to compute the goodness value by means of a fuzzy system that combines these estimations. Then, the following subsection presents our novel method to modify this value according to the personality and trust of the users.

A counterexample against a solution i is another solution i_{ce}, which is rather similar to i but it has a low value of goodness. An agent A_p representing user v will present a counterexample when another agent A_c proposes a solution i that does not suit the preferences of v. The counterexample i_{ce}, is a solution from the memory of A_p that demonstrates that the solution i is not good because both are similar but the counterexample has a low value of goodness. A defense against a counterexample i_{ce} is another solution i_d that is rather similar to i_{ce} and it has a high value of goodness. Here, the defense i_d is presented by the same user that proposed the initial solution.

A key feature of the distributed reasoning protocol described above is the decision system that rates, accepts or rejects proposed solutions, arguments, counter-arguments and defenses. Our proposal relies on a fuzzy reasoner [33] that is able to find counterexamples that cannot be generated by logical induction. We have chosen this technique because it allows us to perform deliberations about incomplete preference memories and eliminates the restrictions that appear when using descriptive logics (used in the original AMAL protocol). The details of fuzzy reasoner are described in [13]. Next we will show how to include the personality and trust factors to improve the ratings given to a product by an agent.

3.3. Including personality and trust in the argumentation process

Each agent A_c representing the user u that receives a query must return a proposed solution i to the lead agent A_p that represents user v. This solution is the best one found in the memory of user u rated with a value $g_{u,i}$ that reflects its goodness. As we have previously explained this value is obtained by means of a fuzzy system that combines both the similarity value of the query with that product and the rating given by the user according to the partial knowledge of the domain stored in the memory of the user (referred to as $rating(u,i)$). To compute both values we apply similarity metrics taken from the CBR domain.

However, we have enhanced this rating value with the social factors described in Sections 2.1 and 2.2: personality and trust. As we stated before, these factors will boost the performance of the group recommender as they represent the real features of human deliberations more accurately.

Our method modifies the rating value according to the difference in personalities between users u and v. If the lead user v has a strong personality it won't easily accept the solution proposed by u. Therefore we increment or decrement the goodness value according to the difference in personalities. Moreover, the trust $t_{u,v}$ between the two users must be

[1] Deliberative, DIstributed and COllaborative extension for jCOLIBRI.

[2] Note that usually $k = 1$.

taken into account when computing the rating value: a low trust value implies a low goodness. Therefore, we obtain a modified goodness value $g'_{u,i}$ computed as follows:

$$g'_{u,i} = t_{uv} \cdot \left(g_{u,i} + \Delta p_{u,v} \right) \tag{2}$$

where $\Delta p_{u,v}$ is the personality difference between users, and $t_{u,v}$ is the trust that exists between users u and v. $g_{(u,i)}$ is the goodness value for the solution i given by user u and computed by means of the fuzzy system.[3]

With this formula we modify the goodness value estimated for every user and item. It will be higher when the personality and trust parameters are also high, and lower in the opposite case. Therefore, the argumentation process presented before uses the $g'_{u,i}$ value instead of the original $g_{u,i}$ goodness to include the social factors in the group decision making process.

Once the distributed argumentation protocol is described, next we describe how to apply it in a real case study.

4. Case study: movie recommendation

To evaluate our GDSS methodology we have chosen the recommendation domain because it is a clear example of GDSS systems. Moreover, movie recommendation is a very accessible area with datasets available and, more importantly, well known to users. The main hypotheses to validate are:

H1. The multi-agent architecture of deliberative agents connected according to a real social network improves the accuracy of standard "fully connected" group recommenders.

H2. The personality and trust factors improve the performance of distributed group recommenders.

H3. Individual satisfaction increases when using this new group recommendation technique because agents are able to argue about the item chosen.

Following sections detail the case study used to demonstrate the hypotheses formulated. Next we describe the experimental set-up, followed by the results.

4.1. Experimental set-up

In order to perform our experiment in the movie recommendation domain, we created two events in two different social networks, Facebook[4] and Tuenti.[5] In these events we asked 58 participants to complete three questionnaires.[6]

The first questionnaire obtains the individual preferences of the user about cinema. Users have to evaluate 50 heterogeneous movies from the MovieLens data set [3] (rating them using a Likert scale from 0.0 to 5.0). On average, each user rated 30 movies. These movies rated make up the list of products that is assigned to each agent as the profile of each participant, that is, the memory of preferences. Next, a second test asks users to choose their 3 favorite movies from a list of 15 recent movies (chosen heterogeneously from movies in the MovieLens database), that represents a movie listing from a cinema. These movies are the ones they would actually like to watch or had enjoyed best, and are denoted as their individual favorites set, if_u.

To measure the accuracy of the group recommendation we created groups with our participants and we asked them to simulate that they were going to the cinema together. We provided them with the 15 movies that represented our movie listing in the second questionnaire and we asked them to choose which 3 movies they would actually watch together. We managed to gather 15 groups of 9, 5 or 3 members. The three movies chosen by each group G are stored as the *real group favorites* set, rgf_G. This way, to evaluate the accuracy of our recommender we were able to compare the set proposed by the recommender – the gf_G set – with the real preferences rgf_G. Particularly, we measure the number of movies in gf_G that are also in rgf_G. The concrete evaluation measures used to compare the two sets are detailed in Section 4.2.

Once we have the memory of preferences for each user, we need the personality and trust factors. A third questionnaire serves to obtain the personality value, p_u, by asking the 30 questions from the TKI personality test [29]. Next, trust among users is calculated by analysing the factors explained in Section 2.2. These factors are combined using a weighted average, see Eq. (1), whose weights are obtained through a genetic algorithm (GA). Our GA manages a population of vectors of weights $(\overline{\alpha})$. These vectors are combined and mutated in order to maximize a fitness function. Therefore, the individuals of the GA population (vectors of weights) are used to compute the trust factor $t_{u,v}$ required by the approach. The fitness function compares the result of our group recommender configured with each individual $(\overline{\alpha})$ to the real rating given by the users. These weights are shown in Fig. 2 where we can observe that common friends, pictures, interests and friendship duration are the most relevant factors.

Next step is the configuration of the multi-agent system. As we have previously detailed every agent is connected to others according to the real relationships of the represented user in a real social network. Therefore, we create an agent for each user that is linked to the real friends of the user.

Finally, every agent requires an individual movie recommender to find suitable movies for a given query. These recommenders return the rating value *rating*(u,i) described in Section 3.3. To obtain this value we use CBR similarity metrics applied to the products and ratings in the memory of user preferences. The individual recommender implemented follows a knowledge based approach [16] that compares descriptions of the products and returns a collection composed of the ones most similar to the query.

4.2. Evaluation metrics

The aim of the evaluation is to compare the results of our recommender system to the real preferences of the users (that is, what would happen in a real life situation). This evaluation has some particular features that must be taken into account. First, we are not interested in a long list of ordered movies when estimating the movies a user or group should watch. Real users are only interested in a few movies they really want to watch. This fact discards several evaluation metrics that compare the ordering of the items in the real list of favorite movies and the estimated one. On the other hand, the number of relevant and retrieved items in our system is fixed. Therefore, we cannot use general measures like recall or precision. However, there are some metrics used in the Information Extraction field that limit the set retrieved. This is the case of the *precision@n* measure, which computes the *precision* after n items have been retrieved. In our case, we can use the *precision@3* to evaluate how many of the movies in gf_G are in the rgf_G set (note that $|\text{rgf}_G| = 3$). This kind of evaluation can be seen from a different point of view: we are usually interested in having at least one of the movies from gf_G in the rgf_G set. This measure is called *success@n* (or *s@n*) and returns 1 if there is at least one hit in the first n positions. Therefore, we could use *s@3* to evaluate our system by computing the rate of recommendations where we have at least "one-hit" in the real group favorites list. For example, a 90% accuracy using *s@3* means that the recommender suggests at least one correct movie for 90% of the groups evaluated. In fact, *s@3* is equivalent to having

[3] The goodness value is modified by $[-1,1]$ by the $p_{u,v}$ factor and decreased according to the trust $t_{u,v}$. These ranges could be scaled according to the concrete range of the goodness function.

[4] http://www.facebook.com.

[5] http://www.tuenti.com. Tuenti is the most popular social network in Spain for people in their twenties.

[6] Questionnaires are accessible at http://www.lara.warhalla.com.

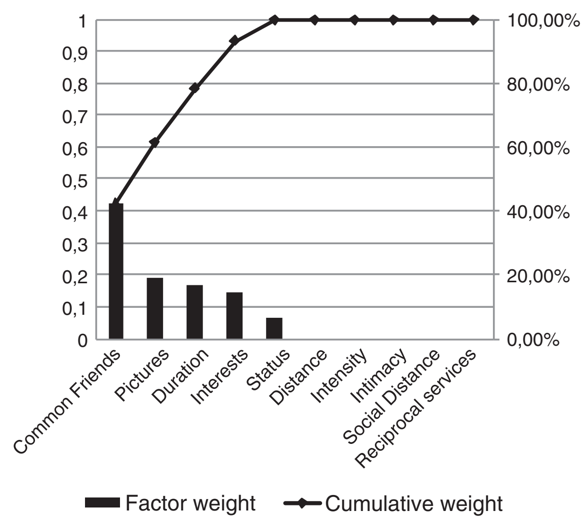


Fig. 2. Weights to obtain the trust value.

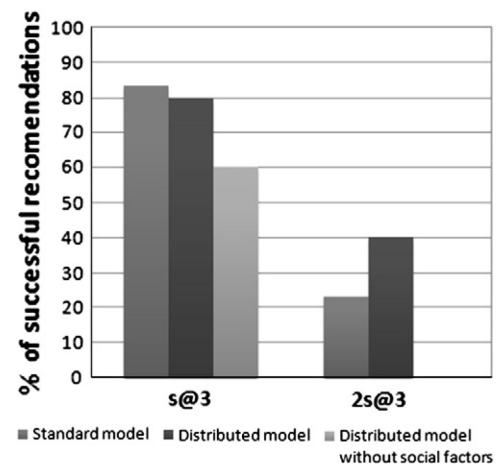


Fig. 3. Comparison of the global results. Standard vs. distributed model.

precision@3 ≥ 1/3. We can also define a "two-hits" metric or double-sucess@3 (2*s*@3), equivalent to *precision*@3 ≥ 2/3, which represents the number of times the estimated favorites list gf_G contains at least two movies from rgf_G. Obviously, it is much more difficult to achieve high results using this second measure.

5. Discussion of the evaluation results

Given the three hypotheses we want to validate, and the experimental setup, in this section we discuss the obtained results. We have compared the proposed distributed model to a standard "fully connected" group recommender to prove H1. Global results are shown in Fig. 3. As we can observe in the two first columns of the figure, the two approaches are similar if we use the *s*@3 measure, the standard model being slightly better by 3%. However, if we evaluate using the more demanding 2*s*@3 metric we obtain an improvement of 17%. Therefore we can confirm the first hypothesis and assert that the social network organization of the agents improves the performance of the system.

Next, we studied the accuracy of our proposal regarding group size. Fig. 4 shows the results of our experiment for the three different group sizes. Here we are using the personality and trust factors plus deliberation capabilities with a social network topology. The *s*@3 line shows how many times our group recommender provided one product within its first three choices that the group would really have selected. And the 2*s*@3 shows the number of times that the group recommender provided two products that the group would have really selected. This last measure is therefore much more difficult to obtain so the percentage is always lower. By studying the size of the groups we can observe that generally the recommender gives better results for smaller groups.

The second hypothesis to be tested (H2) is that social factors improve the performance of our distributed recommender. To confirm it we repeated the experiments using the original goodness value $g_{u,i}$ instead of the modified version $g'_{u,i}$, shown in Eq. (2), which includes the personality and trust parameters. The results obtained are reported in the last column in Fig. 3, where we can observe a poorer performance particularly in the 2*s*@3 metric.

Furthermore, we wanted to check that the improvement achieved when using social factors is due to the argumentation protocol that takes them into account when finding the best alternative for the group. That is, there is no correlation between the performance and the social characterization of the groups being evaluated. To discard this lack of correlation we described the groups regarding the average and standard deviation of the trust and personality values of their members. The average values let us measure if a group is mostly composed of members with a high/low personality or trust; meanwhile the standard deviation reflects the homogeneity of the group regarding the two parameters. After running several statistical tests we found no evidence of correlation between these variables and the performance of the system (evaluated using the *precision*@3 metric that summarizes *s*@3 and 2@3[7]). The resulting correlation matrix is shown in Fig. 5. As we can observe, there is no correlation at all between the variables studied. Therefore we can conclude that the improvement in performance is rooted in the appropriate combination of social factors carried out by the argumentation protocol.

Finally we are very interested in each user's individual satisfaction. Our hypothesis (H3) is that users' preferences are taken more into account by our model because the deliberation process lets them argue about and rebut the product to be chosen by the group. Therefore, their individual satisfaction should be higher.

Individual satisfaction is measured by comparing how many movies proposed by the group recommender – gf_G – are in the user's list of favorite movies – if_u – obtained by means of the third questionnaire. When comparing our distributed model with the standard model we found an increase of 5% in the average satisfaction of the users. Our explanation for this result is that the argumentation method enables users to express their opinions more clearly. For example, if they specially dislike one movie and it is proposed by another agent during the deliberation process, the representing agent can state its dislike in the rounds of argumentation and counter-argumentation.

The experimental validation of our hypothesis represents a valuable contribution to the Group Decision Support Systems field as they confirm the utility of exploiting social network structures to integrate social knowledge in the decision making process. Moreover we propose multi-agent systems as a suitable architecture to implement such decision systems. We also have probed that trust and personality are two social factors that help to obtain an accurate reproduction of the decision making processes run by real groups of people. The inclusion of these factors in GDSS increases the individual satisfaction of the users involved in the group decision making process, and this fact should be taken into account when designing this kind of systems.

6. Conclusions and future work

Group Decision Support Systems represent a wide range of applications with rising impact in the current web [25,32]. Moreover, the need for systems capable of providing decision support for groups of people is attracting more interest as there are many leisure activities that are carried out in groups and organized through social networks. Therefore, we propose a novel approach for GDSS based on a

[7] Note that *precision*@3 only takes the following discrete values: $\left\{\frac{0}{3}, \frac{1}{3}, \frac{2}{3}, \frac{3}{3}\right\}$.

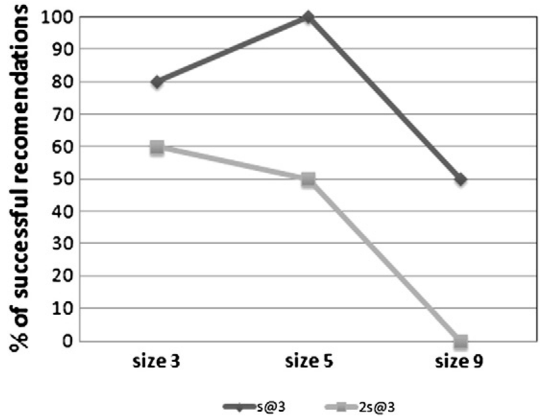


Fig. 4. Comparison of the results obtained using distributed models plus the personality and trust factors for the 3 different sizes of groups.

distributed architecture of agents with deliberation capabilities that argue and defend the preferences of the represented user to reach a joint solution. This architecture exploits social information available in social networks, like the topology of user relationships and their mutual trust to improve the performance of the system. Moreover, our model includes the personality of each member of the group to reflect real argumentation processes accurately.

The personality factor reflects the cooperativeness or selfishness of each user when selecting a product for the whole group. It measures the degree of acceptance of the products proposed by other users and the way to solve conflicts. To obtain this factor we use a popular personality test called TKI [29]. The second factor used by our model is trust among users. Several studies point out the importance of personal trust in real argumentations and the requirement to include this feature in software models reproducing those processes [9,27]. We measure social trust among users by analysing several features found in common social networks. Examples of these social factors are distance in the social network, number of common friends, intensity, intimacy or duration of the relationship.

Both parameters, personality and trust, are used to customize the argumentation processes performed by the agents. However, the most important novelty of this paper is the organization of these agents according to users' real social relationships. We provide empirical evidence that this "social network topology" more accurately reproduces the argumentations carried out by humans when discussing a joint choice.

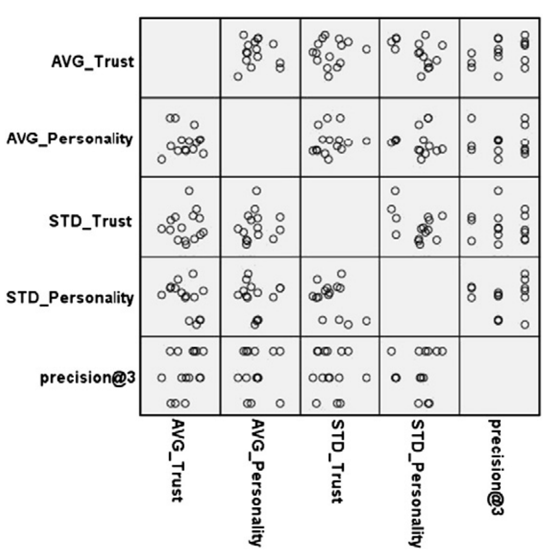


Fig. 5. Correlation between performance (evaluated using *precision*@3) and the average (AVG) and standard deviation (STD) of the personality and trust parameters.

Our approach has been tested in the movie recommendation domain although our proposal is not specific to this domain and could be easily adapted to others. To evaluate the accuracy of the system we compare the products chosen by real groups of users to the ones proposed by our recommender. We apply two different metrics to measure the degree of overlap between both sets of products. Our system performs in a similar way to standard group models when we evaluate internally whether there is at least one product correctly proposed by the recommender. However, when we toughen the evaluation metric and measure whether there are at least two correct proposals, our approach is 17% more accurate.

Our experiments also compare users' individual satisfaction. This is how individual preferences are taken into account when deciding a common product for the group. Thanks to the argumentation capabilities given to agents we obtain an increase of 5% in individual satisfaction.

Regarding future work we think that any improvement in the individual goodness value used internally by each agent to represent the preference of the user for a given product would have a high impact on the final result for the group. To do so, we plan to test other individual preference estimation techniques like collaborative filtering. Furthermore, we are currently implementing a real application in Facebook that will apply the proposed techniques to recommend movies to groups of friends. This application will serve as a real scenario to validate current and future methods.

The most significant limitation of our study is the nature of the groups being evaluated. In this study we have evaluated groups which members had a friendship relation. However the performance of the system when applied to other kinds of groups, such as families, should be also studied.

We can conclude that our model proposed accurately reproduces real decision making processes experienced by groups of people when deciding leisure activities. The improvement is not only due to the inclusion of personality and social trust factors, but also to the agent-based architecture that simulates face-to-face discussions between users.

References

[1] A. Aamodt, E. Plaza, Case-based reasoning: foundational issues, methodological variants, and system approaches, Artificial Intelligence Communications 7 (1994) 39–59.
[2] P. Avesani, P. Massa, R. Tiella, A trust-enhanced recommender system application: Moleskiing, SAC '05: Proceedings of the 2005 ACM Symposium on Applied Computing, ACM, NY, USA, 2005, pp. 1589–1593.
[3] J. Bobadilla, F. Serradilla, A. Hernando, Collaborative filtering adapted to recommender systems of e-learning, Knowledge-Based Systems 22 (2009) 261–265.
[4] Srdjevic Bojan, Linking analytic hierarchy process and social choice methods to support group decision-making in water management, Decision Support Systems 42 (2007) 2261–2273(Decision Support Systems in Emerging Economies).
[5] B. Brehmer, Time scales, distributed decision making and modern information technology, in: J. Rasmussen, B. Brehmer, J. Leplat (Eds.), Distributed decision making: cognitive models for cooperative work, Wiley, 1991.
[6] Y.L. Chen, L.C. Cheng, C.N. Chuang, A group recommendation system with consideration of interactions among group members, Expert Systems with Applications 34 (2008) 2082–2090.
[7] A. Crossen, J. Budzik, K.J. Hammond, Flytrap: intelligent group music recommendation, IUI, 2002, pp. 184–185.
[8] E. Gilbert, K. Karahalios, Predicting tie strength with social media, CHI '09: Proceedings of the 27th international conference on Human factors in computing systems, ACM, New York, NY, USA, 2009, pp. 211–220.
[9] J.A. Golbeck, Computing and applying trust in web-based social networks. Ph.D. thesis, College Park, MD, USA , 2005. (Chair-Hendler, James).
[10] J. Golbeck, Combining provenance with trust in social networks for semantic web content filtering, in: L. Moreau, I.T. Foster (Eds.), Provenance and Annotation of Data, International Provenance and Annotation Workshop, IPAW 2006, Chicago, IL, USA, May 3–5, 2006, Revised Selected Papers, volume 4145 of *Lecture Notes in Computer Science*Springer, 2006, pp. 101–108.
[11] J. Golbeck, Generating predictive movie recommendations from trust in social networks, iTrust: 4th International Conference on Trust Management, IUI, 2006, pp. 93–104.
[12] J. Golbeck, J.A. Hendler, Inferring binary trust relationships in web-based social networks, ACM Transactions on Internet Technology 6 (2006) 497–529.
[13] S. González-Sanz, J.A. Recio-García, B. Díaz-Agudo, D2ISCO: Distributed Deliberative CBR Systems with jCOLIBRI, 1st Int. Conference on Computational Collective Intelligence., volume 5796 of *LNCS*, Springer, 2009, pp. 321–332.
[14] T. Grandison, M. Sloman, A Survey of trust in internet applications, IEEE Communications Surveys and Tutorials 3 (2000) 2–16.

[15] A. Jameson, B. Smyth, Recommendation to groups, in: P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), The Adaptive Web, Methods and Strategies of Web Personalization, Volume 4321 of Lecture Notes in Computer Science, Springer, 2007, pp. 596–627.

[16] D. Jannach, M. Zanker, A. Felfernig, G. Friedrich, Recommender Systems: An Introduction, Cambridge University Press, 2011.

[17] D.B. e. Leake, Case-based reasoning: experiences, lessons, and future directions, Menlo Park, CA: AAAI Press/MIT Press, Menlo Park, CA, 1996.

[18] H. Lieberman, N.W.V. Dyke, A.S. Vivacqua, Let's browse: a collaborative web browsing agent, IUI, 1999, pp. 65–68.

[19] J. Masthoff, Group modeling: selecting a sequence of television items to suit a group of viewers, User Modeling and User-Adapted Interaction 14 (2004) 37–85.

[20] J. Masthoff, A. Gatt, In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems, User Modeling and User-Adapted Interaction 16 (2006) 281–319.

[21] D.W. McDonald, Recommending collaboration with social networks: a comparative evaluation, CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, New York, NY, USA, 2003, pp. 593–600.

[22] L. McGinty, B. Smyth, Collaborative case-based reasoning: applications in personalised route planning, Procs. ICCBR, Springer LNAI, 2001, pp. 362–376.

[23] S. Ontañón, E. Plaza, An argumentation-based framework for deliberation in multi-agent systems, Argumentation in Multi-Agent Systems. LNCS, volume 4946, 2008, pp. 178–196.

[24] L. Quijano-Sánchez, J.A. Recio-García, B. Díaz-Agudo, Personality and social trust in group recommendations, Procs of the 22th International Conference on Tools with Artificial Intelligence, ICTAI'10, IEEE Computing Society, 2010, pp. 121–126.

[25] J. Shim, M. Warkentin, J.F. Courtney, D.J. Power, R. Sharda, C. Carlsson, Past, present, and future of decision support technology, Decision Support Systems 33 (2002) 111–126(Decision Support System: Directions for the Nest Decade).

[26] S.B. Shum, L. Cannavacciuolo, A.D. Liddo, L. Iandoli, I. Quinto, Using social network analysis to support collective decision-making process, IJDSST 3 (2011) 15–31.

[27] R.R. Sinha, K. Swearingen, Comparing recommendations made by online systems and friends, DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries, 2001.

[28] J. Song, F. Zahedi, Trust in health infomediaries, Decision Support Systems 43 (2007) 390–407.

[29] R.H. Kilmann, K.W. Thomas, Developing a Forced-Choice Measure of Conflict-Handling Behavior: The "MODE" Instrument, Educational and Psychological Measurement 37 (1977) 309–325, http://dx.doi.org/10.1177/001316447703700204.

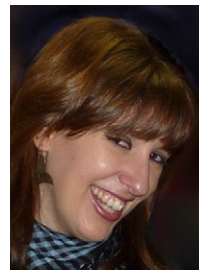
[30] T.W. Wang, S.K. Tadisina, Simulating internet-based collaboration: a cost-benefit case study using a multi-agent model, Decision Support Systems 43 (2007) 645–662.

[31] In: G. Weiss (Ed.), Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, MIT Press, Cambridge, MA, USA, 1999.

[32] C.N. Ziegler, J. Golbeck, Investigating interactions of trust and interest similarity, Decision Support Systems 43 (2007) 460–475.

[33] H.J. Zimmermann, Fuzzy Set Theory—and its Applications, 3rd ed. Kluwer Academic Publishers, Norwell, MA, USA, 1996.

Juan A. Recio-Garcia is Assistant Teacher at the Artificial Intelligence Department at the Computer Science Faculty at the Complutense University of Madrid, where he obtained a PhD in Computer Science in 2008. His research has focused on the confluence of Software Engineering and Case-Based Reasoning, developing the COLIBRI framework for building CBR systems. Currently he is also working in the areas of Recommender Systems and Semantic Web. He has several publications at International Conferences and Journals.



Lara Quijano is a PhD student at the Complutense Univeristy of Madrid. Her areas of research are Recommender Systems, Social Networks, Social Factors, Multi-Agent Systems, Decision Support Systems, etc. She is presently working in Decision Support Systems where she has developed several algorithms to reflect accurately argumentations between users in this kind of systems. Her publications have appeared in various Journals and International Conferences.

Belen Diaz-Agudo is a professor at the Artificial Intelligence Department at the Computer Science Faculty at the Complutense University of Madrid. Her research interests include Case Based Reasoning, Semantic Web, Recommender Systems and their relationships to applied intelligent systems. She has more than 50 publications at International Conferences and Journals, and has participated in the organization of AI conferences.

Chapter 20

Feedback on group recommendations

20.1 Citation

Quijano-Sánchez, L., Bridge, D., Feedback on group recommendations, in Workshop on Case-based Reasoning in Social Web Applications, in: Procs. of the 21st International Conference on Case Based Reasoning, ICCBR'13, Saratoga Springs, NY, USA, pages: 17-26, 2013. ISBN 978-3-642-39055-5. ICCBR is ranked as CORE C in the CORE ranking.

20.2 Contributions covered by this paper

In this paper we have studied what do users expect and consider most important from group recommender applications. To do so, we have analysed the feedback given by our users when using our Facebook social group recommender application, *HappyMovie*. Last but not least, we have started a discussion about how to improve and what key factors should the next generation of group recommender applications take into account.

Feedback on group recommendations

Lara Quijano-Sánchez[1] and Derek Bridge[2]

[1] Department of Software Engineering and Artificial Intelligence, Universidad Complutense de Madrid, Spain
[2] Department of Computer Science, University College Cork, Ireland
lara.quijano@fdi.ucm.es, d.bridge@cs.ucc.ie

Abstract. This is a discussion paper on the subject of group recommender systems. In the recent past, we have built such a recommender system, *HappyMovie*, and we have used variants of it in a number of experiments. In the light of our experience, we look at the the kind of feedback users might give to a group recommender, informed also by new results from a survey that we conducted. We conclude with ideas for the development of the next generation of group recommender systems.

1 Introduction

Recommender Systems use inferred preferences to suggest to their user items that the user might like to consume. Group Recommender Systems do the same, but they recommend items to a group of users, where the group intends to consume the items together.

Case-Based Reasoning (CBR) has a long history of contributing to recommender systems [2]. Most simply, we can build a case-based recommender system where the cases represent the items (e.g. products) and the CBR application recommends cases that are similar to the user's partially-described preferences. More interestingly, the cases in the case base can instead describe the experience of consuming recommended products [12].

We have built a group recommender system for movies. We have also built a variant of our group recommender that uses CBR in the way described at the end of the previous paragraph. We briefly describe our group recommender system and this case-based variant in Section 2.

In the course of developing these recommender systems, we have uncovered a number of perspectives on the kind of feedback that group recommender systems might seek, which we present in Section 3. To make this more concrete, we ran a group recommender system experiment with real users and administered a questionnaire to the participants.We describe the experiment and the results of the questionnaire in Sections 4 and 5. We conclude in Section 6 with ideas for the development of the next generation of group recommender systems.

2 Group Recommender Systems

Commonly, group recommender systems aggregate predicted ratings for group members [5]. First, a single-person recommender system predicts each group member's rating for each candidate item. This might be done, as it is in our *HappyMovie* group recommender system, using a standard user-based, nearest-neighbours collaborative filtering approach. Next, the recommender aggregates the ratings, e.g. by taking their maximum or their average. Finally, it recommends the candidate items that have the highest aggregated predicted ratings.

There are many possible variations on this common approach.Our *HappyMovie* system, for example, applies a function to each predicted rating *before* aggregation [11]:

- On registration with *HappyMovie*, users take a personality test whose results are converted into a personality score between 0 and 1, where 0 means a cooperative person and 1 means a selfish person [15]. A user's predicted rating will count for more in the aggregation if her personality score is higher than that of the other group members.
- After registration, the strength of connection ('trust') between pairs of users is mined from social network data. A person's predicted ratings are pulled towards the opinions of the other group members to a degree based on their strength of connection [3].

In [13], we presented a variant of *HappyMovie* that uses CBR: its aggregation of predicted ratings is a lazy and local generalization of the behaviours captured by the neighbouring cases in the case base. First, it uses a user-based, nearest-neighbours collaborative filtering approach to predict each group member's rating for each candidate item. Next, it retrieves cases, i.e. past group recommendation events, that involve groups that are similar to the active group. Case retrieval uses a user-user similarity measure, and, as a by-product, it aligns each member of the active group with a member of the group in the case. The similarity measure compares group members on their age, gender, personality and ratings and the degrees of trust between members of each group. Then, it reuses each case that is retrieved: the contributions that each group member made in choosing the selected item are transferred to the corresponding member of the active group. This is done by scoring the new candidate items by their item-item similarity to the selected item. In this way, the retrieved cases act as implicit models of group decision-making, which are transferred to the decision-making in the active group. Finally, it recommends the candidate items that have obtained the highest scores.

3 Feedback to Group Recommender Systems

Suppose we have a group recommender; for concreteness, suppose it recommends movies. Consider the scenario where the recommender recommends a movie to a group, the group accept the recommendation, they see the movie together, and some or all of the group members come back and provide explicit feedback in the form of ratings. What sort of feedback should the recommender solicit?

3.1 Actual ratings

Like conventional recommender systems, most group recommender systems ask each user how much she likes the movie, e.g. as a star-rating on a five point scale. User-movie ratings are the most important (and often the only) form of *training data* for collaborative recommender systems. The additional training data may improve single-user predictions. And, since most group recommender systems work by aggregating single-user predictions, this in turn may improve group recommendations.

But in single-user recommender systems, user-movie ratings may be used in another way. They may be used to *evaluate* the recommender. However, let's be more precise. The *actual* user-movie rating that the user supplies can be compared with the *predicted* rating which the system computed when making the recommendation; for example, an error value can be computed from their difference. This evaluates the predictions, not the recommendations. Of course, the assumption is that the better the predictions, the better the recommendations.[3]

3.2 User satisfaction with the recommendation

But, even if prediction accuracy is high, it does not follow that recommendation quality will be high. That depends on how successful the aggregation is. For example, if a user watches a recommended movie in a group and later gives it a low rating, this does not mean that the group recommender has done a poor job. It may even be that the group recommender predicted that this user would give a low rating. But the movie was recommended nonetheless, as it was judged to be the one that best reconciled the different tastes of the group members: sometimes people have to lose out to reach any decision; sometimes people lose out to group members who have special priority such as children or members with disabilities; sometimes the preferences of a user who was favoured on a previous occasion may, in the interests of fairness, be weighted lower on a subsequent occasion [14].

So there is a separate dimension that can be measured: user satisfaction with the recommendation. For example, a user who dislikes the movie (gives it a low rating) may nevertheless be satisfied with the recommendation, especially if she appreciates that it has been necessary to balance conflicting interests. Her satisfaction might be all the greater if she has a more accommodating (less selfish) personality type, or if the recommendation better matches the tastes of group members with whom she has stronger connections through contagion and conformity [10]. A father who takes his children to the cinema provides one such example: if he knows that the recommendation is a good one for his children, his own satisfaction with the recommendation may increase.

[3] It is well-known that error values or measures of accuracy alone are not sufficient for evaluating recommender systems; see, e.g., [4]. Notwithstanding these well-known deficiencies, our purpose here is to show that there are *additional* problems —ones that have not been well-documented before— that arise specially in the case of groups recommender systems.

Additionally, expectations can influence satisfaction [10], even in single-user recommenders, and these can be influenced to some extent through explanations (e.g. "None of this week's movies is a good match to your preferences. The one I'm recommending is the best of a poor crop."). This may be even more important in group recommenders where the trade-offs that have been made can be explained.

3.3 The group experience

But there is yet another dimension to group movie-going which goes beyond both whether each member liked the movie (their rating) and individual user satisfaction with the recommendation. There is what we might call the *experience as a whole* (or just *the experience* for short).[4] Although the movie might be one that a group member would not choose for herself, she may still have had an enjoyable time. She may not have liked the movie; she may not have been satisfied with the recommendation (e.g. in the way that it traded-off her preferences against those of other members of the group), but watching it with her friends was still fun. Indeed, it might even be the case that the majority of the group thought a movie was terrible but they may still have enjoyed the outing, e.g. perhaps its awfulness provoked hilarity or heated discussion. The father watching a movie with his children may have had a great time, and this is distinct from, although not wholly uncorrelated with, his movie rating and his satisfaction with the way the recommendation traded-off group interests. The same is true of most consumption done in groups, e.g. dining out together, making excursions together, and so on —the quality of the experience is not necessarily related to the what each user thought of the item, nor the user's satisfaction with the recommendation.

Different members of the group may evaluate the group experience in different ways. For example, the heated debate that ensued from a controversial movie may be perceived by one group member to have been exhilarating but perceived by another to have been uncomfortable. On the whole, however, we probably expect some agreement about the group experience due to the contagion and conformity effects mentioned earlier [10].

4 *HappyMovie* Experiment

In an effort to explore these issues further, we ran an experiment with real users. Sixty students from a masters-level Artificial Intelligence course participated. They were between 20 and 26 years' old. Twenty-three were female (38.3%); thirty-seven were male (61.6%). Individually, each student completed a Personality Survey, which used TKI's Alternative Movie Metaphor [15]: for each of five different dimensions of personality, we showed the student two well-known

[4] We are not referring here to the user experience [6] that comes from engaging with the software; we are referring to the experience of consuming (in our case, in a group) the recommended items.

movie characters whose personalities oppose each other along that dimension; the student selected the member of the pair with which she most identifies. The result is a numeric score in $[0, 1]$. In essence, a value of zero is a very cooperative person and a value of one is a very selfish person. Each student also completed a Preferences Survey: we asked them to rate 70 well-known movies using a five-point rating scale. *HappyMovie* uses these ratings for its collaborative filtering. Finally, the strength of connection ('trust') between pairs of users was mined from Facebook interactions.

We formed 20 groups, each comprising three students. Each group used *HappyMovie* to create a group event —an outing to the cinema together; they received three movie recommendations from *HappyMovie* —the three that the recommender decided were best for the group, from a listing of current movies; and they agreed on one of the recommended movies —the one that their group would go to see. We asked them to imagine going to the cinema to watch that movie with the members of their group.

Then, individually they answered a questionnaire of eight questions.[5] The first seven questions were about the movie that they had selected:

1. Give your personal rating for this movie (0 for a movie you really disliked, up to 5 for a movie you really liked).
2. Give the rating that you think your friend 1 in the group will give to this movie (0 if you think s/he really disliked it, up to 5 if you think s/he really liked it).
3. Give the rating that you think your friend 2 in the group will give to this movie.
4. Evaluate the enjoyability of your experience of watching this movie with your group (0 for a really bad experience, up to 5 for a good experience — where you had a great time together).
5. Evaluate the enjoyability of the experience that you think your friend 1 in the group will have by watching this movie with your group.
6. Evaluate the enjoyability of the experience that you think your friend 2 in the group will have by watching this movie with your group.
7. Out of the listing of current movies, do you think that this would have been your choice if you had to go to the movies together in reality — without using *HappyMovie* (0 for 'No, we would have never chosen this movie', up to 5 'Yes, we would have definitely chosen this movie').

The eighth question asked a more general question about recommendations:

8. When you go to the movies with a group of friends, what do you value most about a recommendation? Order the options by importance (most important first):
   (a) That the movie was a good movie —in terms of quality.
   (b) That you enjoyed the movie individually.

[5] We ran the experiment with students whose first language was Spanish. The questions that we show here are paraphrases into English of the Spanish questionnaire.

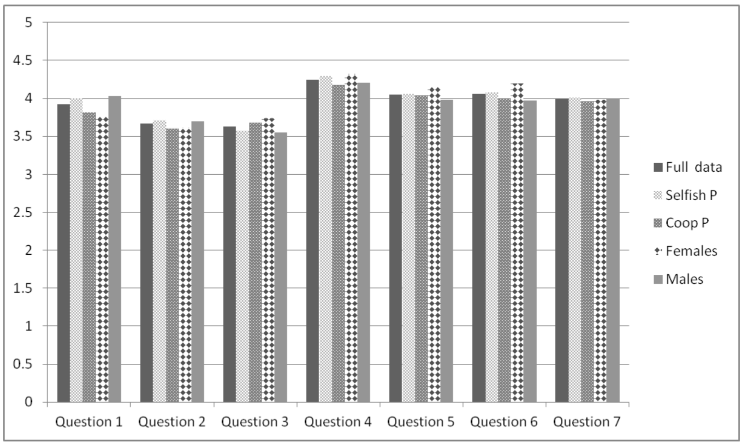


Fig. 1. Average rating by user group of responses to questions 1–7

(c) That you and your friends had a good experience watching the movie.
(d) That the recommended movie was the one that you would have chosen as a group.

These relate to the discussion in the previous section in the following way: option (b) is related to movie rating (Section 3.1); option (c) is what we called the group experience (Section 3.3); and option (d) is about user satisfaction with the recommendation (Section 3.2). Option (a) is an 'objective' notion of quality.

5 *HappyMovie* Experiment Results

For analysis of the results of the questionnaires, we consider five types of user:

Full data: all sixty users;
Selfish P: the thirty-five users with a more selfish personality, i.e. users whose TKI personality score is no less than 0.6;
Coop P: the twenty-five users with a more cooperative personality, i.e. users whose TKI personality score is less than 0.6;
Females: the twenty-three females; and
Males: the thirty-seven males.

A background observation is that the male students tended to have higher TKI personality values(average 0.68784), implying more selfish personalities, whereas the female students had a lower average TKI personality value (0.46052), implying less selfish personalities.

The results for the first seven questions are in Figure 1. We can conclude:

– On average, these users rate the group experience more highly than they rate the movie (compare Questions 4 and 1), and they think their friends will do the same (Questions 5 & 6 versus 2 & 3).

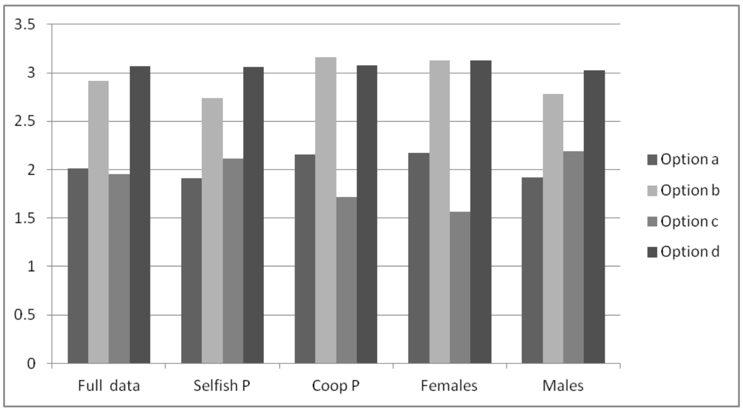


Fig. 2. Average rank by user group of responses to question 8

- On average, these users give higher ratings to the selected movie (Question 1) than they think their friends will give to the movie (Questions 2 and 3). Similarly, their rating of the experience of seeing the movie with these friends (Question 5) is higher than what they think their friends' ratings of the experience will be (Questions 6 and 7). So they feel that the recommender has favoured them, or that they have 'won' in the decision about which movie the group will go to see. This raises the question of whether users tend to rationalise decisions even when the decision goes against them.
- The results for users with the more selfish personality values are very similar to the results for male users; and the results for users with the less selfish personality values are very similar to the results for female users. This follows from the background observation we made, that the male students had on average more selfish personalities than the female students.

The results for the eighth question are in Figure 2. In this Figure, if the bar for, e.g., option (a) is shorter than the bar for option (b), then this means that, on average, users gave option (a) greater importance than option (b).

Looking first at the results for the full set of users, we see that on average they ordered the options in decreasing importance as follows: good group experience (option c); good quality movie (option a); high rating (option b); and high satisfaction with the recommendation (option d). From the Figure, we see that the first two options are very close in their average rank. Bear in mind, though, that this experiment has more males than females and hence more users who, on average, are more selfish. A clearer picture emerges when we look at these different types of user separately.

If we look at users with less selfish personalities (and, equally, the female students in this experiment), we see that this ordering is accentuated: the group experience (option c) is more markedly important than the movie quality (option a), and there is more equivocation between options (b) and (d). But for users with more selfish personalities (and, equally, the male students), we see that the ordering of the first two options is reversed: recommending a good quality movie (option a) is more important that recommending a movie that results in a good group experience (option c). It is perhaps no surprise that more selfish users treat the group experience as less important. It is interesting though that movie quality is more important than whether they like the movie (option b) and whether they are satisfied with the recommendation (option d).

Overall, there are two surprises in the results. First, across all users the idea that a recommender does a good job when it recommends the movie that the users would have gone to see in reality (option d) is always treated as being of low importance. Second, across all users 'objective' movie quality is important: perhaps we need to ensure that we recommend items whose expert reviews or population average ratings exceed a minimum quality.

It would be unwise to draw firm conclusions from experiments like this one, particularly because the questions make rather subtle distinctions which the respondents may have misunderstood and the number of respondents is quite low. What we are probably safe to conclude is the importance of the group experience, the importance too of choosing high quality movies, and the sense that, if there is a trade-off to be made, the less selfish people are the ones who can remain satisfied even when the trade-off is at their expense.

6 Discussion

Our investigation has implications for the evaluation of group recommender systems. It is not enough to ask users only for a movie rating: doing so, fails to explicitly evaluate other dimensions of recommendation quality. But we want to focus in this paper on the implications for the design of group recommender systems, rather than their evaluation.

A first implication is that group recommender systems need to model, and hence predict, the three dimensions. For each candidate movie, they need to predict how much each user will like the movie; how satisfied the group members will be with the different ways in which their preferences are traded-off; and the group experience. Our experimental results suggest that it may even be important to be able to predict some sort of 'objective' movie quality, since this was given high importance by the students in the experiment.

One way a recommender can predict these factors is for us to *design* prediction models. Nearly all work on group recommender systems has taken this approach to the prediction of users' satisfaction with the recommendation. This is what the different aggregation functions do, including our own social recommender that takes personalities and trust into account (Section 2). But designing such models is difficult. There is a risk that our models are too simplistic, failing to take into account the richness of group dynamics.

A better approach might be to try to learn these models, using the feedback that we have been discussing to give us training data. This, after all, is how we predict single-user ratings. Why should we not take the same approach to predictions of recommendation satisfaction and of the group experience? An approach that generalises from training data might be more sensitive to nuances in the ways that groups operate. The case-based variant of our group recommender system (Section 2) works in this way —at least, in a simple-minded form: aggregation is based on 'replaying' the decision-making from similar movie-going events. It does not go so far as to predict the group experience.

CBR might be very well-suited to this task. After all, CBR is all about reasoning with experiences [1]. A rich case structure can capture multiple aspects of the movie-going event. The problem description part of the case can contain some or all of the following: (a) information about each member of the group — demographic information, personality information, and information about tastes, e.g. in the form of ratings; (b) information about relationships between group members; (c) the candidate movies, i.e. the ones from which the recommender made its recommendations; (d) predicted ratings for each group member and each candidate movie; and even (e) predictions about the other dimensions (user satisfaction and the group experience). The solution part of the case can contain at least the movie or movies that were recommended and might contain more than this (e.g. the ranking of all the candidate movies). Since groups recur (with small variations) and groups structures (such as a parent and his or her children, or a group of university-age friends) recur, the CBR assumption (similar problems have similar solutions[9]) might apply.

But to make good recommendations, we cannot simply retain cases of this kind in a case base and replay them. The case may be suboptimal; the movie that the group went to see may not have been the best movie for this group. If we retain it, we will replay it in any future recommendation where it gets retrieved as a neighbour, where it may contribute to suboptimal decisions in the future. We need to store information about how successful each case is. Cases can include a third component (alongside the problem description and the solution), namely the outcome [7]. In a recommender system, the outcome records user feedback —the main subject of this paper. The feedback can be compared with predicted values to give a measure of the (sub)optimality of the case.

But there remains a question of practicality. We suspect that users will be either unwilling or unable to give each of the three kinds of feedback. Furthermore, when current group recommender systems ask their uses for a movie rating, it is probable that users do not wholly distinguish between movie ratings (whether they liked the movie), satisfaction with the recommendation (whether it tarded-off preferences in a good way) and the group experience. The move rating they supply is likely to be influenced by the other two factors.[6]

Perhaps if group recommender systems are to ask for only one form of feedback, they should instead ask user for just their rating of the group experience. This is easily understood: "On a scale of 1 to 5 (where 1 means 'Not at all' and 5 means 'A very great deal'), how much did you enjoy your trip to the movies with your friends?" This by no means solves all the problems we face in building

[6] Ratings in single-user recommenders also exhibit contextual influences [8]. But, again, here we are focussing on issues that are specific to, or accentuated in, group recommender systems.

a new generation of group recommender systems. If we ask for only one form of feedback, we then face a *credit assignment problem*: determining how much of their enjoyment (or lack of it) was attributable to various factors, and representing and reasoning with the uncertainty that arises from this credit assignment. Furthermore, in a group recommender, we may have varying degrees of feedback incompleteness: some group members may return to the system and supply a rating; others may not, and this increases uncertainty.

We cannot conclude this paper with a design prescription. But we hope that our reflection on our experience of building a number of group recommender systems, along with some of the insights that come from our experiment, suggest a direction of travel for future work or, at least, will provoke useful discussion.

References

1. Ralph Bergmann. *Experience Management: Foundations, Development Methodology, and Internet-Based Applications.* Springer, 2002.
2. Derek Bridge, Mehmet H. Göker, Lorraine McGinty, and Barry Smyth. Case-based recommender systems. *Knowledge Engineering Review*, 20(3):315–320, 2005.
3. J. Golbeck. Generating predictive movie recommendations from trust in social networks. In *4th International Conference on Trust Management*, pages 93–104, 2006.
4. Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.
5. A. Jameson and B. Smyth. Recommendation to groups. In *The Adaptive Web, Methods and Strategies of Web Personalization*, pages 596–627. Springer, 2007.
6. B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell. Explaining the user experience of recommender systems. *User Model. User-Adapt. Interact.*, 22(4-5):441–504, 2012.
7. Janet Kolodner. *Case-Based Reasoning.* Morgan Kaufmann Publishers Inc., 1993.
8. Y. Koren and R. M. Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 145–186. Springer, 2011.
9. David B. Leake, editor. *Case-Based Reasoning: Experiences, Lessons, and Future Directions.* MIT Press, 1996.
10. Judith Masthoff and Albert Gatt. In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. *User Modeling and User-Adapted Interaction*, 16(3-4):281–319, 2006.
11. L. Quijano-Sánchez, J. A. Recio-García, B. Díaz-Agudo, and G. Jiménez-Díaz. Social factors in group recommender systems. *ACM Transactions on Intelligent Systems and Technology*, 4(1), 2013.
12. F. Ricci, B. Arslan, N. Mirzadeh, and A. Venturini. ITR: A Case-Based Travel Advisory System. In *6th ECCBR*, pages 613–641. Springer, 2002.
13. L. Quijano Sánchez, D. Bridge, B. Díaz-Agudo, and J. A. Recio-García. Case-Based Aggregation of Preferences for Group Recommenders. In *20th ICCBR*, pages 327–341. Springer, 2012.
14. L. Quijano Sánchez, J. A. Recio-García, and B. Díaz-Agudo. User satisfaction in long term group recommendations. In *19th ICCBR*, pages 211–225. Springer, 2011.
15. K.W. Thomas and R.H. Kilmann. *Thomas-Kilmann Conflict Mode Instrument.* Tuxedo, N.Y., 1974.

Chapter 21

A Reusable Methodology for the Instantiation of Social Recommender Systems

21.1 Citation

Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B., A Reusable Methodology for the Instantiation of Social Recommender Systems, in: Procs. of the 25th International Conference on Tools with Artificial Intelligence, ICTAI'13, IEEE Computing Society, Washington DC, USA, pages: 777-782, 2013. ISBN: 978-1-57735-633-2. ICTAI is ranked as CORE B in the CORE ranking.

Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B., A Reusable Methodology for the Instantiation of Social Recommender Systems, in:International Journal on Artificial Intelligence Tools (IJAIT), volume 23, issue 6, World Scientific, 2014. ISSN: 0950-7051. JCR (2013): 3.058. (Computer Science, Artificial Intelligence, 15 out of 121, Q1).

21.2 Contributions covered by this paper

In this paper we have have abstracted the recommendation process and compiled our techniques and algorithms in an organized generic architecture named ARISE. We have also created a set of templates that help developers create social recommender systems in a semi-automatic way. Where the common and key factor in all the different types of recommenders that can be built in all sort of domains using this generic architecture and the social templates is the inclusion of social elements.

2013 IEEE 25th International Conference on Tools with Artificial Intelligence

A Reusable Methodology for the Instantiation of Social Recommender Systems

Lara Quijano-Sánchez, Juan A. Recio-García and Belén Díaz-Agudo
Department of Software Engineering and Artificial Intelligence,
Universidad Complutense de Madrid, Spain
lara.quijano@fdi.ucm.es, jareciog@fdi.ucm.es, belend@fdi.ucm.es

***Abstract*—Social recommender systems exploit the social knowledge available in social networks to provide accurate recommendations. However, their instantiation is not straightforward due to its complexity. To alleviate this development complexity, we propose a methodology based on *templates* that conceptualize the behavior of such applications and can be reused to create several social recommender applications in social networks. This development methodology comprises not only templates but also a generic architecture named ARISE and a collection of software components that provide the required functionality. We prove that our *social templates* speed up and facilitate the development process, and demonstrate the viability of our generic architecture in two different case studies.**

***Keywords*-Templates; Generic Architecture; Social recommenders**

I. INTRODUCTION

Recommender systems are born from the idea of suggesting automatically items to users that they may find appealing (e.g. see [2] for an overview). When users of such systems operate not individually but in groups, recommendations to groups of users appear [11]. In the literature, (e.g. [8], [10], [14]) it was shown that using social network information in addition to feedback data (e.g. ratings) can significantly improve recommendations' accuracy.

Social systems by their definition encourage interaction between users and both online content and other users, thus generating new sources of knowledge for recommender systems. Web 2.0 users explicitly provide personal information and implicitly express preferences through their interactions with others and the system (e.g. commenting, friending, rating, etc.). These various new sources of knowledge can be leveraged to improve recommendation techniques and develop new strategies which focus on social recommendation [16].

Our previous work [21], [18] showed an improvement in the accuracy of recommendations for groups by taking into account social information from the group, namely, the *personality* of the users in the group, and the strength of their connections which we refer to as their *trust*. These techniques and their associated algorithms have been compiled in an organized generic architecture named ARISE (Architecture for Recommendations Including Social Elements) that can be instantiated into any kind of social recommender systems that take into account the personality composition of the group and the social connections between the group members. Finally, we have applied our method of social recommendations to an instantiation of our model in a real-life scenario: *HappyMovie* [19], which is a particular instantiation of our generic ARISE architecture for the movie recommendation domain in the social network Facebook. *HappyMovie* has served us as a use case and experimental environment where we have evaluated our ARISE architecture and our social recommendation methods with real data.

In this paper we present the development methodology to reuse the ARISE architecture and the social recommendation methods. More precisely, we propose a semi-automatic way of designing social recommender systems through *social templates* in a CBR way. Case-based reasoning (CBR) has been used in recommender systems before (e.g. [24]) and explicit parallels between CBR and recommenders have been drawn (e.g. [17]). *Templates* are explicit formalizations that abstract the behaviour of a recommender system and can be reused to instantiate custom applications through, for example, the tools provided by the COLIBRI STUDIO platform [23].

The first contribution of this paper is to prove the usability and acceptance of the set of *social templates* that we have designed for the construction of social recommender applications. We want to prove that when developers use these templates their work is quickened and facilitated thus they prefer to use them than starting a whole project from scratch.

The second goal of this paper is to test the ARISE platform and the *social templates* here proposed by building a second social recommender, that belongs to a different domain from the one that we had already tested (movies): *HappyShopping*. *HappyShopping* is a social recommender system that provides clothing recommendations to people connected through social networks, i.e. clothing recommendations to Facebook users. One of the main ideas followed in our *social recommendation method* and adopted in the development of our applications *HappyMovie* and *HappyShopping* is that everyone is influenced by their social context. Social media highly influences our shopping, relationships, and education. Several researchers study the impact of social media in our lives [5]. The social context, refers to the immediate physical

1082-3409/13 $31.00 © 2013 IEEE
DOI 10.1109/ICTAI.2013.119

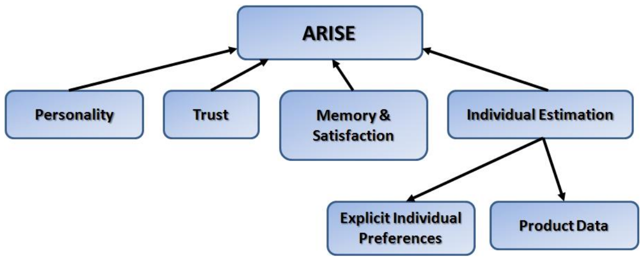


Figure 1. Overview of ARISE

and social setting in which people live. It includes the culture that the individual was educated or lives in, and the people and institutions with whom they interact. Circumstantial life events, influences, and surroundings can further change our behaviour [3].

The paper runs as follows. Section II presents an overview of our architecture ARISE; Section III describes our set of *social templates* as an intermediate step between ARISE and any social application that can be built following ARISE's structure; Section IV describes *HappyShopping* as a case study of instantiation of the templates and gives a preliminary evaluation of the effort and viability of using ARISE and our templates; and Section V concludes and presents some ideas for future work.

II. Generic architecture for group recommenders using social elements

ARISE[1] is a theoretical organization of the modules required to build social recommenders [20]. This architecture allows us to simulate in a more realistic way the decision making process performed by people when choosing an item of their liking. Note that this social architecture is viable not only for group recommender systems, as it was used when building *HappyMovie* [19], but also for individual recommender systems, as it has been used when building *HappyShopping*.

The common and key factor in all the different types of recommenders that can be built in all sort of domains using this generic architecture is the inclusion of *social elements*. These *social elements*, that in our social recommendation method are the personality and trust factors, define each person (our users involved in the recommendation processes) as a potentially influenced component of a social community or group determined by the environment, in most cases social networks, s/he belongs to. In our *social* method, we have simulated people's behaviour based on the idea that the relationship between individuals and their networks of people directly influence their lives [5].

The architecture ARISE is represented in Figure 1. We can see that it is divided in seven different modules: personality, trust, memory & satisfaction, individual estimation, explicit individual preferences, product data, and the ARISE module itself (which is only necessary when using the architecture for group recommendations as we will later explain). Below we summarize each of the modules (further details can be found in [20]):

1) Personality Module: This module fulfils the task of obtaining a value that represents the personality of each user. This personality value, p_u, fits within a range of (0,1], 0 being the reflection of a very cooperative person and 1 the reflection of a very selfish one.

2) Trust Module: This module fulfils the task of obtaining the trust values, $t_{u,v}$, between every user u and v that belong to a common social environment or group. Note that $t_{u,v} \in$ (0,1], 0 being the reflection of a person not to be trusted and 1 the reflection of a highly trusted one.

3) Memory & Satisfaction Module: This module stores all the recommendations that have been made for every user and every group [1]. This avoids repeating past recommendations and also ensures a certain degree of fairness in the long run. We believe that this is a necessary step when providing a whole set of fair recommendations. This way, if one user accepts a proposal that s/he was not interested in, next time s/he will have some kind of priority in the recommendation process.

4) Individual Estimation Module: This module is in charge of computing individual predictions, $pred(u, i)$, for each user u and each item i in the catalogue. The individual predictions, or recommendations, consist on a basic building block of the architecture as our recommendation approach predicts the rating that each user would assign to every item in the catalogue and later, if used for group recommender applications, these estimated ratings are aggregated to obtain a global prediction for the group.

5) Explicit Individual Preferences Module: This module obtains information about the user, which is required to predict the rating for a new item. Commonly, it just consists of ratings given to some products in the catalogue.

6) Product Data Module: This module obtains the catalogue of products to be recommended.

7) ARISE Module: This module is only needed when using the architecture for group recommender systems. It combines all the information provided by the rest of the modules using our social group recommendation methods and offers a recommendation for the group. Space limitations preclude a detailed description of the social group recommendation process but it is described in [21], [20].

III. Templates

As we have introduced, our goal with the development of our *social templates* is to create an intermediate step between our generic architecture ARISE and any social application that can be built following ARISE's structure.

In order to facilitate the architecture instantiation process, we propose a case-based approach where the designer retrieves a system (i.e. our *social templates*) from a library

[1]Architecture for Recommenders Including Social Elements

(case base) of previously designed CBR systems (i.e. social recommenders) and, if needed, adapts it by adding, removing or substituting components in the selected system. Retrieval and adaptation of systems are possible through the use of semantic templates that have been previously abstracted from available systems. Each template is a generalization of several CBR systems and also include semantic annotations from human experts. Templates store the control flow of CBR systems, conceptualizing their behavior, and including the concepts and constraints required to model a number of related systems [22].

The templates that we have created for our *social recommendation method* are composed by *tasks* that identify the steps of the recommender system and *methods* that solve each task with a particular implementation. In this paper we present *generic social templates* that provide a high-level view of a set of final *social templates*. These templates are composed by *generic tasks* and *simple tasks*. Generic tasks encapsulate sequences of simple tasks. Depending on the decomposition of each generic task into sequences of simple tasks, we obtain several final templates.

In order to design our templates we have used the COLIBRI STUDIO Integrated Development Environment[2] that facilitates the creation of templates for CBR systems.

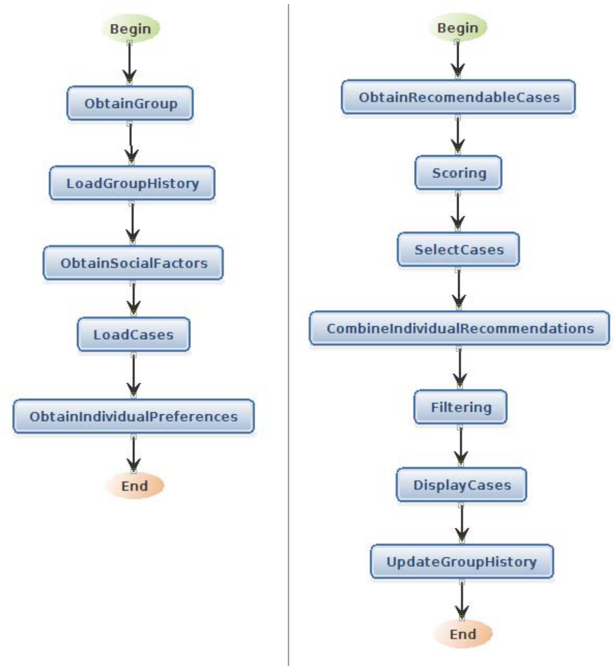


Figure 2. Precycle template (left) and Cycle template(right)

A. ARISE's *Social Templates*

COLIBRI STUDIO comprises a set of tools to instantiate CBR and recommender applications based on the jCOLIBRI framework. This framework provides the basic building blocks required to easily develop such systems. Because jCOLIBRI is aimed at developer users, COLIBRI STUDIO alleviates the programming tasks and provides several graphical tools that can be used to generate automatically CBR systems. The generation of CBR applications in COLIBRI STUDIO is guided by the COLIBRI development process that proposes the reuse of existing designs –templates– of CBR applications and its adaptation to the concrete target system. These templates must follow the conceptual organization of CBR systems stated by the jCOLIBRI framework: a precycle where cases and reasoning resources are loaded; the CBR reasoning cycle itself; and an eventual postcycle step where initial resources are released.

Templates represent, in a conceptual level, the behaviour of a family of CBR systems (such as our social recommenders) but do not provide the functionality required to build them. This functionality is provided by the components in jCOLIBRI or developed on-demand. This way, the templates for building social group recommenders require the components that provide its functionality, and these components are organized in the ARISE's modules. We will see that some of the different tasks of the templates correspond to some of ARISE's modules. Note that each task can be implemented by several different methods, we will here present some of the possible methods that can be used to perform each task. Most of them are already implemented in jCOLIBRI and therefore will facilitate the process of building a new application by using our *social templates* as we will see in Section IV.

Figure 2 left, shows the *pre-cycle template*. The *pre-cycle template* is formed by the following tasks:

1) ObtainGoup: Consists of obtaining the id of each user $u \in G$, being $G = \{u : 1 \ldots n\}$ the active group of users and $|G| > 1$. The active group of users G is defined for social group recommender systems as the people that intend to realize an activity together, and for social individual recommender systems as the people who belong to the circle of trusted people in the social environment of the user receiving the recommendation. For both options, the group is defined in the framework of social networks. Some of the different methods that can be used to obtain G are:

- Through the creation of an event to perform an activity together (social group recommender systems).
- Calculating the group of closest friends in the social network (social individual recommender systems). To do so, the method obtains a trust value (as we will explain bellow in the *ObtainTrustFactors* task) with all the user's friends in Facebook that also use the application being implemented.

2) LoadGroupHistory: (Corresponds to the *Memory & Satisfaction* module in ARISE) Assume a case base CB in

[2]http://www.colibricbrstudio.net

which each case $c \in CB$ records a previous recommendation event. This task consists of retrieving the case c that corresponds to the active user u or group of users G. Note that this task is optional and can be skipped if developers do not want to build a system with memory of past recommendations.

3) ObtainSocialFactors: Consists of a *generic task* that encapsulates the following subtasks:

- **ObtainUsersPersonality**: (Corresponds to the *Personality* module in ARISE) Consists of obtaining the personality of each user u, denoted p_u, by making users complete a personality test on registration with the recommender. This task can be fulfilled by the following methods:
  - The Thomas-Killmann Conflict Mode Instrument (TKI) [27] that proposes 30 situations where the user has to think about how s/he will react. (e.g. as used in [18])
  - TKI's alternative movie metaphor, that consists of displaying two well known movie characters with opposite personalities for each of five possible categories. One character represents the essential characteristics of one category, while the other one represents all the opposite ones. What the user has to do is to choose with which of each pair of characters s/he feels more identified with by simple moving an arrow (e.g. as used in *HappyMovie* [19] and in *HappyShopping*).
  - Any other personality test from which the p_u can be defined as a single numeric value.
- **ObtainTrustFactors**: (corresponds to the *Trust* module in ARISE) Consist of obtaining the trust $t_{u,v}$ between users u and v ($u \neq v \in G$). It can be based on distance in the social network, the number of friends in common, relationship duration, and so on.

4) LoadCases: Consists of obtaining all the items i in the domain catalogue $I = \{i : 1 \ldots m\}$.

5) ObtainIndividualPreferences: (Corresponds to the *Explicit Individual Preferences* module in ARISE) Consists of obtaining the ratings $r_{u,i}$ that each user u in G assigns to items i in I. Ratings are on a numeric scale, e.g. 1 = terrible and 5 = excellent.

Now, we continue the templates explanation with the *cycle template* shown in the Figure 2 right. This template is principally designed for its use in the process of building social group recommender systems, however it can also be used for social individual recommenders leaving the last 4 tasks unimplemented. The *cycle template* is formed by the following tasks:

6) ObtainRecommendableCases: (Corresponds to the *Product Data* module in ARISE) Consists of obtaining all the candidate target items i in the recommendation catalogue $T = \{i : 1 \ldots t\}$.

7) Scoring: (Corresponds to the *Individual Estimation* module in ARISE) Consists of obtaining predicted ratings $pred(u, i)$ for each active user $u \in G$ and target item $i \in T$. Some of the different methods that can be used to implement this task are:

- *Collaborative recommenders* [7], [12], [9], use the ratings already assigned by the users to several products. Users are selected according to their similarity with the individual receiving the recommendation (by comparing the ratings given to the products). Most similar users are used as predictors and their ratings are combined to estimate the rating that the target user would assign to a new product.
- *Content-based recommenders* [13], compare each item in the catalogue with the items already rated by the target user. Then the ratings of the most similar rated items are combined to provide a recommendation.
- *Hybrid recommenders* [4], that are a combination of the two previous ones.
- *Asking other G users to give an estimated rating for the product i* [6], this method relies heavily on explicit feature-level feedback from users.
- *Influence based recommenders* [21], [18], modify the non-social predictions $pred(u, i)$ obtained with one of the above methods with the personality and trust factors. We will detail this method as it is the one that is used for *HappyShopping*'s recommendations as we will see in Section IV. It supposes that the user may modify her/his preference for an item depending on the preferences given by her/his friends to the same item. For example, if our rating for an item is 3 and our friend has a 5 rating for the same item, we could think on modifying our rating to 4. Depending on the trust in this friend, we decide the level of variation for our rating (i.e. 3.5 if the trust is low, and 4.5 if trust is high). Furthermore, the variation of our rating also depends on our personality. If we have a strong personality (high personality value) we will not be willing to change our rating, but if we have a weak personality (low value) we could be easily influenced by other users.

  The method combines the personality and trust factors using the following equation:

  $$ibr(u, i) = pred(u, i) + (1 - p_u) \cdot \frac{\sum_{v \neq u \in G} t_{u,v} \cdot (\, pred(v, i) - pred(u, i)\,)}{|G| - 1} \quad (1)$$

  In this equation, $pred(u, i)$ is modified according to its difference with the ratings of other users ($pred(v, i) - pred(u, i)$). This difference is weighted with the trust between users ($t_{u,v}$), where $v \in G$, being G the group formed by the people who would have the most influence in user u, and therefore, the people who s/he trusts the most. Finally, the accumulated difference is weighted according to u's personality in an inverse way $(1 - p_u)$.

8) SelectCases: Consists of selecting for each active user $u \in G$ the k items from T whose predicted ratings are highest. For example, in *HappyMovie* and in *HappyShopping*, we use $k = 3$. Note that the next 3 tasks are specific for social group recommendations, and therefore the method that implements this task will need to have a *display cases* option for implementations of just individual recommenders.

9) CombineIndividualRecommendations: Consists of obtaining the group prediction $gpred$ aggregating the predicted ratings of the members of the group, $pred(u, i)$ for each $u \in G$ and $i \in T$ (see Equation 2). Possible aggregation functions ($\bigsqcup$ in the equation) include *least misery* (where the minimum is taken) and *most pleasure* (where the maximum is taken). Methods for aggregating ratings are reviewed in [15]. It is the *most pleasure* principle that we used in *HappyMovie*.

$$gpred(G, i) = \bigsqcup_{\forall u \in G} pred(u, i) \tag{2}$$

However, in our social group recommendation method [21], [18], we modify the individual ratings with the personality and trust factors. This way, we modify the impact of individual preferences as shown in Equation 3.

$$gpred(G, i) \;=\; \bigsqcup_{\forall u \neq v \in G} f(\,pred(v, i)\,,\, p_u\,,\, t_{u,v}\,) \tag{3}$$

where $gpred(G, i)$ is the group rating prediction for a given item i; $pred(v, i)$ is the original individual prediction for user v and item i; p_u is the personality value for user u and $t_{u,v}$ is the trust value between users u and v.

There are several methods to modify the rating predicted for a user according to personality and trust factors. It is represented as the $f()$ function in the Equation 3, some of these ways are the *delegation-based method* or the *influence-based method* (see Equation 1) among others. We point interested readers to [21] were several of these social group recommendation methods are detailed.

10) Filtering: Consists of selecting the k' items in T that have the highest predicted ratings for the group. For example, in *HappyMovie*, we used $k' = 3$.

11) DisplayCases: Consists of displaying to each user u receiving the recommendation the k' items obtained by the group recommender.

12) UpdateGroupHistory: Consists of revising the case c that corresponds to the active user u (for individual recommenders) or the active group of users G (for group recommenders) with the new recommendation and retaining it in the case base CB for future recommendations. Note that this task is optional and can be skipped if developers do not want to build a system with memory of past recommendations.

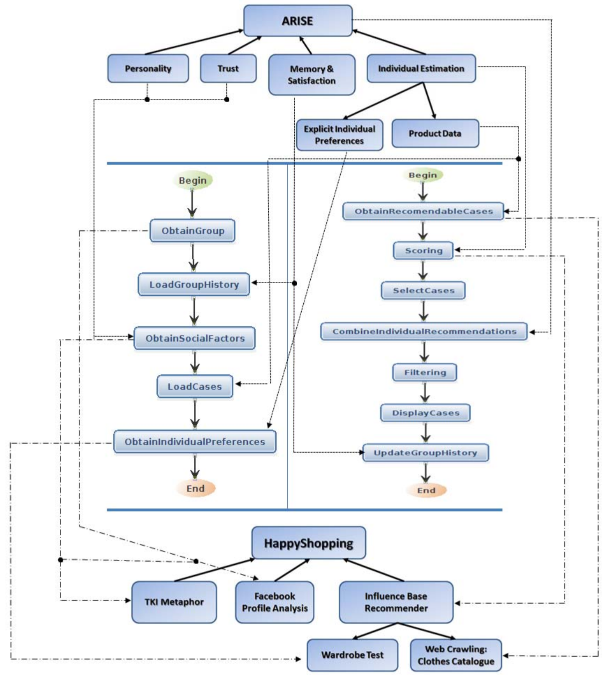


Figure 3. Relationship between the ARISE architecture, the proposed templates and its instantiation in *HappyShopping*

IV. *HappyShopping*

In this section we present *HappyShopping*[3]: a Facebook social individual recommender application for clothes that follows our generic architecture ARISE and has been built using our *social templates*. With the development of this application we study and prove the two goals of this paper:

- The usability of our templates (detailed in Section III-A).
- The viability of our generic architecture ARISE (detailed in Section II) in other domains.

In Sections II and III we have detailed how to design a social recommender using ARISE and how to implement it using our social templates, hence, we will only detail now the concrete choices of domain and development that delimit *HappyShopping*. To understand how each module from the ARISE architecture is defined in one task of the templates and how *HappyShopping* implements the methods of the needed tasks of the templates, we introduce Figure 3. In the top of the Figure we see ARISE's modules, each line that goes out of a module points the concrete task in the *social templates* that corresponds it and each line that comes out of a task in the templates points the concrete module in *HappyShopping*'s structure that implements it.

[3]http://www.happyshopping.es/

A. *Details of* HappyShopping

Traditional recommender systems do not take into consideration explicit social relations among users, yet the importance of social influence in product marketing has long been recognized [26]. Intuitively, when we want to buy a product that is not familiar, we often consult with our friends who have already had experience with the product, since they are those that we can reach for immediate advice. When friends recommend a product to us, we also tend to accept the recommendation because their inputs are trustworthy. *HappyShopping* exploits this fact and takes into account preferences of the users' closest friends in order to recommend which piece of clothing users should purchase and later propose an argumentation process with these closest friends about the recommended items. *HappyShopping*'s main goal is to present a recommender system that proposes pieces of clothing by taking into account users social context. The recommendation process is summarized in the steps below:

- **Product Comparison with user preferences**: The application requires the user to explicitly identify products that are of her/his interest, which will form the users' "wardrobe".
- **Product Comparison with the preferences of most influential friends**: In this step we model the impact of the preferences of the people influencing the user that is being recommended. The proximity between users (users trust) is obtained by analyzing the information available on the social network: messages exchanged, shared photos, etc.
- **Weighting of items regarding the degree of influence of individuals**: The influence of other group members not only depends on their proximity or trust in them, but also in the degree of personality or leadership of these influencers. In this step the products to propose are reconsidered depending on these factors. This process requires obtaining the personality information from the social network.

Using the* HappyShopping *system: Users start their Facebook account and look for *HappyShopping* in the applications section. *HappyShopping*'s main page is shown in Figure 4. The required steps to obtain a clothing recommendation with *HappyShopping* are explained below:

- **Creating the user profile in the application:** Before any user can access the clothing recommendation results users have to create their individual "recommendation profile" which is necessary for our recommendation method. This profile is based on three different aspects: personality, individual preferences and trust in other users.
  - To obtain the personality users have to choose a series of characters to whom they feel identified, Figure 5 up shows *HappyShopping*'s personality test implementation. This step corresponds to the *Personality* module in ARISE and its solved by *ObtainUsersPersonality* task in our template. The concrete method that implements this task is the *TKI's alternative movie metaphor* explained in Section III-A.
  - To obtain the preferences profile users have to rate a set of clothes (at least 20 pieces), where they enter their personal preferences, Figure 5 bottom shows *HappyShopping*'s preferences test implementation. This step corresponds to the *Explicit Individual Preferences* module in ARISE and its solved by *ObtainIndividualPreferences* task in our template. The specific pieces that are displayed for the user to rate (users can rate 100 pieces at the most) are selected automatically from *HappyShopping*'s catalogue trying to maximize the diversity. To do so, a similar metric as the one presented in the system ExpertClerk [25] is used.
  - To obtain the trust, the application reads the information stored in Facebook personal profiles. It calculates the trust that the user has in all the other users in her/his close circle (G). To obtain the circle of trusted people in the social environment of the user receiving the recommendation, the application needs to calculate which other application users should form the group G. This step is solved by the task *ObtainGroup* in our template. The concrete method that implements this task is the *Calculating the group of closest friends in the social network* explained in Section III-A. Note that the trust value is general to friends and not specific to the domain (clothes) as it has been proven most efficient in [21], [18].

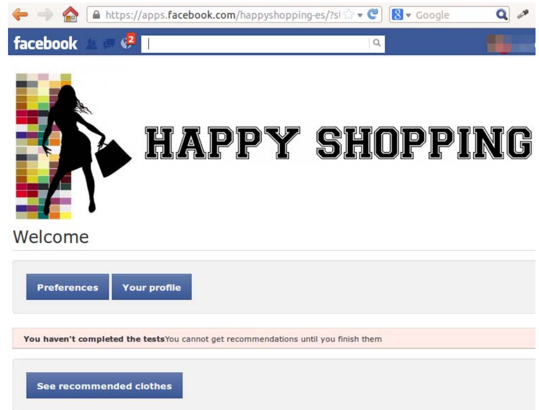


Figure 4. *HappyShopping*'s main page

- **Recommendation:** Once the application has obtained the factors that identify each user receiving a recommendation (personality, individual preferences and trust

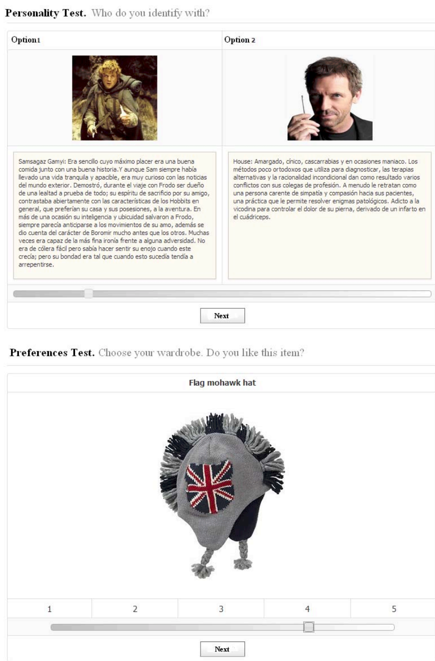


Figure 5. Personality test (up) and Preferences test (bottom) in *HappyShopping*

in other users) user are able to click the "See recommended clothes" bottom (see Figure 4) and see their individual recommendations. This step corresponds to the *Individual Estimation* module in ARISE and its solved by the *Scoring* task in our template. The concrete method that implements this task is the *Influence based recommenders* explained in Section III-A.

- **Once the recommendation is made:** *HappyShopping* provides a list of the best 3 pieces of clothing that the recommender has found in the catalogue. For each of them the user will have two options:
  - Purchase the product. Note that this function is not part of the application.
  - Start an argumentation process with the G members. Where the user will ask her/his closest friends which piece of clothing fits her/him best. Note that this option has not been implemented yet as it is not part of the *social templates*.

HappyShopping counts with a catalogue of 1887 pieces. This catalogue has been obtained by parsing the web searching for different types of clothes and styles to wear. Each item in the catalogue is formed by a picture of the piece of clothing plus the concrete characteristics of the piece like material, colour, style, size, prize, etc.

B. Evaluation of using our social templates and ARISE

Regarding the effort and viability of using ARISE and our *social templates* for the development of *HappyShopping*, we have counted with 3 developers. The skills and background of *HappyShopping*'s developers are summarized in Figure 6, that reflects the average of the answers given by the 3 developers to a questionnaire about how they grade themselves. These developers have reused our generic architecture ARISE and its associated templates. We have questioned them about the usability of the set of templates and they all answered that the templates had facilitated and quickened their work thus they all preferred to have the templates to assist them. About the effort that they put on the construction of *HappyShopping*, we asked them how long it took them to build an initial version of the application, they answered that it took them 5 weeks to develop an initial version and 10 weeks to develop the final version of *HappyShopping*. If we compare these results with the time that took us (the authors) to develop *HappyMovie* (which is our other social recommender application in the movies domain as introduced in the previous sections) we can conclude that the usage of our social templates and ARISE has been a success. It took us more than 5 months to develop *HappyMovie*, and we were 3 expert programmers specialized in CBR and recommender systems. Obviously, as it was the first time that the social recommender system was being implemented there was a high cost in the design and development of *HappyMovie*, which has been captured in the social templates and the generic architecture and makes the cost of a second social recommender application descend. Therefore, we consider that the use of our social templates and ARISE indeed facilitates and eases the construction of other social recommender applications.

When you started the development of the application, How do you define your knowledge in? :	**Developers answer**
Developper skills	4
Programming recommender systems	2
Programming CBR systems	1
Facebook programming	1
Programming with COLIBRI STUDIO	2

Figure 6. Questionnaire answers of *HappyShopping*'s developers about their skills and background. Answers are in a scale 0 to 5. Being 0 very little and 5 a lot.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a generic architecture ARISE and a set of templates that formalize the behaviour of social recommender systems. We have proven ARISE's suitability by building two different recommending applications in two different domains. *HappyMovie* is a particular instantiation of ARISE for recommending movies to groups of people connected through the social network Facebook. The second case study, *HappyShopping*, is an individual recommender system that follows our method of making recommendations to people using their social information stored in the social network Facebook. We have also presented a set of templates that represent an intermediate step in the development of social group recommender applications and proven that they quicken and facilitate the process of building new applications. Thus, developers prefer to use these templates than starting a new application from scratch. There is much that can be done to take this work forward. For us, the next step is taking *HappyShopping* one step forward and make it a richer application with actual ratings from users, from which we hope to gather data and use it as the basis for future experiments. We also want to develop the argumentation process step of the application that was mentioned in Section **??**.

ACKNOWLEDGMENT

Supported by Spanish Ministry of Science & Education (TIN2009-13692-C03-03) and Madrid Education Council and UCM (Group 910494) and Spanish Ministry of Economy and Competitiveness under grant IPT-2011-1890-430000.

REFERENCES

[1] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variants, and system approaches. *Artificial Intelligence Communications*, 7(1):39–59, 1994.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749, 2005.

[3] M. Bazire and P. Brézillon. Understanding context before using it. In *Context*, pages 29–40, 2005.

[4] R. D. Burke. Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact.*, 12(4):331–370, 2002.

[5] N. A. Christakis and J. H. Fowler. Social contagion theory: Examining dynamic social networks and human behavior. *CoRR*, 2011.

[6] E. Costello, J. Doody, L. McGinty, and B. Smyth. *i*care: intelligent customer assistance for recommending eyewear. In *IUI*, pages 282–284, 2006.

[7] M. D. Ekstrand, J. Riedl, and J. A. Konstan. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, 4(2), 2011.

[8] J. Golbeck. Generating predictive movie recommendations from trust in social networks. In *iTrust: 4th International Conference on Trust Management*, 2006.

[9] J. L. Herlocker, J. A. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Inf. Retr.*, 5(4), 2002.

[10] M. Jamali and M. Ester. Using a trust network to improve top-n recommendation. In *RecSys*, pages 181–188, 2009.

[11] A. Jameson and B. Smyth. Recommendation to groups. In *The Adaptive Web, Methods and Strategies of Web Personalization*, pages 596–627. Springer, 2007.

[12] Y. Koren and R. M. Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 145–186. 2011.

[13] P. Lops, M. de Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*. 2011.

[14] P. Massa and P. Avesani. Trust-aware recommender systems. In *RecSys*, pages 17–24, 2007.

[15] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, 14(1):37–85, 2004.

[16] B. Mobasher, D. Jannach, W. Geyer, and A. Hotho. 4th acm recsys workshop on recommender systems and the social web. In *RecSys*, pages 345–346, 2012.

[17] D. O'Sullivan, D. C. Wilson, and B. Smyth. Improving case-based recommendation: A collaborative filtering approach. In *ECCBR*, pages 278–291. Springer, 2002.

[18] L. Quijano-Sánchez, J. A. Recio-García, and B. Díaz-Agudo. Personality and social trust in group recommendations. In *ICTAI'10.*, pages 121–126. IEEE Computing Society, 2010.

[19] L. Quijano-Sánchez, J. A. Recio-García, and B. Díaz-Agudo. Happymovie: A facebook application for recommending movies to groups. In *ICTAI'11.*, pages 239–244, 2011.

[20] L. Quijano-Sánchez, J. A. Recio-García, and B. Díaz-Agudo. An architecture and functional description to integrate social behaviour knowledge into group recommender systems. In *Applied Intelligence*. in press, 2013.

[21] L. Quijano-Sánchez, J. A. Recio-García, B. Díaz-Agudo, and G. Jiménez-Díaz. Social factors in group recommender systems. In *ACM Transactions on Intelligent Systems and Technology*, 2013.

[22] J. A. Recio-García, B. Díaz-Agudo, and P. A. González-Calero. Semantic templates for case-based reasoning systems. *Knowledge Eng. Review*, 24(3):245–264, 2009.

[23] J. A. Recio-García, P. A. González-Calero, and B. Díaz-Agudo. Template-based design in colibri studio. *Information Systems*, (0):In press, 2012.

[24] F. Ricci, B. Arslan, N. Mirzadeh, and A. Venturini. ITR: A case-based travel advisory system. In S. Craw and A. Preece, editors, *ECCBR*, pages 613–641. Springer, 2002.

[25] H. Shimazu. Expertclerk: A conversational case-based reasoning tool for developing salesclerk agents in e-commerce webshops. *Artif. Intell. Rev.*, 18, 2002.

[26] M. R. Subramani and B. Rajagopalan. Knowledge-sharing and influence in online social networks via viral marketing. *Commun. ACM*, 46(12):300–307, 2003.

[27] K.W. Thomas and R.H. Kilmann. *Thomas-Kilmann Conflict Mode Instrument*. Tuxedo, N.Y., 1974.

International Journal on Artificial Intelligence Tools
Vol. 23, No. 6 (2014) 1460031 (25 pages)

DOI: 10.1142/S0218213014600318

World Scientific
www.worldscientific.com

A Reusable Methodology for the Instantiation of Social Recommender Systems

Lara Quijano-Sánchez, Juan A. Recio-García and Belén Díaz-Agudo
Department of Software Engineering and Artificial Intelligence
Universidad Complutense de Madrid, Spain
lara.quijano@fdi.ucm.es, jareciog@fdi.ucm.es, belend@fdi.ucm.es

Received 28 April 2014
Accepted 26 September 2014
Published 26 December 2014

Social recommender systems exploit social knowledge available in social networks to provide accurate recommendations. However, their instantiation is not straightforward due to its complexity. To alleviate this development complexity, we propose a methodology based on *templates* that conceptualize the behavior of such applications and can be reused to create several social recommender applications in social networks. This development methodology comprises not only templates but also a generic architecture named ARISE and a collection of software components that provide the required functionality. We prove that our *social templates* speed up and facilitate the development process, and demonstrate the viability of our generic architecture in two different case studies.

Keywords: Templates; generic architecture; social recommenders.

1. Introduction

Recommender systems are born from the idea of suggesting automatically items to users that they may find appealing (e.g. see Ref. 3 for an overview). When users of such systems operate not individually but in groups, recommendations to groups of users appear.[19]

In the literature, (e.g. Refs. 13, 17 and 24) it was shown that using social network information in addition to feedback data (e.g. ratings) can significantly improve recommendations' accuracy. Social systems by their definition encourage interaction between users and both online content and other users, thus generating new sources of knowledge for recommender systems. Web 2.0 users explicitly provide personal information and implicitly express preferences through their interactions with others and the system (e.g. commenting, friending, rating, etc.). These various new sources of knowledge can be leveraged to improve recommendation techniques and develop new strategies which focus on social recommendation.[28]

Social relations provide an independent source for recommendation; various approaches have been proposed to build social recommender systems such as trust ensemble,[23] trust propagation,[18] or directly trust user based recommenders.[45] Besides, there is recent work reporting significant recommendation performance improvement for social recommender systems.[13,29,31,44,5,16,40]

Our previous work[34,32] showed an improvement in group recommendations' accuracy when taking into account social group information, namely, group users' *personality*, and the strength of their connections which we refer to as their *trust*. These techniques and their associated algorithms have been compiled in an organized generic architecture named ARISE (Architecture for Recommendations Including Social Elements) that can be instantiated into any kind of social recommender system that takes into account personality composition inside a group and social connections between group members. Finally, we have applied our method of social recommendations to an instantiation of our model in a real-life scenario: *HappyMovie*,[33] which is a particular instantiation of our generic ARISE architecture for the movie recommendation domain in the social network Facebook. *HappyMovie* has served us as a use case and experimental environment where we have evaluated our ARISE architecture and our social recommendation methods with real data.

In this paper we present the development methodology to reuse the ARISE architecture and its social recommendation methods. More precisely, we propose a semi-automatic way of designing social recommender systems through *social templates* in a CBR way. Case-based reasoning (CBR) has been used in recommender systems before (e.g. Ref. 39) and explicit parallels between CBR and recommenders have been drawn (e.g. Ref. 30). *Templates* are explicit formalizations that abstract the behaviour of a recommender system and can be reused to instantiate custom applications through, for example, the tools provided by COLIBRI STUDIO platform.[38]

The first contribution of this paper is to prove the usability and acceptance of the set of *social templates* that we have designed for the construction of social recommender applications. We want to prove that when developers use these templates their work is quickened and facilitated thus they prefer to use them rather than starting a whole project from scratch.

The second goal of this paper is to test the ARISE platform and *social templates* here proposed by building a second social recommender that belongs to a different domain from the one that we had already tested (movies): *HappyShopping*. *HappyShopping* is a social recommender system that provides clothing recommendations to people connected through social networks, i.e. clothing recommendations to Facebook users. One of the main ideas followed in our *social recommendation method* and adopted in the development of our applications *HappyMovie* and *HappyShopping* is that everyone is influenced by their social context. Social media highly influences our shopping, relationships, and education. Several researchers study the impact of social media in our lives.[9] The social context refers to the immediate physical and social setting in which people live. It includes the culture that the individual was educated or lives in, and the people and institutions with whom s/he interacts. Circumstantial life events, influences and surroundings can further change our behaviour.[6]

The paper runs as follows. Section 2 presents an overview of our architecture ARISE; Section 3 describes our set of *social templates* as an intermediate step between ARISE and any social application that can be built following ARISE's structure; Section 4 describes *HappyShopping* as a case study of instantiation of the templates and gives an evaluation of the effort and viability of using ARISE and our templates; Finally Section 5 concludes and presents some ideas for future work.

2. Generic Architecture for Group Recommenders Using Social Elements

ARISE[a] is a theoretical organization of the modules required to build social recommenders.[36] This architecture allows us to simulate in a more realistic way the decision making process performed by people when choosing an item of their liking. Note that this social architecture is viable not only for group recommender systems, as it was used when building *HappyMovie*,[33] but also for individual recommender systems, as it has been used when building *HappyShopping*.

The common and key factor in all the different types of recommenders that can be built in all sort of domains using this generic architecture is the inclusion of *social elements*. These *social elements*, that in our social recommendation method are the *personality* and *trust* factors, define each person (our users involved in the recommendation processes) as a potentially influenced component of a social community or group determined by the environment, in most cases social networks, s/he belongs to. In our *social* method, we have simulated people's behaviour based on the idea that the relationship between individuals and their networks of people directly influence their lives.[9]

The architecture ARISE is represented in Figure 1. We can see that it is divided in seven different modules: personality, trust, memory & satisfaction, individual estimation, explicit individual preferences, product data, and the ARISE module itself (which is only necessary when using the architecture for group recommendations as we will later explain). Below we summarize each of the modules (further details can be found in Ref. 36):

(1) *Personality module*

This module fulfils the task of obtaining a value that represents the personality of each user. This personality value, p_u, fits within a range of $(0, 1]$, 0 being the reflection of a very cooperative person and 1 the reflection of a very assertive one. This personality value serves us to identify different behaviors that people have in conflict situations. This follows the common belief that when people face situations

[a]Architecture for Recommenders Including Social Elements.

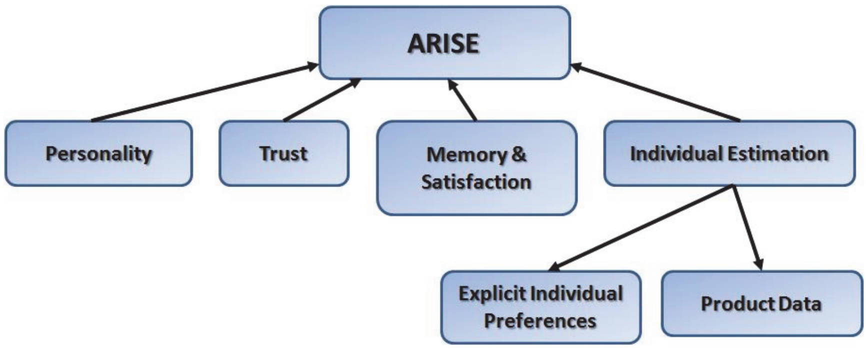


Fig. 1. Overview of ARISE.

where their interests or preferences are incompatible with others *conflict situations* arise. Here conflict is understood as a difference that prevents agreement. More concretely, in group interactions it is defined as a competitive or opposing action of incompatibles: antagonistic state or action (as of divergent ideas, interests, or persons).[1] According to Thomas-Kilmann's study[43] users that present a low personality value ($p_u < 0.4$) are considered *cooperative*, which reflects highly tolerant people, meaning that even if the selected item is not the one of their choice, it is good enough for them if a trusted source recommends it. On the other hand, users that present a high personality value ($p_u > 0.6$) are considered *assertive*, which reflect more selfish people, meaning that other people's choices do not satisfied their personal concerns.

(2) *Trust module*

This module fulfils the task of obtaining the trust values, $t_{u,v}$, between every user u and v that belong to a common social environment or group. Note that $t_{u,v} \in (0, 1]$, 0 being the reflection of a person not to be trusted and 1 the reflection of a highly trusted one. Inter-personal trust or social tie between users can be estimated following different approaches, being most of them manual,[14,12] task that users resented and found very tedious. Hence, in ARISE we propose its elicitation from Social Networks. The process consists of calculating the inter-personal trust by analysing users' profiles and interactions in a social network. For example, users in Facebook post a huge amount of personal information that can be extracted to compute the trust with other users: likes and interests, personal preferences, pictures, games, etc. Details about our trust computation can be found in Ref. 34.

(3) *Memory & satisfaction module*

This module stores all the recommendations that have been made for every user and every group.[2] This avoids repeating past recommendations and also ensures a certain degree of fairness in the long run. We believe that this is a necessary step when providing a whole set of fair recommendations. This way, if one user accepts a proposal that s/he was not interested in, next time s/he will have some kind of priority in the recommendation process.[35] This means that her/his opinion will have a higher weight next time. These weights will also be influenced by the different personalities of each user. For example, in the concrete case of a group cinema outing a user who dislikes the movie (gives it a low rating) may nevertheless be satisfied with the group decision, especially if s/he appreciates that it has been necessary to balance conflicting interests. Her/his satisfaction might be all the greater if s/he has a more accommodating (less selfish) personality type, or if the recommendation better matches the tastes of group members with whom s/he has stronger connections through contagion and conformity.[26] This behaviour is modelled by immediately compensating users who have been negatively affected and have strong personalities and bearing in mind that users with mild personalities might not mind giving in several times.

(4) *Individual estimation module*

This module is in charge of computing individual predictions, $\mathrm{pred}(u, i)$, for each user u and each item i in the catalogue. The individual predictions, or recommendations, consist on a basic building block of the architecture as our recommendation approach predicts the rating that each user would assign to every item in the catalogue and later, if used for group recommender applications, these estimated ratings are aggregated to obtain a global prediction for the group.

(5) *Explicit individual preferences module*

This module obtains information about user's preferences, which is needed so that the recommender is able to make an estimation of which item from a whole catalogue best suits the user. Commonly, it just consists of ratings given to some products in the catalogue.

(6) *Product data module*

This module obtains the catalogue of products to be recommended.

(7) *ARISE module*

This module is only needed when using the architecture for group recommender systems. It combines all the information provided by the rest of the modules using our *Social Group Recommendation* method and offers a recommendation for the group. A detailed description of our *Social Group Recommendation* method can be later found in Section 3.1.9 and also in Refs. 34 and 36.

3. Templates

As we have introduced, our goal with the development of our *social templates* is to create an intermediate step between our generic architecture ARISE and any social application that can be built following ARISE's structure.

In order to facilitate the architecture instantiation process, we propose a case-based approach where the designer retrieves a system (i.e. our *social templates*) from a library (case base) of previously designed CBR systems (i.e. social recommenders) and, if needed, adapts it by adding, removing or substituting components in the selected system. Retrieval and adaptation of systems are possible through the use of semantic templates that have been previously abstracted from available systems. Each template is a generalization of several CBR systems and also include semantic annotations from human experts. Templates store the control flow of CBR systems, conceptualizing their behavior, and including the concepts and constraints required to model a number of related systems.[37]

The templates that we have created for our *social recommendation method* are composed by *tasks* that identify the steps of the recommender system and *methods* that solve each task with a particular implementation. In this paper we present *generic social templates* that provide a high-level view of a set of final *social templates*. These templates are composed by *generic tasks* and *simple tasks*. *Generic tasks* encapsulate sequences of *simple tasks*. Depending on the decomposition of each *generic task* into sequences of *simple tasks*, we obtain several *final templates*.

In order to design our templates we have used COLIBRI Studio Integrated Development Environment[b] that facilitates the creation of templates for CBR systems.

3.1. *ARISE's social templates*

COLIBRI Studio comprises a set of tools to instantiate CBR and recommender applications based on the jCOLIBRI framework. This framework provides the basic building blocks required to easily develop such systems. Because jCOLIBRI is aimed at developer users, COLIBRI Studio alleviates the programming tasks and provides several graphical tools that can be used to generate automatically CBR systems. The generation of CBR applications in COLIBRI Studio is guided by the COLIBRI development process that proposes the reuse of existing designs — templates — of CBR applications and its adaptation to the concrete target system. These templates must follow the conceptual organization of CBR systems stated by the jCOLIBRI framework: a precycle where cases and reasoning resources are loaded; the CBR reasoning cycle itself; and an eventual postcycle step where initial resources are released.

Templates represent, in a conceptual level, the behaviour of a family of CBR systems (such as our social recommenders) but do not provide the functionality required to build them. This functionality is provided by the components in jCOLIBRI or developed on-demand. This way, templates for building *Social Group Recommenders* require different components that provide its functionality, and these components are organized in ARISE's modules. We will see that some of the different *templates's tasks* correspond to some of ARISE's modules. Note that

[b]http://www.colibricbrstudio.net

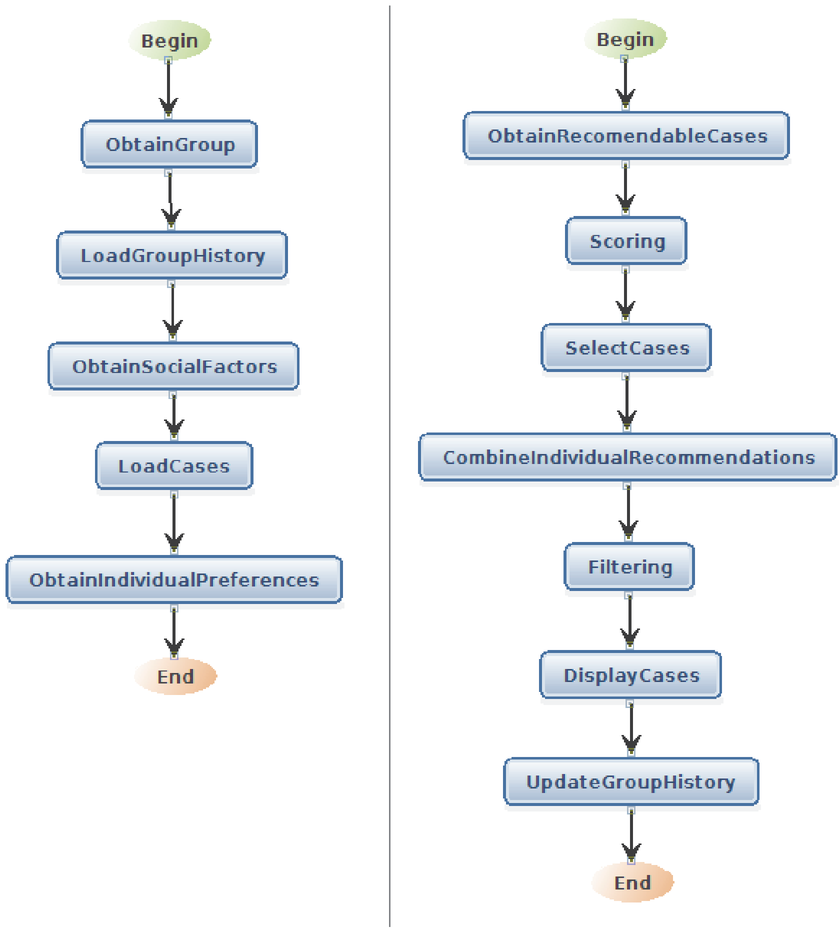


Fig. 2. Precycle template (left) and Cycle template (right).

each *task* can be implemented by several different *methods*, we will here present some of the possible *methods* that can be used to perform each *task*. Most of them are already implemented and therefore will facilitate the process of building a new application by using our *social templates* as we will see in Section 4.

Figure 2 left, shows the *pre-cycle template*. The *pre-cycle template* is formed by the following *tasks*[c]:

3.1.1. *ObtainGoup*

Consists of obtaining the *id* of each user $u \in G$, being $G = \{u : 1 \ldots n\}$ the active group of users and $|G| > 1$. The active group of users G is defined for social group

[c]When using our designed *social templates* along with COLIBRI Studio developers have access to at least one implemented *method* for each of the presented options.

recommender systems as the people that intend to realize an activity together, and for social individual recommender systems as the people who belong to the circle of trusted people in the social environment of the user receiving the recommendation. For both options, the group is defined in the framework of social networks. Some of the different *methods* that can be used to obtain G are:

- Through the creation of an event to perform an activity together (social group recommender systems).
- Calculating the group of closest friends in the social network (social individual recommender systems). To do so, the method obtains a trust value (as we will explain below in the *ObtainTrustFactors task*) with all the user's friends in Facebook that also use the application being implemented.

3.1.2. *LoadGroupHistory*

(Corresponds to *Memory & Satisfaction* module in ARISE) Assume a case base CB in which each case $c \in CB$ records a previous recommendation event. This *task* consists of retrieving the case c that corresponds to the active user u or group of users G. Note that this *task* is optional and can be skipped if developers do not want to build a system with memory of past recommendations.

3.1.3. *ObtainSocialFactors*

Consists of a *generic task* that encapsulates the following *subtasks*:

- **ObtainUsersPersonality**: (Corresponds to *Personality* module in ARISE) Consists of obtaining the personality of each user u, denoted p_u, by making users complete a personality test on registration with the recommender. This *task* can be fulfilled by the following methods[d]:
  - — The Thomas-Killmann Conflict Mode Instrument (TKI)[43] that proposes 30 situations where the user has to think about how s/he will react. (e.g. as used in Ref. 32)
  - — TKI's alternative movie metaphor, that consists of displaying two well known movie characters with opposite personalities for each of five possible categories. One character represents the essential characteristics of one category, while the other one represents all the opposite ones. What the user has to do is to choose with which of each pair of characters s/he feels more identified with by simple moving an arrow (e.g. as used in *HappyMovie*[33] and in *HappyShopping*).
  - — Any other personality test from which the p_u can be defined as a single numeric value.

[d]The first two of them are already implemented in our templates so developers who wish to use them only need to select them when using our designed *social templates* along with COLIBRI STUDIO.

- **ObtainTrustFactors**: (Corresponds to *Trust* module in ARISE) Consist of obtaining the trust $t_{u,v}$ between users u and v ($u \neq v \in G$). It can be based on distance in the social network, the number of friends in common, relationship duration, and so on. For example, Ref. 12 identified 74 Facebook variables as potential predictors of tie strength.[e] On the other hand, in Ref. 34 we presented a method (offered in our templates as *implemented method*) to compute $t_{u,v}$ by automatically eliciting needed variables from Facebook.

3.1.4. *LoadCases*

Consists of obtaining all the items i in the domain catalogue $I = \{i : 1 \ldots m\}$.

3.1.5. *ObtainIndividualPreferences*

(Corresponds to *Explicit Individual Preferences* module in ARISE) It can be based on a combination of implicit data, i.e. according to the user's patterns of use (e.g. Refs. 4 and 46) or, explicit data, where the user briefly, and throughout usage, specifies their preferences to the system (e.g. Refs. 7, 27 and 33). For example, a system which sells books may recommend new books for a user to buy based on which books they have looked at or bought in the past (implicit rating), or how they have actively rated books (explicit rating). The *implemented method* that we provide for this *task* consists of obtaining the ratings $r_{u,i}$ that each user u in G assigns to items i in I. Ratings are on a numeric scale, e.g. 1 = terrible and 5 = excellent.

Now, we continue templates explanation with the *cycle template* shown in the Figure 2(right). This template is principally designed for its use in *social group recommender* systems building processes, however it can also be used for *social individual recommenders* leaving the last 4 *tasks* unimplemented. The *cycle template* is formed by the following *tasks*:

3.1.6. *ObtainRecommendableCases*

(Corresponds to *Product Data* module in ARISE) Consists of obtaining all the candidate target items i in the recommendation catalogue $T = \{i : 1 \ldots t\}$. For example, for *HappyMovie* we built a *Web Crawler* that searches a leisure guide web[f] and retrieves all the movies and movie sessions being displayed in Spain's cinemas. We provide this *Web Crawler* as a *method* that implements this *task* as it can easily be adapted to other leisure activity domains offered by this web like restaurants, theatres, concerts or museums for instance.[g]

[e] Previous works have reported that trust and tie strength are conceptually different but that there is a correlation between them.[21]
[f] http://www.guiadelocio.com/
[g] We are aware that this is limited to Spain's leisure offers but believe that it could easily be adapted to other leisure webs in other countries. Therefore we have included it as a possible *method* that implements this *task*.

3.1.7. *Scoring*

(Corresponds to *Individual Estimation* module in ARISE) Consists of obtaining predicted ratings $\text{pred}(u, i)$ for each active user $u \in G$ and target item $i \in T$. Some of the different *methods* that can be used to implement this *task* are:

- *Collaborative recommenders*,[11,20,15] that use already assigned ratings by users to several products. Users are selected according to their similarity with the individual receiving the recommendation (by comparing their given ratings to products). Most similar users are used as predictors and their ratings are combined to estimate the rating that the target user would assign to a new product.
- *Content-based recommenders*,[22] that compare each item in the catalogue with items already rated by the target user. Then ratings of the most similar rated items are combined to provide a recommendation.
- *Hybrid recommenders*,[8] that are a combination of the two previous ones.
- *Asking other G users to give an estimated rating for the product i*,[10] this method relies heavily on explicit feature-level feedback from users.
- *Influence based recommenders*,[34,32] that modify non-social predictions $\text{pred}(u, i)$ obtained with one of the above methods with the *personality* and *trust* factors computed in previous *tasks* (3.1.3). We detail this method as it is the one used in *HappyShopping*'s recommendations as we will see in Section 4. It supposes that the user may modify her/his preference for an item depending on the preferences given by her/his friends to the same item. For example, if our rating for an item is 3 and our friend has a rating of 5 for the same item, we could think on modifying our rating to 4. Depending on the trust in this friend, we decide the level of variation for our rating (i.e. 3.5 if the trust is low, and 4.5 if trust is high). Furthermore, the variation of our rating also depends on our personality. If we have a strong personality (high personality value) we will not be willing to change our rating, but if we have a weak personality (low value) we could be easily influenced by other users.

  The method combines the *personality* and *trust* factors using the following equation:

$$ibr(u, i) = \text{pred}(u, i) + (1 - p_u) \cdot \frac{\sum_{v \neq u \in G} t_{u,v} \cdot (\text{pred}(v, i) - \text{pred}(u, i))}{|G| - 1}. \quad (1)$$

  In this equation, $\text{pred}(u, i)$ is modified according to its difference with other users's ratings $(\text{pred}(v, i) - \text{pred}(u, i))$. This difference is weighted with the trust between users $(t_{u,v})$, where $v \in G$, being G the group formed by the people who would have the most influence in user u, and therefore, the people who s/he trusts the most. Finally, the accumulated difference is weighted according to u's personality in an inverse way $(1 - p_u)$.

3.1.8. *SelectCases*

Consists of selecting for each active user $u \in G$ the k items from T whose predicted ratings are highest. For example, in *HappyShopping*, we use $k = 4$. Note that the next 3 *tasks* are specific for social group recommendations, and therefore the *method* that implements this *task* will need to have a *display cases* option for single individual recommender implementations.

3.1.9. *CombineIndividualRecommendations*

(Corresponds to ARISE module in ARISE) Consists of obtaining a group prediction, gpred, aggregating group members predicted ratings, $\mathrm{pred}(u, i)$ for each $u \in G$ and $i \in T$ (see Eq. (2)). Possible aggregation functions ($\bigsqcup$ in the equation) include *least misery* (where the minimum is taken), *most pleasure* (where the maximum is taken) or *average satisfaction* (where the average of the predicted ratings of each group member is taken). With the data retrieved in our experiments in simulated environments[32,34,36] we have performed a conscientious experimentation comparing the recommendation results of the state-of-the-art aggregation functions presented by Ref. 25 when applying them to what we will next define as *standard* (Eq. (2)) and *social* (Eq. (3)) recommendations approaches. The results of these experiments are out of the scope of this paper but can be found in Ref. 36. During this experimentation we found that *average satisfaction* reported better results for small groups (we consider groups of 10 or less as small) than the other studied aggregation functions, and therefore it is the strategy adopted in *HappyMovie* (as for the moment we do not expect to have large groups using the application). Now we will detail the differences between *standard* and *social* group recommenders. The following equation, represents a baseline group recommender:

$$\mathrm{gpred}(G, i) = \bigsqcup_{\forall\, u \in G} \mathrm{pred}(u, i)\,. \tag{2}$$

We designate this baseline recommender by *Standard Group Recommender* which represents state-of-the-art recommenders that do not use social factors. Differently, in our *Social Group Recommendation* method,[34,32,36] we modify the individual ratings with the personality and trust factors. This way, we modify the impact of individual preferences as shown in Eq. (3).

$$\mathrm{gpred}(G, i) = \bigsqcup_{\forall\, u \neq v \in G} f(\mathrm{pred}(v, i), p_u, t_{u,v}) \tag{3}$$

where $\mathrm{gpred}(G, i)$ is the group rating prediction for a given item i; $\mathrm{pred}(v, i)$ is the original individual prediction for user v and item i; p_u is the personality value for user u and $t_{u,v}$ is the trust value between users u and v.

There are several methods to modify a user's predicted rating predicted according to personality and trust factors, it is represented as the $f(\ )$ function in

Eq. (3). Some of these ways are the *influence-based method* (see Eq. (1)[h]) or the *delegation-based method* which we next detail:

The *delegation-based method* recognizes that a person's opinions may be based in part on the opinions of other group members. Basically, in each user's turn the user's opinion is not taken into account but it is considered in the other $(n-1)$ turns that is when the user influences others. We know that this is not an intuitive idea. Basically, instead of taking users' opinion once into account, the method takes it several times into account, once for each other user in the group. In our previous work,[34,36] when testing our method, we showed that our delegation-based method improves the accuracy of predicted group ratings more than any other *standard* or *social* approach that we have studied. The formula, which we explain below, is as follows:

$$
\begin{aligned}
dbr(\mathrm{pred}(u,i),G) &= \frac{1}{T}\sum_{v\neq u\in G} t_{u,v}[\mathrm{pred}(v,i)+\theta_{\mathrm{pred}(v,i)}\cdot\Delta p_{u,v}]+m_v \\
\text{where:}\quad T &= \sum_{v\neq u\in G} t_{u,v} \\
\Delta p_{u,v} &= p_v - p_u \\
m_v &= \alpha(1-s_v)p_v\,.
\end{aligned}
\tag{4}
$$

In Eq. (4), $t_{u,v}$ denotes the trust between u and v, which is a real number between 0.0 (no connection) and 1.0 (strong connection). For a given user u in group G_a, we take into account the predicted ratings, $\mathrm{pred}(v,i)$, for the rest of the group members, $v \in G, v \neq u$, weighted by the trust between the two users, $t_{u,v}$. This follows,[13] where a method for group recommendations using trust is proposed.

Variable p_u denotes user u's personality, also, as we remember, a real number between 0.0 (very cooperative) and 1.0 (very selfish). The predicted rating of the other group members $\mathrm{pred}(v,i)$ is increased or decreased depending on the difference in personalities, $\Delta p_{u,v}$. This way, users with stronger personalities will contribute more to the final score. A user v with a positive opinion of i, i.e. where $\mathrm{pred}(v,i)$ is greater than the mid-point of the ratings scale, will want to increase u's opinion of i; but if v has a negative opinion, i.e. where $\mathrm{pred}(v,i)$ is less than the mid-point of the scale, then v will want to decrease u's opinion. We model this through a function θ:

$$
\theta_{\mathrm{pred}(v,i)} = \begin{cases} +5 & \text{if } \hat{r}_{v,i} \geq \mathrm{mid} \\ -5 & \text{otherwise} \end{cases}
\tag{5}
$$

where mid is the mid-point of the ratings scale, e.g. 3 on a five-point Likert scale. We have chosen constants 5 and -5 because after several studies in group personality

[h]Note that if we want a system that takes into account user satisfaction, a memory factor m_u, presented in next equation, is also used in *influence-based* original Eq. (1). However, for the *influence-based method*, differently to the *delegation-based method* (where v's satisfaction is taken into account) we use $m_u = (1-s_u)p_u$.

composition[33,34] we have observed that the mean difference in group personality composition is 0.2 and therefore the impact of $\theta_{r_{v,i}} \cdot \Delta p_{u,v}$ in Eq. (4) will typically be 1 or -1, which in comparison with other tested ranges has proven to be the most adequate.

Finally, we also include m_v, that represents the memory of past recommendations. The satisfaction value s_v is the level of satisfaction of user v.[i] A user who is extremely happy with the recommendations will have this satisfaction value close to 1. However, the more dissatisfied with the recommendations s/he is, the more that this value will decrease, reaching down to 0 in the worst case. Note that initially all users are assigned a $s_v = 1$. Therefore, the first time that a group receives a recommendation the memory factor is nullified in the formula as it is not necessary because there are not previous recommendations. Parameter α is used to modify the impact of memory in *delegation-based method*. It has a positive or negative value according to $\mathrm{pred}(v, i)$ in the same way that $\theta_{\mathrm{pred}(v,i)}$ has. It is important to note that this satisfaction value is also weighted depending on user v's personality to reflect the importance of satisfying that concrete user. Once the recommendation process has finished the s_v value is updated for every user (this is done in the *UpdateGroupHistory task*).

3.1.10. *Filtering*

Consists of selecting the k' items in T that have the highest predicted ratings for the group. For example, in *HappyMovie*, we used $k' = 3$.

3.1.11. *DisplayCases*

Consists of displaying to each user u receiving the recommendation the k' items obtained by the group recommender.

3.1.12. *UpdateGroupHistory*

Consists of revising the case c that corresponds to the active user u (for individual recommenders) or the active group of users G (for group recommenders) with the new recommendation and retaining it in the case base CB for future recommendations. Note that this *task* is optional and can be skipped if developers do not want to build a system with memory of past recommendations.

4. *HappyShopping*

In this section we present *HappyShopping*[j]: A Facebook social individual recommender application for clothes that follows our generic architecture ARISE and has

[i] The followed procedure for the computation of satisfaction value s_v can be found in Ref. 35.
[j] http://www.happyshopping.es/

been built using our *social templates*. With the development of this application we study and prove the two goals of this paper:

- The usability of our templates (detailed in Section 3.1).
- The viability of our generic architecture ARISE (detailed in Section 2) in other domains.

In Sections 2 and 3 we have detailed how to design a social recommender using ARISE and how to implement it using our social templates, hence, we will only detail now the concrete choices of domain and development that delimit *HappyShopping*. To understand how each module from the ARISE architecture is defined in one *task* of the templates and how *HappyShopping* implements the *methods* of the needed *tasks* of the templates, we introduce Figure 3. In the top of the Figure we see ARISE's modules, each line that goes out of a module points the concrete *task* in the *social templates* that corresponds it and each line that comes out of a *task* in the templates points the concrete module in *HappyShopping*'s structure that implements it.

4.1. *Details of* HappyShopping

Traditional recommender systems do not take into consideration explicit social relations among users, yet the importance of social influence in product marketing has long been recognized.[42] Intuitively, when we want to buy a product that is not familiar, we often consult with our friends who have already had experience with the product, since they are those that we can reach for immediate advice. When friends recommend a product to us, we also tend to accept the recommendation because their inputs are trustworthy. *HappyShopping* exploits this fact and takes into account preferences of users' closest friends in order to recommend which piece of clothing users should purchase and later propose an argumentation process with these closest friends about the recommended items. *HappyShopping*'s main goal is to present a recommender system that proposes pieces of clothing by taking into account users social context. The recommendation process is summarized in steps below:

- **Product Comparison with User Preferences.** The application requires the user to explicitly identify products that are of her/his interest, which will form the users' "wardrobe".
- **Product Comparison with the Preferences of Most Influential Friends.** In this step we model the impact of the preferences of people influencing the user that is being recommended. The proximity between users (users trust) is obtained by analyzing the information available on the social network: messages exchanged, shared photos, etc.
- **Weighting of Items Regarding the Degree of Influence of Individuals.** Influence of other group members not only depends on their proximity or trust in them, but also in the degree of personality or leadership of these influencers. In this step the products to be proposed are reconsidered depending on these

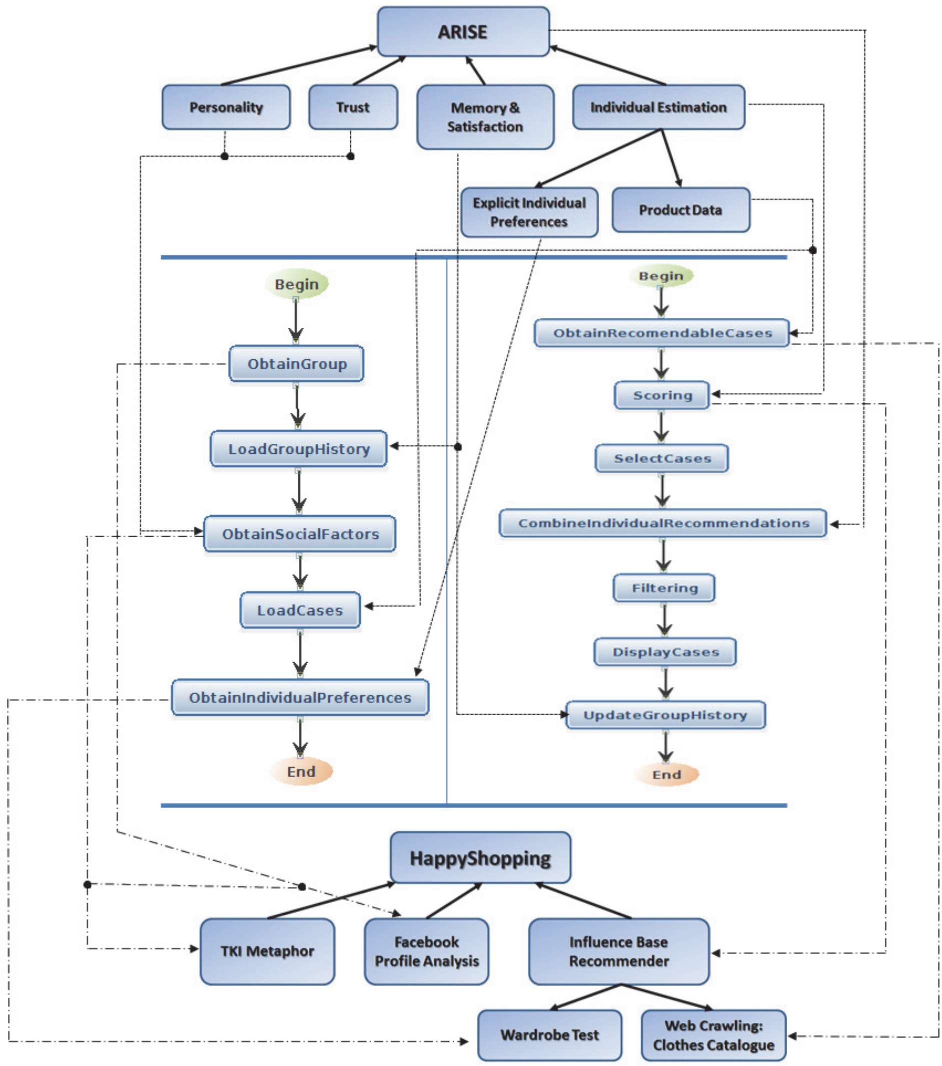


Fig. 3. Relationship between the ARISE architecture, the proposed templates and its instantiation in *HappyShopping*.

factors. This process requires obtaining the personality information from the social network.

4.2. *Using* HappyShopping

Using the* HappyShopping *System. Users start their Facebook account and look for *HappyShopping* in the applications section. *HappyShopping*'s main page is shown in Figure 4. The required steps to obtain a clothing recommendation with *HappyShopping* are explained below:

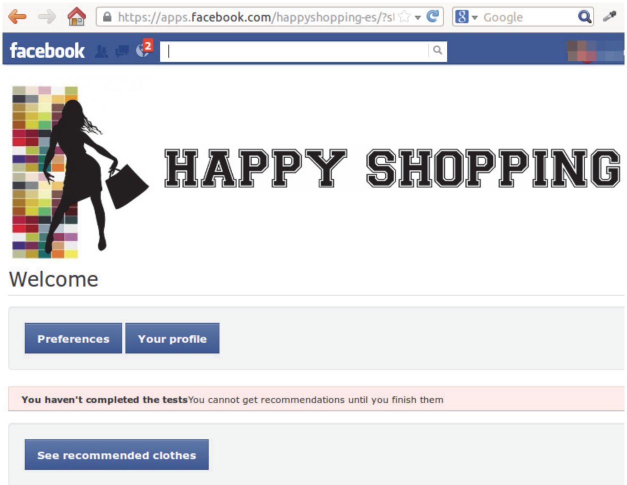


Fig. 4. *HappyShopping*'s main page.

- **Creating a User Profile in the Application.** Before any user can access the clothing recommendation results users have to create their individual "recommendation profile" which is necessary for our recommendation method. This profile is based on three different aspects: personality, individual preferences and trust in other users.

  — To obtain the personality users have to choose a series of characters to whom they feel identified, Figure 5 shows *HappyShopping*'s personality test implementation. This step corresponds to the *Personality* module in ARISE and its solved by *ObtainUsersPersonality task* in our template. The concrete *method* that implements this *task* is the *TKI's alternative movie metaphor* explained in Section 3.1.

  — To obtain the preferences profile users have to rate a set of clothes (at least 20 pieces), where they enter their personal preferences, Figure 6 shows *HappyShopping*'s preferences test implementation. This step corresponds to the *Explicit Individual Preferences* module in ARISE and its solved by *ObtainIndividualPreferences task* in our template. The specific pieces that are displayed for the user to rate (users can rate 100 pieces at the most) are selected automatically from *HappyShopping*'s catalogue trying to maximize diversity. To do so, a similar metric as the one presented in the system ExpertClerk[41] is used.

  — To obtain the trust, the application reads the information stored in Facebook personal profiles. It calculates the trust that the user has with all the other users in her/his close circle (G). To obtain the circle of trusted people in the social environment of the user receiving the recommendation, the application needs to calculate which other application users should form the group G.

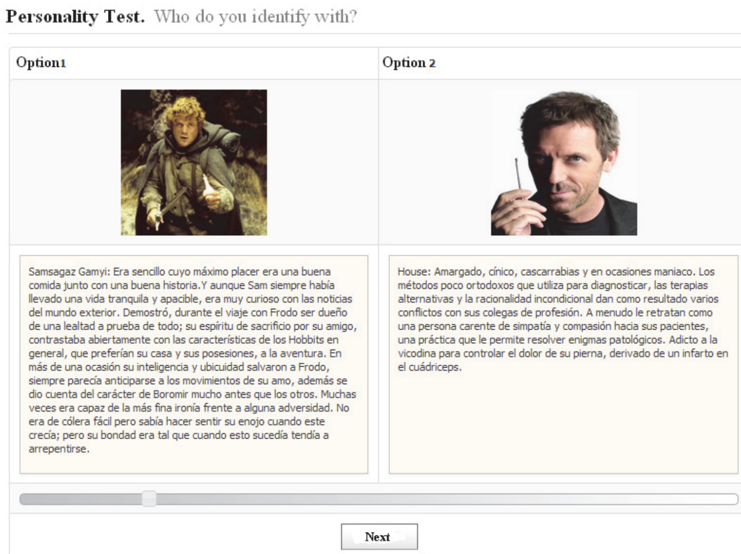


Fig. 5. Personality test in *HappyShopping*.

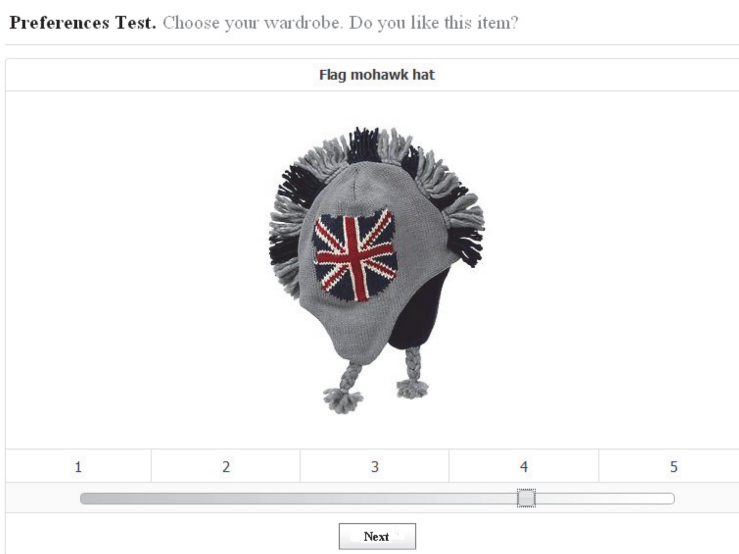


Fig. 6. Preferences test in *HappyShopping*.

This step is solved by the *task ObtainGroup* in our template. The concrete *method* that implements this *task* is *Calculating the group of closest friends in the social network* explained in Section 3.1. Note that trust value is general to friends and not domain specific (clothes) as it has been proven most efficient in Refs. 34 and 32.

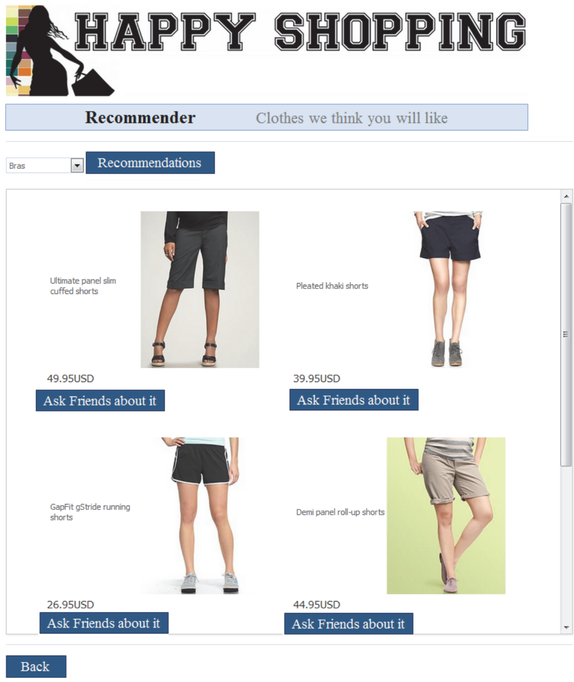


Fig. 7. *HappyShopping*'s recommendation page.

- **Recommendation.** Once the application has obtained the factors that identify each user receiving a recommendation (personality, individual preferences and trust in other users) user are able to click the "See recommended clothes" bottom (see Figure 4) and see their individual recommendations. This step corresponds to the *Individual Estimation* module in ARISE and its solved by the *Scoring task* in our template. The concrete *method* that implements this *task* is *Influence based recommenders* explained in Section 3.1.
- **Once the Recommendation is Made.** *HappyShopping* provides a list with the best 4 pieces of clothing that the recommender has found in the catalogue (See Figure 7). For each of them the user will have two options:
  - — Purchase the product. Note that this function is not part of the application.
  - — Start an argumentation process with group G members. Where the user will ask her/his closest friends which piece of clothing fits her/him best (see Figure 8). Note that although we have designed this option it is not part of the *social templates*.

HappyShopping counts with a catalogue of 1887 pieces. This catalogue has been obtained by parsing the web searching for different types of clothes and styles to wear. Each item in the catalogue is formed by a picture of the piece of clothing plus the concrete characteristics of the piece like material, colour, style, size, prize, etc.

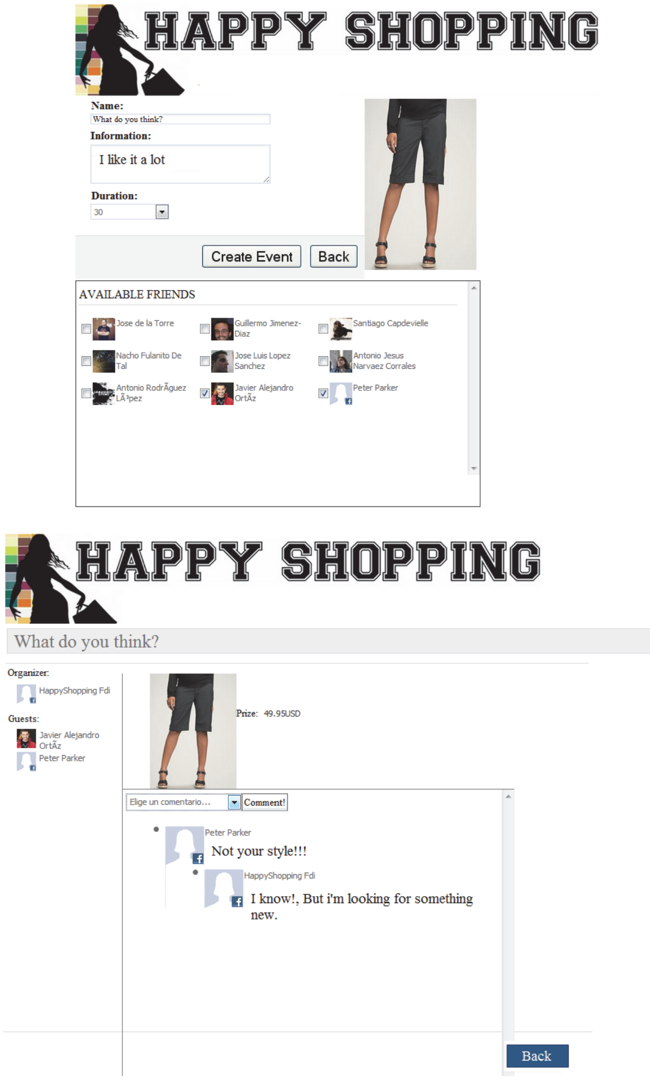


Fig. 8. Check clothes with *HappyShopping*'s friends page.

4.3. *Usage evaluation of our social templates and ARISE*

Regarding the effort and viability of using ARISE and our *social templates* for the development of *HappyShopping*, we have counted with three developers. The skills and background of *HappyShopping*'s developers are summarized in Figure 9, that reflects the average of the answers given by the three developers to a questionnaire about how they graded themselves. These developers have reused our generic architecture ARISE and its associated templates when building this new social recommender application from scratch.

When you started the development of the application, How do you define your knowledge in? :	Developers answer
Developer skills	4
Programming recommender systems	2
Programming CBR systems	1
Facebook programming	1
Programming with COLIBRI Studio	2

Fig. 9. Questionnaire answers of *HappyShopping*'s developers about their skills and background. Answers are in a scale 0 to 5. Being 0 very little and 5 a lot.

When we questioned our three developers about having ARISE to guide them in the construction of *HappyShopping* they pointed out that it was a very useful architecture that had really come in handy and that had helped them and quickened their work in tasks like *HappyShopping*'s general design, the project's work planning and the later understanding of the *social templates*. For all these reasons they defined ARISE as a key element in the construction of *social recommender* applications.

Regarding the usability of the set of templates developers answered that templates had facilitated and quickened their work thus they all preferred to have the templates to assist them. Besides, we handed them a paper questionnaire where they had to elaborate on which parts of the templates could be completely reused (as we remember semantic templates' definition, presented in Section 3, templates are used for both, retrieval and adaptation of systems), which ones needed to be adapted, where was the bottleneck of creating a new social recommender application, if knowledge in Facebook programming was required and if extensive knowledge in recommender systems was needed:

Regarding which parts could completely be reused, developers pointed that the whole social factor elicitation process could completely be reused, the personality test was automatically generated by the templates and therefore in their project they just had to include it (i.e. include personality.php and read the $\$personality$ variable) as well as the trust elicitation part. More importantly, they remarked that the bottleneck of generating a new social recommender application on Facebook was the required knowledge in Facebook programming. For example, they pointed out that: "One specific difficult part, was the knowledge required to extract social information from users' Facebook profiles. This was necessary in order to compute users' trust. With the *social templates* the process of computing trust between users is done by automatically attaching the *trust elicitation project* to our application. Therefore, we only had to read the automatically generated comments that implemented *tasks methods* present and learn how to use the offered web service (*http://recoserver.fdi.ucm.es/TrustGen/user*). On the other hand, we estimate that it would have taken us more than 2 months to learn the necessary Facebook programming and replicate for *HappyShopping* the *trust elicitation project*, as it has more than 42408 code lines".

Developers also commented that thanks to the *social templates*, extensive knowledge in programming CBR and recommender systems was not needed. They pointed out that as the *scoring task* provides the *influence based* implemented method (more concretely through the templates developers were able to select as *implemented method HappyMovie*'s individual recommender) the task of implementing *HappyShopping*'s individual recommender was almost immediate as they only had to modify the domain specific attributes (i.e. genre, cast, directors, synopsis, etc.) being compared in *HappyMovie*'s individual recommender (which is a *social* recommender that uses as *standard* recommender a content-based recommender) in favour for the domain specific attributes that each of *HappyShopping*'s items present (i.e. material, colour, style, size, prize, etc.).

Methods for *tasks* like *ObtainRecommendableCases* or *LoadCases* had to be fully implemented, as the provided *implemented methods* were not useful for the proposed domain. Therefore, developers had to implement a web crawler specific for the clothing domain.

Finally, developers commented that even though most of the required programming was automatically generated through the templates, the body of the web application itself had to be newly implemented (all the php code for the different *HappyShopping* pages) and they suggested that as some parts are common in most Facebook recommender applications (like invite Facebook friends, create events, etc.) and require knowledge in Facebook programming, we could offer a complete framework where new developers just have to select which general characteristics they want in their application and these would also be automatically generated as done with the *social templates*. Note that we considered that this was a valuable feedback and are now working in extending the proposed templates so that they can help users not only with the recommenders design but also with the web part of Facebook recommender applications.

Summing up, regarding the effort in building *HappyShopping*, when we asked developers how long it took them to build an initial version of the application, they answered that it took them five weeks to develop an initial version and ten weeks to develop the final version of *HappyShopping*. If we compare these results with the time that took us (the authors) to develop *HappyMovie* (which as we remember is our other social recommender application, and it belongs to the movies domain) we can conclude that the usage of our social templates and ARISE has been a success. It took us more than five months to develop *HappyMovie*, and we were three expert programmers specialized in CBR and recommender systems. Obviously, as it was the first time that the social recommender system was being implemented there was a high cost in the design and development of *HappyMovie*, which has been captured in the social templates and the generic architecture and makes the cost of a second *social recommender application* descend. Therefore, we consider that the use of our social templates and ARISE indeed facilitates and eases the construction of other social recommender applications.

5. Conclusions and Future Work

In this paper we have presented our generic architecture, ARISE, and a set of templates that formalize the behaviour of social recommender systems. We have proven ARISE's suitability by building two different recommender applications in two different domains. Besides we have proven that is a viable architecture for building any kind of social recommender system, as it has been proven to be adequate for both *social group recommender* systems (its initial purpose, which was tested with *HappyMovie*) and for *social individual recommender* systems, which was a new challenge that has been now tested with *HappyShopping*.

As a result of ARISE's design, we now count with two different case studies, *HappyMovie*, which is a particular instantiation of ARISE for recommending movies to groups of people connected through the social network Facebook. And, *HappyShopping*, which is a clothing individual recommender system that follows our method of making recommendations to people using their social information stored in the social network Facebook.

Most importantly, we have presented a set of templates that represent an intermediate step in the development of *social group recommender applications.* After an experiment where three developers have used ARISE's architecture and our set of *social templates* to develop from scratch a new *social recommender* system, *HappyShopping*, we have proven that the usage of *social templates* indeed quickens and facilitates the process of building new applications. The estimated time-improvement is five times quicker than without using the *social templates.* Thus, developers have concluded that they prefer to have the templates to assist them. This statements let us conclude that the usage of our *social templates* and ARISE has been a success.

There is much that can be done to take this work forward. For us, the next step is taking *HappyShopping* one step forward and make it a richer application with actual ratings from users, from which we hope to gather data and use it as basis for future experiments. We also want to develop and interactive framework for the development of new *social group recommender* applications, were developers can select application properties from a list and the framework automatically generates the new application.

Acknowledgment

Supported by Spanish Ministry of Science & Education (TIN2009-13692-C03-03) and Madrid Education Council and UCM (Group 910494) and Spanish Ministry of Economy and Competitiveness under grant IPT-2011-1890-430000.

References

1. *Merriam-Webster's Collegiate Dictionary*, 10th edn. (2002).
2. A. Aamodt and E. Plaza, Case-based reasoning: Foundational issues, methodological variants, and system approaches,*Artificial Intelligence Communications* **7**(1) (1994) 39–59.
3. G. Adomavicius and A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Trans. Knowl. Data Eng.* **17**(6) (2005) 734–749.
4. L. Ardissono, C. Gena, P. Torasso, F. Bellifemine, A. Difino and B. Negro, User modeling and recommendation techniques for personalized electronic program guides, in *Personalized Digital Television — Targeting Programs to Individual Viewers*, Vol. 6 of Human-Computer Interaction Series (Kluwer Academic Publishers, 2004), Chapter 1, pp. 3–26.
5. J. Bao, Y. Zheng and M. F. Mokbel, Location-based and preference-aware recommendation using sparse geo-social networking data, in *SIGSPATIAL/GIS* (2012) 199–208.
6. M. Bazire and P. Brézillon, Understanding context before using it, in *Context* (2005), pp. 29–40.
7. D. Billsus and M. J. Pazzani, A personal news agent that talks, learns and explains, in *Agents* (1999), pp. 268–275.
8. R. D. Burke, Hybrid recommender systems: Survey and experiments, *User Model. User-Adapt. Interact.* **12**(4) (2002) 331–370.
9. N. A. Christakis and J. H. Fowler, Social contagion theory: Examining dynamic social networks and human behavior, *CoRR* (2011).
10. E. Costello, J. Doody, L. McGinty and B. Smyth. *i*care: Intelligent customer assistance for recommending eyewear, in *IUI* (2006), pp. 282–284.
11. M. D. Ekstrand, J. Riedl and J. A. Konstan, Collaborative filtering recommender systems, *Foundations and Trends in Human-Computer Interaction* **4**(2) (2011).
12. E. Gilbert and K. Karahalios, Predicting tie strength with social media, in *CHI '09: Proc. of the 27th Int. Conf. on Human factors in Computing Systems* (ACM, 2009), pp. 211–220.
13. J. Golbeck, Generating predictive movie recommendations from trust in social networks, in *iTrust: 4th Int. Conf. on Trust Management* (2006).
14. J. Golbeck, Combining provenance with trust in social networks for semantic web content filtering, in L. Moreau and I. T. Foster (eds.) *Provenance and Annotation of Data, Int. Provenance and Annotation Workshop, IPAW 2006*, Chicago, IL, USA, May 3–5 (2006), Revised Selected Papers, Vol. 4145 of Lecture Notes in Computer Science (Springer, 2006), pp. 101–108.
15. J. L. Herlocker, J. A. Konstan and J. Riedl, An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms, *Inf. Retr.* **5**(4) (2002).
16. X. Hu, C. Chen, X. Chen and Z.-K. Zhang, Social recommender systems based on coupling network structure analysis, *CoRR* (2012) abs/1204.1949.
17. M. Jamali and M. Ester, Using a trust network to improve top-n recommendation, in *RecSys* (2009), pp. 181–188.
18. M. Jamali and M. Ester, *TrustWalker*: A random walk model for combining trust-based and item-based recommendation, in *KDD* (2009), pp. 397–406.
19. A. Jameson and B. Smyth, Recommendation to groups, in *The Adaptive Web, Methods and Strategies of Web Personalization* (Springer, 2007), pp. 596–627.
20. Y. Koren and R. M. Bell, Advances in collaborative filtering, in *Recommender Systems Handbook* (2011), pp. 145–186.

21. D. Z. Levin, R. Cross and L. C. Abrams, The strength of weak ties you can trust: The mediating role of trust in effective knowledge transfer, *Management Science* **50** (2004) 1477–1490.
22. P. Lops, M. de Gemmis and G. Semeraro, Content-based recommender systems: State of the art and trends, in *Recommender Systems Handbook* (2011).
23. N. Ma, E.-P. Lim, V.-A Nguyen, A. Sun and H. Liu, Trust relationship prediction using online product review data, in *CIKM-CNIKM* (2009), pp. 47–54.
24. P. Massa and P. Avesani, Trust-aware recommender systems, in *RecSys* (2007), pp. 17–24.
25. J. Masthoff, Group modeling: Selecting a sequence of television items to suit a group of viewers, *User Modeling and User-Adapted Interaction* **14**(1) (2004) 37–85.
26. J. Masthoff and A. Gatt, In pursuit of satisfaction and the prevention of embarrassment: Affective state in group recommender systems, *User Modeling and User-Adapted Interaction* **16**(3-4) (2006) 281–319.
27. K. McCarthy, J. Reilly, L. McGinty and B. Smyth, Thinking Positively – Explanatory Feedback for Conversational Recommender Systems, Technical report, in *Proc. of the ECCBR 2004 Workshops*, 2004.
28. B. Mobasher, D. Jannach, W. Geyer and A. Hotho, 4th ACM RecSys Workshop on Recommender Systems and the Social Web, in *RecSys* (2012), pp. 345–346.
29. S. Nepal, C. Paris and S. K. Bista, SREC: A social behaviour based recommender for online communities, in *UMAP Workshops* (2012).
30. D. O'Sullivan, D. C. Wilson and B. Smyth, Improving case-based recommendation: A collaborative filtering approach, in *ECCBR* (Springer, 2002), pp. 278–291.
31. M. S. Pera, N. Condie and Y.-K. Ng, Personalized book recommendations created by using social media data, in *WISE Workshops* (2010), pp. 390–403.
32. L. Quijano-Sánchez, J. A. Recio-García and B. Díaz-Agudo, Personality and social trust in group recommendations, in *ICTAI'10* (IEEE Computing Society, 2010), pp. 121–126.
33. L. Quijano-Sánchez, J. A. Recio-García and B. Díaz-Agudo, Happymovie: A facebook application for recommending movies to groups, in *ICTAI'11* (2011), pp. 239–244.
34. L. Quijano-Sánchez, J. A. Recio-García, B. Díaz-Agudo and G. Jiménez-Díaz, Social factors in group recommender systems, in *ACM Transactions on Intelligent Systems and Technology* (2013).
35. L. Quijano-Sánchez, J. Antonio Recio-García and B. Díaz-Agudo, User satisfaction in long term group recommendations, in *Procs. of the ICCBR-2011 19th Int. Conf. on Case Based Reasoning* (2011).
36. L. Quijano-Sánchez, J. A. Recio-García and B. Díaz-Agudo, An architecture for developing group recommender systems enhanced by social elements, *Applied Intelligence* (2014).
37. J. A. Recio-García, B. Díaz-Agudo and P. A. González-Calero, Semantic templates for case-based reasoning systems, *Knowledge Eng. Review* **24**(3) (2009) 245–264.
38. J. A. Recio-García, P. A. González-Calero and B. Díaz-Agudo, Template-based design in colibri studio, *Information Systems* **40** (2014) 168–178.
39. F. Ricci, B. Arslan, N. Mirzadeh and A. Venturini, ITR: A case-based travel advisory system, in S. Craw and A. Preece (eds.) *ECCBR* (Springer, 2002), pp. 613–641.
40. U. Shardanand and P. Maes, Social information filtering: Algorithms for automating "word of mouth", in *CHI* (1995), pp. 210–217.
41. H. Shimazu, Expertclerk: A conversational case-based reasoning tool for developing salesclerk agents in e-commerce webshops, *Artif. Intell. Rev.* **18** (2002).
42. M. R. Subramani and B. Rajagopalan, Knowledge-sharing and influence in online social networks via viral marketing, *Commun. ACM* **46**(12) (2003) 300–307.

43. K. W. Thomas and R. H. Kilmann, *Thomas-Kilmann Conflict Mode Instrument* (Tuxedo, N.Y., 1974).
44. X. Yang, H. Steck, Y. Guo and Y. Liu, On top-k recommendation using social networks, in *RecSys* (2012), pp. 67–74.
45. F. Zhang, L. Bai and F. Gao, A user trust-based collaborative filtering recommendation algorithm, in *ICICS* (2009), pp. 411–424.
46. J. Zimmerman, K. Kurapati, A. L. Buczak, D. Schaffer, S. Gutta and J. Martino, in *Personalized Digital Television — Targeting Programs to Individual Viewers*, Vol. 6 of Human-Computer Interaction Series (Kluwer Academic Publishers, 2004), Chapter 2, pp. 27–51.

Chapter 22

An Architecture for Developing Group Recommender Systems Enhanced by Social Elements

22.1 Citation

Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B., An Architecture for Developing Group Recommender Systems Enhanced by Social Elements, Applied Intelligence, volume 40, issue 4, pages: 732-748, Springer, 2014. ISSN: 0924-669X. JCR (2012): 1.853. (Computer Science, Artificial Intelligence, 32 out of 111, Q2).

22.2 Contributions covered by this paper

In this paper we have have abstracted the recommendation process and compiled our techniques and algorithms in an organized generic architecture named ARISE. With it we have proven that our social group recommendation approach is domain independent and we have provided to the recommenders community a tool to easily reproduce these type of social systems. We also present and experimental analysis with both synthetic and real data where we measure the improvement in the accuracy of the recommenders that use a social approach. Besides we have implemented several group recommender configurations varying the number of social factors included (none, just the personality factor, just the trust factors and both social factors) and tested their performance in our social group recommender application *HappyMovie*. We have also determined which aggregation functions perform better with which group configurations.

Appl Intell (2014) 40:732–748
DOI 10.1007/s10489-013-0504-y

An architecture and functional description to integrate social behaviour knowledge into group recommender systems

Lara Quijano-Sanchez · Juan A. Recio-Garcia · Belen Diaz-Agudo

Published online: 11 January 2014

Abstract In this paper we consider the research challenges of generating a set of recommendations that will satisfy a group of users with potentially competing interests. We review different ways of combining the preferences of different users and propose an approach that takes into account the social behaviour within a group. Our method, named *delegation-based prediction method*, includes an analysis of the group characteristics, such as size, structure, *personality* of its members in conflict situations, and *trust* between group members. A key element in this paper is the use of social information available in the Web to make enhanced recommendations to groups. We propose a generic architecture named ARISE (Architecture for Recommendations Including Social Elements) and describe, as a case study, our Facebook application *HappyMovie*: a group recommender system that is designed to provide assistance to a group of friends that might be selecting which movie to watch on a cinema outing. We evaluate the performance (compared with the real group decision) of different recommenders that use increasing levels of social behaviour knowledge.

Keywords Group recommender systems · Social networks · Personality · Trust · Generic architecture

L. Quijano-Sanchez (✉) · J.A. Recio-Garcia · B. Diaz-Agudo
Facultad de Informatica, Universidad Complutense de Madrid, C/Prof. Jose Garcia Santesmases s/n, 28040 Madrid, Spain
e-mail: lara.quijano@fdi.ucm.es

J.A. Recio-Garcia
e-mail: jareciog@fdi.ucm.es

B. Diaz-Agudo
e-mail: belend@fdi.ucm.es

1 Introduction

It is becoming common to employ recommendation technologies to aid users in the task of finding interesting items in the Web [37]. There is a wide range of products such as books, music, games, trips, etc. that are difficult to discover in the Web due to the overwhelming amount of information available. Recommender systems [24] enable users to find items and provide a richer and more interactive user experience than classical interfaces based on catalogues of products.

Initially, existing recommenders were focused on individual users [16, 31]. Nowadays, however the rise of the collaborative Web (a.k.a. Web 2.0) has encouraged the development of activity-planning through social networks, like watching a movie, going to a restaurant, listening to a radio station or traveling with friends. A clear example are events organized through social networks like Facebook. Here, recommender systems can play a significant role, since agreement on a common item by several users is not a simple task. To address this issue, the number of recommender systems that deal with the challenge of making recommendations for groups of people has increased [34, 38]. Group recommendation, however, is not a mere aggregation of individual preferences. Humans are social individuals and, therefore, social behaviour has a great impact on their group decision-making processes. Our proposal takes into account this fact and assumes that the general satisfaction of the group does not always mean aggregating its members' preferences. It is clear that groups have an influence on individuals when coming to a decision. This is commonly referred to as *emotional contagion*: the effect of individuals' affective state on others in the group [6, 21, 33]. This contagion is usually proportional to the *tie strength* or *trust* between individuals as closer friends have a higher influence [19, 39, 50]. However, the influence of the group also depends on the individual's degree of *conformity* [33]. It has been proved that humans adjust their opinions to conform with those of a group when the majority of the group expresses a different opinion. The degree of conformity is counteracted by the individual's behaviour when facing a conflict situation. Here, *personality* influences the acceptance of others' proposals [44].

Previous research on group recommendation considers the preferences of every member in the group with the same degree of importance and try to satisfy the preferences of every individual. However, all these social elements (emotional contagion, trust, personality, ...) should be included in the recommendation model to fully represent the group behaviour when choosing a shared item. Although it seems natural to model this social knowledge, a major limitation appears: social factors are very difficult to estimate. Up to now, it was impossible to obtain these factors without annoying users with several questionnaires. But nowadays the collaborative Web provides a tool that can be used to lighten this problem: social networks. Social networks let users interact and develop their social relationships in a computer-based environment. Indeed, several works have pointed out that social elements can be inferred from them [9, 20]. For example, we can estimate a tie between users by measuring the number of messages exchanged or the number of friends in common.

The first contribution of this paper is the compilation of our ideas in an organized generic architecture named ARISE (Architecture for Recommendations Including Social Elements) that can be instantiated into group recommender systems that take into account social behaviour knowledge. In the functional description of our architecture we will detail how social knowledge provided by the modules inside ARISE is combined to obtain a recommendation that integrates the individual preferences and social features of the group.

To do so, individual preferences are modified according to the social environment of the user. This idea is reflected in our novel technique to estimate an individual's preference for a given item based on social factors. A preliminary version of this technique was introduced in [42]. The research presented in this paper shows a more mature work, where we have refined, tested and justified the ideas and decisions made in [42]. We have named this new approach the *delegation-based prediction* (*dbp*) method. As the name suggests, the idea behind this method is that users create their preferences based on others' opinions. We consider this new perspective of our past work as the second contribution of the paper.

In the next step, these individual predictions are combined to generate an aggregated preference for the group. Masthoff [32] presents a compilation of the most important preference aggregation techniques pointing out that the selection of a proper aggregation strategy is a key element in the success of the generated recommendation for the group. Therefore, our third contribution is the adaptation of these techniques to our *delegation-based* method, plus a comparative analysis of their performance. This study indicates which is the best aggregation strategy depending on the characteristics and nature of the group.

To perform this evaluation we have instantiated our generic architecture into a real application called *HappyMovie*, that conforms the last contribution of this paper. It is a Facebook system for the movie recommendation domain. Although we have chosen this domain as a case study, we discuss how the architecture and group recommendation approaches presented in this paper could be applied to any other domain.

The discussion about this architecture and its instantiation is presented first in Sects. 2 (ARISE) and 4 (*HappyMovie*). Next, Sect. 3 includes the functional description of ARISE and introduces the delegation-based prediction method. The experimental evaluation and comparative analysis of this method together with the aggregation strategies are presented in Sect. 5. Finally, Sect. 6 introduces related work on group recommender systems, and Sect. 7 concludes the paper.

2 Generic architecture for group recommenders using social elements

ARISE[1] is a theoretical organization of the modules required to build social group recommenders. The architecture of ARISE is represented in Fig. 1. We can see that it is divided into six different modules: *cooperation*, *trust*, *individual prediction*, *explicit individual preferences*, and *product data*. This architecture allows us to simulate, in a realistic way, the social behaviour followed by groups of people when arguing on a joint activity.

The *personality* factor lets us model the behaviour of each member in a conflict situation such as the ability to agree on a common group activity. During this decision-making process each member must give up some preferences to reach a consensus. This preference variation is directly influenced by the confidence or social *trust* in other members of the group.

A basic building block of our group recommender is the individual estimation module that predicts the preferences for a given user. It requires an *explicit profile of the individual's preferences* and a *product data* set to be recommended. As we will see in Sect. 3.2, our *delegation-based prediction method* biases these individual estimations according to the personality and trust factors.

[1] Architecture for Recommenders Including Social Elements.

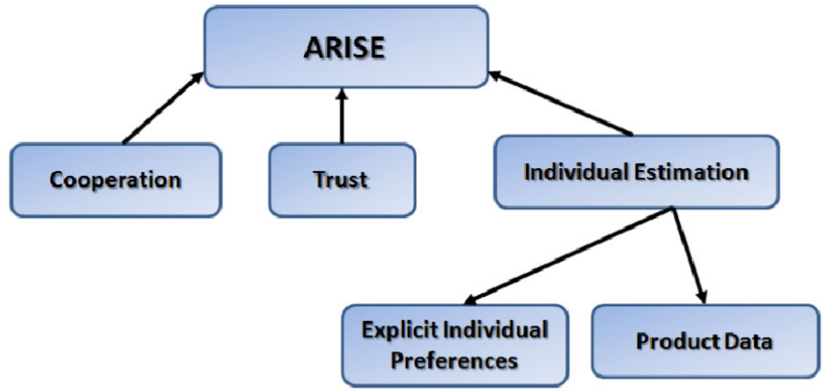


Fig. 1 Overview of ARISE

Finally, the information provided by each module is combined by the ARISE's aggregation techniques to obtain a recommendation for the group. These combination strategies are explained in Sect. 3 whereas the architecture modules are explained in the following sections.

2.1 Cooperation module

It is a fact that when we face a situation in which the concerns of people appear to be incompatible, *conflict situations* arise. Different people have different expectations and behaviour in conflict situations, and therefore they should be taken into account. When we started our research to improve the group recommendation process, we decided to study the different behaviours that people have in conflict situations according to their personality [41, 42, 44].

This module fulfils the task of obtaining a value that represents the personality of each user. This personality value, p_u, fits within a range of $(0, 1]$, 0 being the reflection of a very cooperative person and 1 the reflection of a very selfish one. In the ARISE architecture it is described as a high-level module that can be implemented in different ways depending on the resources available and the domain of the recommender application.

2.2 Trust module

Current research has pointed out that people tend to rely more on recommendations from people they trust (friends) than on recommendations based on anonymous ratings [47]. This social element is even more important when we are performing a group recommendation where users have to choose an item for the whole group. Note that *trust* is also related to *tie strength*; previous works have reported that both are conceptually different but there is a correlation between them [28].

This module fulfils the task of obtaining the trust values, $t_{u,v}$, between every user u and v that belong to the group that is being recommended. Note that $t_{u,v} \in (0, 1]$, 0 being the reflection of a person not to be trusted and 1 the reflection of a highly trusted one.

2.3 Individual estimation

Our recommendation approach predicts the rating that each user would assign to every item in the catalogue and then these estimated ratings are aggregated to obtain a global prediction for the group. Therefore, a basic building block of the architecture is the module in charge of computing individual predictions. We will denote the individual predicted rating as: $pred(u, i)$, u being a user and i an item from the catalogue. There are several options for obtaining these predictions that have been broadly studied in the recommendation research. In a general way there are two different approaches [45]. *Collaborative recommenders* use ratings already assigned by other users to several products. Users are selected according to their similarity with the target individual (by comparing the ratings given to the products). Most similar users are used as predictors and their ratings are combined to estimate the rating that the target user would assign to a new product. On the other hand, the *Content-based approach* compares each item to be proposed with items already rated by the target user. Then the ratings of the most similarly rated items are combined to provide a prediction.

Regardless of the approach chosen to implement this generic module of the ARISE architecture, there are two components (or sub-modules) that are always required by the individual recommender: explicit individual preferences and the product data set. Explicit individual preferences span any kind of information about the user that is required to predict the rating for a new item. Commonly, it just consists of the ratings given to some products in the catalogue. These ratings will be used later by the collaborative or content-based approach to predict new ratings. The product data module provides the information about the items in the catalogue that should be recommended to the group.

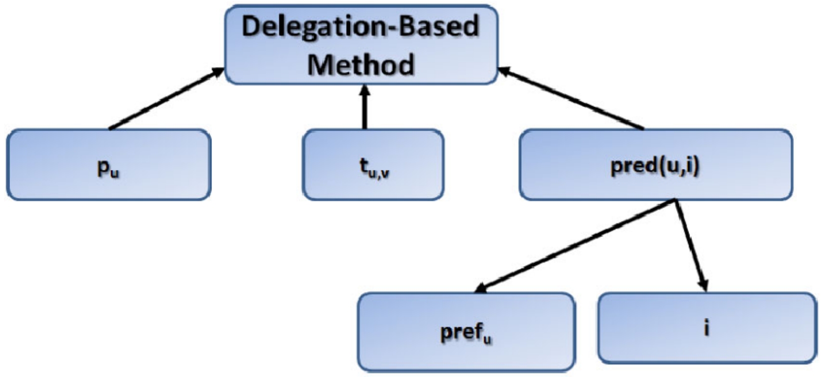


Fig. 2 Functional Description of ARISE

The next section presents the functional description of our approach.

3 Functional description of ARISE

In this section we explain the process of combining the social knowledge obtained from each of the ARISE's modules in order to provide a recommendation for the group.

Although we have already explained these modules, we summarize their output values to introduce some notation. These values and its corresponding dataflow are displayed in Fig. 2. The *cooperation* module obtains a personality factor p_u; the *trust* module returns the trust factor $t_{u,v}$; the *individual prediction* module obtains $pred(u,i)$ that is the result of applying a content-based predictor that compares the user's preferences $pref_u$, given by the *explicit individual preferences* module, to the *product data* module, that stores every item i in the catalogue.

Our group recommendation method is based on preference aggregation approaches. These approaches [33, 38] aggregate the individual ratings, predicted for every user u given an item i-denoted as $pred(u,i)$-, to obtain a prediction for the group:

$$gpred(G,i) = \bigsqcup_{\forall u \in G} pred(u,i) \tag{1}$$

Here G is a group of users, which user u belongs to, and $pred(u,i)$ is the individual prediction for user u and item i returned by the individual estimation module. There are several aggregation functions -represented with the $\sqcup$ symbol- that can be chosen to obtain the group prediction. These functions provide an aggregated value that predicts the group preference for a given item i. Then, our group recommender proposes the k items with the highest estimated group scoring.

As we previously explained, individual predictions are biased by our *delegation-based prediction* method that takes into account the personality and trust factors. This way, recommendations are computed as shown in (2).

Table 1 Example of a possible estimation of the ratings given by every user for different items. (Note that this table shows the values once they have been modified by our method to reflect personality and trust)

$dbp(u,i)$	Item A	Item B	Item C	Item D	Item E	Item F
u_1	5	2	1.5	3.5	5	4.5
u_2	0.5	4.5	4	4.5	3.5	4.5
u_3	5	2.5	1	3.5	4.5	4

$$gpred(G,i) = \bigsqcup_{\forall u \in G} dbp(u,i) \tag{2}$$

We will explain next the aggregation functions and later the *delegation-based* prediction method.

3.1 Aggregation functions

A wide set of aggregation functions has been devised to combine individual preferences [32]. Choosing the aggregation function that performs best is a key element in providing good recommendations. Here we explain the functions that we have studied for our social prediction method, *dbp*, which will be elaborated on in Sect. 3.2. We explain how to calculate group ratings with each of these methods through an example. Table 1 contains an example of predicted individual ratings returned by the *dbp* method, whereas Tables 2 to 8 show how these individual predictions are modified and/or combined in order to get the final group recommendation.

- **Average Satisfaction:** Refers to the common arithmetic mean, which is a method to derive the central tendency of a sample space [1]. It computes the average of the predicted ratings of each member of the group. The function

Table 2 Example of aggregation with Average Satisfaction (from ratings in Table 1). Predicted group preference: E, F ≻ D ≻ A ≻ B ≻ C

Avg. Sat.	Item A	Item B	Item C	Item D	Item E	Item F
Group prediction	10.5/3	9/3	6.5/3	11.5/3	13/3	13/3

Table 3 Example of aggregation with Borda Count (from ratings in Table 1). Predicted group preference: *F* ≻ A, E ≻ D ≻ B ≻ C

Borda count	Item A	Item B	Item C	Item D	Item E	Item F
u_1	4.5	1	0	2	4.5	3
u_2	0	4	2	4	1	4
u_3	5	1	0	2	4	3
Group prediction	9.5	6	2	8	9.5	10

that represents this strategy is:

$$gpred(G,i) = \frac{1}{|G|}\sum_{u\in G} dbp(u,i) \tag{3}$$

Where $dbp(u,i)$ is the socially modified predicted rating for each user u, and every item i. $gpred(G,i)$ is the final prediction of item i for the group. An example of this strategy is shown in Table 2.

– **Borda Count:** The Borda count is a single-winner election method in which users rank candidates in order of preference [46]. The Borda count determines the winner of an election by giving each candidate a certain number of points corresponding to the position in which s/he is ranked by each voter. Once all votes have been counted the candidate with more points is the winner. Because it sometimes elects broadly acceptable candidates, rather than those preferred by the majority, the Borda count is often described as a consensus-based electoral system, rather than a majoritarian one. We can see how the Borda count measure works in our example in Table 3. For instance, u_1 has the lowest rating for C, and hence, C is awarded 0 points. Next rating is for item B and it gets 1 point, and so on with the rest of its rankings. Finally, to obtain the group preference order, the points awarded to the individuals are added up.

$$\begin{aligned}
&gpred(G,i) = \sum_{u\in G} bs(u,i)\\
&bs(u,i) = pos\big(i, OL(u)\big)\\
&OL(u) = \{i_1,, i_2, \ldots, i_n\}\\
&\quad \text{where } dbp(u,i_p) \le dbp(u,i_{p+1})
\end{aligned} \tag{4}$$

Where $bs(u,i)$ is the Borda score assigned to each item rated by user u. It is obtained as the position of the item i in the ordered list OL. This list arranges the items according to the ranking estimated for user u. A problem arises when an individual has multiple alternatives with the same rating. In this case we have decided to distribute the points. So, for example, in u_2's list B, D and F share the place and get $(3+4+5)/3 = 4$ points each. (Note that this modification is not included in the previous formula for the sake of readability.)

Table 4 Example of aggregation with Copeland Rule (from ratings in Table 1). Predicted group preference: A, E ≻ F ≻ D ≻ B ≻ C

Copeland rule	Item A	Item B	Item C	Item D	Item E	Item F
Item A	0	−1	−1	−1	0	−1
Item B	+1	0	−1	+1	+1	+1
Item C	+1	+1	0	+1	+1	+1
Item D	+1	−1	−1	0	+1	+1
Item E	0	−1	−1	−1	0	−1
Item F	+1	−1	−1	−1	+1	0
Group prediction	+4	−3	−5	−1	+4	+1

– **Copeland Rule:** Alternatives are ordered by the number of pairwise victories, minus the number of pairwise defeats. It is a good procedure to overcome problems resulting from voting cycles [26]. In the example A beats B as both u_1 and u_3 prefer it, so the result in Table 4 shows a +1 for column A vs row B.

$$\begin{aligned}
&gpred(G,i) = \sum_{j\in Catalog, j\neq i} cs(i,j)\\
&cs(i,j) = \begin{cases} +1 & \text{if } wins(i,j) > losses(i,j)\\ -1 & \text{if } wins(i,j) < losses(i,j)\\ 0 & \text{a.o.c.}\end{cases}\\
&wins(i,j) = |u\in U : dbp(u,i) > dbp(u,j)|\\
&losses(i,j) = |u\in U : dbp(u,i) < dbp(u,j)|
\end{aligned} \tag{5}$$

– **Approval Voting:** This is a single-winner voting system used for elections. Each voter may vote for (approve of) as many of the candidates as they wish. The winner is the candidate that receives more votes [17]. In our example, we could assume that u_1, u_2 and u_3 vote for all alternatives with a rating above a certain threshold δ, meaning that they vote for any alternative provided that it seems a little interesting for them. An example of this strategy is reflected in Table 5 with $\delta = 2.5$.

$$\begin{aligned}
&gpred(G,i) = \sum_{u\in G} as(u,i)\\
&as(u,i) = \begin{cases} 1 & \text{if } dbp(u,i) \ge \delta\\ 0 & \text{a.o.c.}\end{cases}
\end{aligned} \tag{6}$$

– **Least Misery:** This strategy follows the idea that, even if average satisfaction is high, a solution that leaves one

Table 5 Example of aggregation with Approval Voting ($\delta = 2.5$) (from ratings in Table 1). Predicted group preference: D, E, F $\succ$ A, B $\succ$ C

Approval voting	Item A	Item B	Item C	Item D	Item E	Item F
u_1	1			1	1	1
u_2		1	1	1	1	1
u_3	1	1		1	1	1
Group prediction	2	2	1	3	3	3

Table 6 Example of aggregation with Least Misery (from ratings in Table 1). Predicted group preference: F $\succ$ E, D $\succ$ B $\succ$ C $\succ$ A

Least misery	Item A	Item B	Item C	Item D	Item E	Item F
Group prediction	0.5	2	1	3.5	3.5	4.5

Table 7 Example of aggregation with Most Pleasure (from ratings in Table 1). Predicted group preference: A, E $\succ$ B, D, F $\succ$ C

Most pleasure	Item A	Item B	Item C	Item D	Item E	Item F
Group prediction	5	4.5	4	4.5	5	4.5

Table 8 Example of aggregation with Avg. Without Misery, $\delta = 2$ (from ratings in Table 1). Predicted group preference: E, F $\succ$ D $\succ$ B

Avg. w/out misery	Item A	Item B	Item C	Item D	Item E	Item F
Group prediction	–	9/3	–	11.5/3	13/3	13/3

or more members very dissatisfied is likely to be considered undesirable. This strategy considers that a group is as happy as its least happy member. The final list of ratings is the minimum of each of the individual ratings. A disadvantage can be that even if the majority really likes one item, if one person does not, then it will never be chosen [32]. An example of this is shown in Table 6 where u_1 and u_3 vote very highly for item A but its final rating is the lowest one, because u_2 does not like it.

$$gpred(G, i) = \min_{u \in G} dbp(u, i) \tag{7}$$

– **Most Pleasure Strategy:** It is the opposite of the previous strategy, Least Misery; it chooses the highest rating for each item to form the final list of predicted ratings [32], as we can see in Table 7.

$$gpred(G, i) = \max_{u \in G} dbp(u, i) \tag{8}$$

– **Average Without Misery:** Assigns a preference to the average of the weights in the individual ratings. The difference here is that those items that have predicted ratings under a certain threshold will not be considered [32]. Table 8 shows an example of how the group ratings are calculated using a threshold of $\delta = 2$.

$$\begin{aligned} gpred(G, i) &= \frac{\sum_{u \in G} pred_{wm}(u, i)}{|u \in U : dbp(u, i) > \delta|} \\ pred_{wm}(u, i) &= \begin{cases} dbp(u, i) & \text{if } dbp(u, i) > \delta \\ 0 & \text{a.o.c.} \end{cases} \end{aligned} \tag{9}$$

Once we have described the aggregation functions that can be used to combine individual predictions, the following section details how these individual estimated ratings are modified with our social factors. We present, as the core of ARISE, our *delegation-based* method, that improves group recommendations by means of personality and trust factors.

3.2 Modifying individual predictions with social elements

The idea adopted in our method is that everyone is influenced by their social context. Social media highly influences our decisions, relationships, and education. Several researchers study the impact of social media in our lives [12]. The social context refers to the immediate physical and social setting in which people live. It includes the culture that the individual was educated or lives in, and the people and institutions with whom they interact. Circumstantial life events, influences, and surroundings can further change our behaviour [7]. Social elements, that in our social recommendation method are the personality and trust factors, define each person (our users involved in the recommendation processes) as a potentially influenced component of a social community or group determined by the environment, in most cases social networks, s/he belongs to. In our social method, we have simulated people's behaviour based on the idea that the relationship between individuals and their networks of people directly influence their lives [12]. This way, we use the trust factor to model the impact of the preferences of the people that belong to the close circle of the user in her/his social environment and that therefore might influence her/him. This proximity between users (users' trust) is obtained by analyzing the information available on the social network. But, the influence of other group members not only depends on their proximity or trust in them, but also on the degree of personality or leadership of these influencers and on the degree in which the user might be influenced according to her/his personality. This degree of compliance or leadership is computed through the personality factor with the *assertiveness* and *cooperativeness* dimensions (as we will explain next in Sect. 4.1).

Hence, our recommendation approaches consist of evaluating the different behaviours that people have when participating in a decision-making process. To do so we use the personality and trust factors to modify the predictions made by the individual recommender. In that way not all the predictions are taken into account equally. We use a novel approach, which we have named *delegation-based* prediction method, to compute the new individual prediction, $dbp(i,u)$, used in (2).

The idea behind this approach is that users create their opinions based on their friends' opinions. So basically, in each user's turn in $\forall u \in G$, $|G| = n$. In (2), the user's opinion is not taken into account, but in the other $(n-1)$ turns, the user influences others. Instead of storing the information contained in a user's opinion just once, the method takes it into account every time another user of the same group states an opinion. We know that this idea is not at all intuitive. However, we performed several experiments with other simpler methods and they all provided worse recommendations than our *dbp* method. The delegation-based prediction method tries to simulate the following behaviour: when we are deciding which item to choose within a group of users we ask people whom we trust. This method follows a collaborative approach where a user's opinion is generated based on others' preferences. This way we apply the principles of emotional contagion. Moreover, we also take into account their personality in order to give certain importance to their opinions (for example, because we know that a selfish person may get angry if we do not choose her/his preferred item). The tie strength is also reflected in the formula by means of the trust between the users. The delegation-based prediction, $dbp(u,i)$, given a user u and an item i is computed in this way:

$$dbp(u,i) = \frac{1}{T} \sum_{v \neq u \in G} t_{u,v}\big[pred(v,i) + \theta_{v,i} \cdot (p_v - p_u)\big]$$
$$\text{where } T = \sum_{v \neq u \in G} t_{u,v} \tag{10}$$

In this formula, we take into account the predicted preference $pred(v,i)$ of every friend v for item i. This rating is increased or decreased depending on the differences of personality between both friends, $p_v - p_u$. This way if user v has a strong personality s/he will have a higher impact on the prediction for user u. However, it is important to note that a user v with a strong personality and a high preference for item i, $pred(v,i)$, would try to increase the opinion of user u about that item. In the opposite case, a low preference for the item, user v would try to decrease u's opinion. This behaviour is modeled using the $\theta_{v,i}$ parameter as follows, lets say that $pred(v,i)$ is in a range of $[a,b]$:

$$\theta_{v,i} = \begin{cases} 5 & \text{if } pred(v,i) \geq \frac{b-a}{2} \\ -5 & \text{if } pred(v,i) < \frac{b-a}{2} \end{cases} \tag{11}$$

We have chosen those constant values (5 and -5) because the mean difference in the personality values is 0.2 and therefore the impact of the difference of personality in the formula will be $\sim \pm 1$. Finally, the prediction of user v that has been modified according to the personalities is also weighted by the trust between both users $t_{u,v}$. Note that this formula is not normalized by the group size and uses the accumulated trust[2] (represented as T). We have chosen this option following the findings of [20] where a method for group recommendations using trust is proposed.

We will now explain the details of our case study *HappyMovie*:[3] a Facebook application for recommending movies to groups of users.

4 Case Study: HappyMovie

HappyMovie is a particular instantiation of our generic ARISE architecture for the movie recommendation domain. It serves as a use case and experimental environment where we can evaluate our architecture with real products. This way we can validate and improve our previous results obtained in simulated environments [41, 42, 44].

This application has been developed for Facebook. With it we are able to offer group recommendations to people connected through this social network and obtain valuable feedback.

There are several reasons for this choice. Firstly, Facebook is used by users to create events and invite their friends to join activities, so our system can help them in the organization of such events. Secondly, users' activities in the social network can be tracked to obtain information about their trust with other users. And finally, it is a perfect environment to obtain users' social factors required by our model as it is user-friendly, easily accessible, has a lot of daily users and is adapted to run questionnaires, applications and games.

HappyMovie's architecture is depicted in Fig. 3. It is easy to compare it with the generic design of the ARISE architecture (Fig. 1) described in Sect. 2. Next we summarize the way we have implemented the generic modules in our concrete system whereas Sect. 4.2 presents a functional overview of *HappyMovie*.

4.1 *HappyMovie* modules

HappyMovie instantiates the generic architecture of ARISE through the following modules (Fig. 3):

Cooperation, TKI metaphor: There are different approaches that can be used in order to obtain the different personalities or roles that people play when interacting in a decision making process. In our previous studies [41, 42, 44]

[2] Trust values always are greater than 0 so we do not have problems with this normalization.

[3] http://www.happymovie.net.

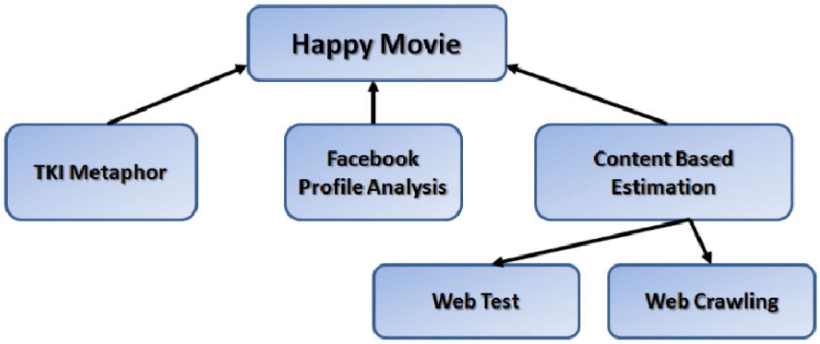


Fig. 3 Overview of *HappyMovie*

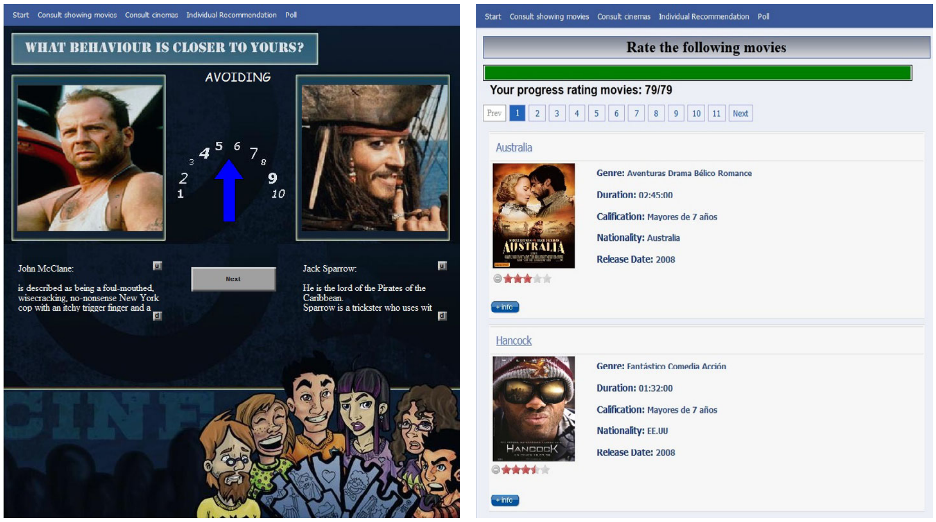
we used the Thomas-Kilmann test [48]. We chose this test because it is the most commonly used in the human-machine interaction area, due to its efficiency and that it is easy to evaluate and use for people not related to the psychology area. It provides a tangible and measurable value, easier to interpret than similar tests. According to this test, we can describe an individual's behaviour along two basic dimensions in conflict situations: (1) *assertiveness*, the extent to which a person attempts to satisfy her own concerns, and (2) *cooperativeness*, the extent to which a person attempts to satisfy other people's concerns. These two basic dimensions of behaviour define five different modes of responding to conflict situations: Competing, Accommodating, Avoiding, Collaborating and Compromising.

Our approach combines these 5 modes to obtain a personality value, p_u, representing the user's personality. To obtain the score that the user has in each mode, the TKI personality test proposes 30 situations where the user has to think about how s/he will react. Initially we used the original TKI test. However, when we asked our users about it, they described the test as tedious, long and not very clear in some of the questions. To make the application more easy to get through, in [43] we studied and validated the use of an alternative method to obtain data about a certain user's personality. It consists of an interactive metaphor that displays two well known movie characters with opposite personalities for each of five possible categories. One character represents the essential characteristics of one category, while the other one represents all the opposite ones. What the user has to do is to choose, using a moving arrow, with which of each pair of characters s/he feels more identified. We have performed an experiment with real users using both tests, and proven that it is possible to replace the original TKI test with the new one (the metaphor) because the results obtained with the two tests are equitable (see [43] for the details of the experiment). In Fig. 4(a) we can see how the personality test is presented in the application.

Trust, Facebook profile analysis: The Trust Module is the module that receives the largest benefit because the application is embedded in a social network. We are able to calculate the trust between users by extracting specific information from each of their own profiles in the social network. Facebook users usually post a huge amount of personal information that can be analysed to compute the trust in other users: distance in the social network, number of comments shared, likes and interests, personal information, pictures, games, duration of friendships, etc. [18, 19].

In order to switch from theory to practice it is important to take into account that these elements are not easy to quantify and that obtaining them is limited by the extraction power that Facebook APIs give us. In *HappyMovie* we analyse the following factors: common friends, pictures in common, common interests (music, movies, series..) and comments on each other's Facebook walls. Afterwards, these factors are combined using a weighted average. We have adjusted the weights of these factors when calculating trust after an experiment with real users where they indicated the real trust that they had in each other. The trust between pairs of users is computed every time a user joins an active group (or in terms of *HappyMovie*, when a user joins an event related to a cinema outing). This calculation is done between the active user and the rest of the group members. For each pair of users and each event the trust value is only computed once. However, we do compute it again for each new event as Facebook profiles keep changing and so does the trust between two people. A detailed explanation of the trust factors obtained from Facebook and the combination process is provided in [42].

Individual prediction: Our group recommendation strategies combine individual predictions to find an item (movie) suitable for the group. This individual prediction module is built using the jCOLIBRI framework [15] and follows a content-based approach [40] to estimate the ratings a user would assign to each product in the catalogue. It compares the description of each product in the catalogue and selects those ones that are most similar to the user's preferences, and therefore, have the highest estimated rating. We have chosen a content-based system and not a collaborative one [16] because the movies to be recommended (i, in Fig. 2) are



(a) Personality test in *HappyMovie* (b) Preference test in *HappyMovie*

Fig. 4 User tests in *HappyMovie*

too recent to have enough user ratings. Therefore, we could not use those ratings as collaborative recommenders do.

Consequently, this module has two requirements that must be fulfilled: the catalogue of products to be recommended, and the individual preferences of each user. In *HappyMovie* we obtain them with two sub-modules: a web crawling module that obtains new movie listings directly from the web and a web test module that obtains users' preferences.

To obtain the catalogue of products we have implemented a web crawler that obtains new movie listings from the web. This module is executed off-line and creates a data base of movies being played in cinemas. This data base also contains information about the location of the cinemas, the description of the movies and any other data required by our system.

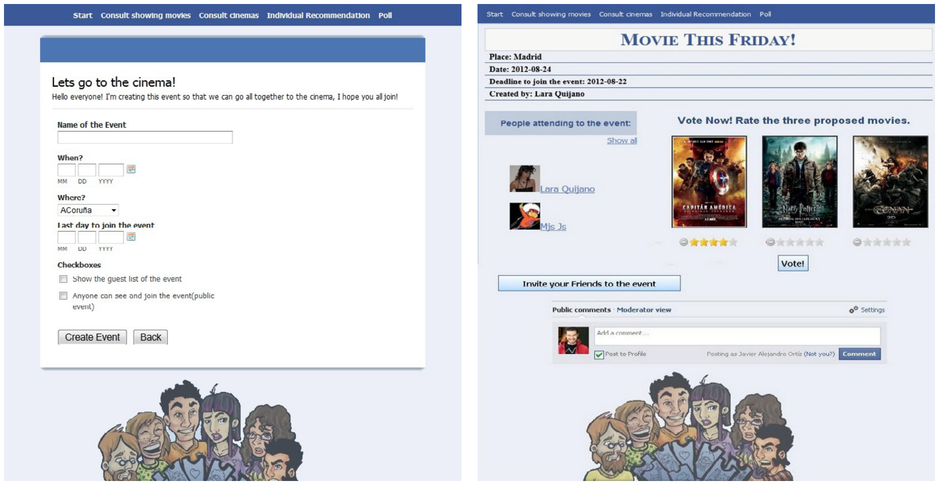
The web test module is in charge of obtaining users' preferences for movies. It consists of a test where users are provided with a set of heterogeneous movies that they should rate (20 at least) in a Likert scale from 0 to 5, as shown in Fig. 4(b). This test must be run before using the *HappyMovie* application although it can also be run on demand to increase the accuracy of the system. These preferences will be later used to evaluate the satisfaction of users regarding the items proposed to the group.

Having described the implementation of the *HappyMovie* application we will briefly detail its behaviour to let readers understand its functionality.

4.2 Using the *HappyMovie* system

The necessary steps to obtain a movie group recommendation with *HappyMovie* are:

1. **Prerequisites**. Before any user can access the movie recommendation functionality we collect the individual information required by our recommendation method. As we have previously explained, this information is the user's personality, trust and individual preferences.
   - In order for us to gather the necessary data about the user's personality, he or she will be made to choose among a set of characters the one they feel the most identified with (cooperation module), as shown in Fig. 4(a).
   - In order to store information about users' preferences, he or she will be made to rate a set of movies (at least 20 movies), where they enter their personal preferences (this is the web test used by the individual prediction module), as shown in Fig. 4(b).
   - In order to understand the user's circles of trust the application reads the information stored in the Facebook personal profile. It calculates the trust that the active user has in all the other users that have joined the event up to now.
2. **Activity definition**. *HappyMovie* identifies two different user roles. `Organizers` create the events as shown in Fig. 5(a) and define the place, date or invited people. `Attenders` accept the invitation (delivered through the Facebook capabilities) and can see the movies proposed by the system based on the current configuration of the group. As attenders they can invite further users or withdraw from the event: recommendations are proposed dynamically.
3. **Final choice**. Once the deadline is reached, the system recommends the (estimated) three best movies for the group. At this point they are allowed to rate each movie



(a) Activity creation in *HappyMovie* (b) Group recommendation in *HappyMovie*

Fig. 5 Events in *HappyMovie*

individually. This process lets them decide which movie they are finally going to watch and, more importantly, it gives the system the feedback required to evaluate the level of satisfaction of the group.

In next section we present an evaluation of our system where we compare the performance of the *delegation-based* prediction method, all the different aggregation functions and the impact of social factors.

5 Experimental evaluation

We have evaluated our movie recommendation method making use of *HappyMovie*. We have firstly performed our experiment with groups of real users and secondly repeated it with synthetic data. The most important goal of our experiment is to measure the performance of our group recommendation method in a real scenario. However we have decided to experiment also with synthetic data in order to explore extreme cases that could appear in conflict situations. We also want to have control of the data distribution, an impossible situation when using real data. This synthetic data lets us explore every group composition and personality distribution within a group. It also lets us reproduce the behaviour of large groups that are very difficult to organize in experiments with real users. In [42] we used this very same method to create synthetically generated data and proved that the results obtained were valid and equivalent to the ones obtained with real data.

The other goals of this experiment are:

- Prove that the delegation-based prediction method has a higher performance than the standard non-social group recommendation approach.
- Study which of the possible aggregation functions reports the highest performance with the *dbp* method.
- Analyse the impact of the personality and trust factors in the *dbp* method.

5.1 Experimental set-up

We developed a configurable group recommender implementing the aggregation strategies described in Sect. 3.1. This recommender can be configured to use the standard individual predictions returned by the individual estimations module, $pred(u, i)$, or our social-based prediction method, $dbp(u, i)$.

The inputs of the recommender are those defined by the modules in ARISE: personality p_u, trust t_u, and individual preferences $pref_u$ for each user u. The output is a set of items recommended for a given group configuration $rec(G)$. Finally, the validation data is the real group choice: an ordered list with the favourite items that the group would actually have chosen $fav(G)$. The size of both lists was limited to 3 items assuming that it is the maximum number of movies that a user/group would be really interested in watching at a time. The accuracy of the system will be measured by comparing $rec(G)$ and $fav(G)$. The more the recommended movie list resembles the real one, the better results our application provides. The evaluation metrics applied to compare both sets are explained in Sect. 5.2.

Each configuration of the recommender was evaluated with two different input datasets. The first one was obtained from real users and the second dataset contains information from synthetically generated users. This artificial dataset let us explore the behaviour of the recommender with extreme or unusual group configurations. For example, we analysed the range of personality values for the real dataset and almost every user had a mild personality. Therefore we could not conclude whether or not our method performs accurately with extreme personalities. Additionally, the real-users dataset has a limitation regarding the size of the groups. To study the performance with large groups (over 10 members) we needed a considerable number of participants. The synthetic dataset solved this limitation and reproduced such an amount of users. We must note that the validity of this dataset has been already proven in our previous studies [42]. Next we describe the features of each dataset:

- **Real dataset**: As we mentioned above, the most important goal of our experiment was to measure the performance of our group recommendation method in a real scenario. To do so we used our Facebook application *HappyMovie*. We created different events in the social network as explained in Sect. 4 and asked volunteers to use it. The demographic data about our participants (mean age, gender, etc.) was quite varied because they were selected among colleagues and students. 58 users participated in our experiment. Input values p_u and $pref_u$ were obtained from the tests presented in Sect. 4 (Figs. 4(a) and 4(b)). The last input t_u was obtained by analysing users' Facebook profiles. The validation data $fav(G)$ was obtained by putting together groups of users that simulated going to the cinema together and gave us a 3 item list that contained the group choice. We managed to gather 15 groups of 9, 5 and 3 members (4, 6 and 5 groups respectively). To obtain the output list $rec(G)$, users created events in *HappyMovie* and joined them with the same configuration as they did in the simulation.
- **Synthetic dataset**: This second dataset lets us explore unusual group configurations. By using this approach we were able to group users in sets of 3, 5, 10, 15, 20 and 40 people.

  Personality p_u is assigned randomly, but following certain restrictions, to ensure that we obtain groups composed of people with all the possible combinations of extreme personalities (very selfish, selfish, tolerant, cooperative and very cooperative).

  In the end we had 76 groups (13 different distributions for each size, except for the 40-person group where we only had 11 combinations due to the similarity of personalities in such big groups). The second input variable is the individual preferences $pref_u$. This is a very delicate step that we have resolved by assigning profiles to each user. These profiles are generated from the Movielens data set [10] according to typical preferences about movies stratified according to their age, sex and likes.

  The last input, t_u, was assigned randomly to each simulated user according to the typical distribution of trust in a population. To obtain the validation data $fav(G)$ we asked our volunteers to estimate which movies each artificial group would have chosen. The recommended output list $rec(G)$ was computed by applying our recommendations algorithms to the input data.

5.2 Evaluation metrics

Our experiment requires an evaluation function to measure the accuracy of the group recommendation. To do so, we compared the results of our recommender system $rec(G)$ to the real preferences of the users $fav(G)$. However the choice of a suitable evaluation metric requires the consideration of several factors.

The first one is the length limitation in the $fav(G)$ list. Real users are only interested in a few movies they really want to watch and consequently we limited them to 3 elements. Therefore, we cannot use general measures like recall or precision. Secondly, $rec(G)$ is an unordered set because our recommender proposes three movies without any kind of ranking that are afterwards voted on by the members of the group to make their decision. This feature discards several evaluation metrics that compare the ordering of the output and validation lists like the Mean Absolute Error (MAE) [2, 22] or the Normalized Discounted Cumulative Gain (nDCG) [5].

However, there are some metrics used in the Information Extraction field [49] that are suitable for our scenario. In our case, we can use *precision@3* to evaluate how many of the movies in $rec(G)$ are in the $fav(G)$. This kind of evaluation can be seen from a different point of view: we are usually interested in having at least one of the movies from $rec(G)$ in the $fav(G)$ list. This measure is called *success@3* and returns 1 if there is at least one hit in the first *3* positions. Therefore, we could use $success@3$ (or simply $s@3$) to evaluate our system by computing the rate of recommendations where we have at least one hit in $fav(G)$. For example, 90 % accuracy using $s@3$ represents that the recommender suggests at least one correct movie for 90 % of the groups being evaluated. In fact, $s@3$ is equivalent to having $precision@3 > 1/3$. We can also define a $2s@3$ metric (equivalent to $precision@3 > 2/3$), which represents how many times $fav(G)$ contains at least two movies from $rec(G)$. Obviously, it is a much more restrictive measure.

5.3 Results

In this section we detail the results obtained with different configurations of the recommender. Each configuration is defined by the input dataset (real or synthetic), the aggregation function (from Sect. 3.1) and the estimation method: $pred(u,i)$ or $dbp(u,i)$. As we have have studied 7 aggregation strategies we have ended up with 28 different configurations. To simplify the reporting of the results we group these configurations into two sets according to the estimation method. Those configurations using the basic estimation $pred(u,i)$ without social knowledge conform the

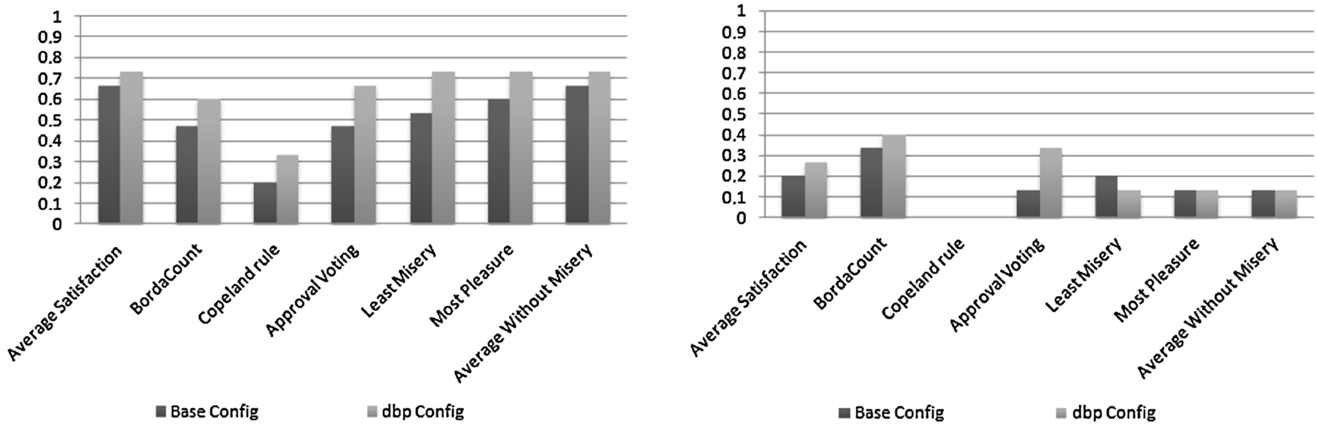


Fig. 6 Performance of the base and dbp configurations w.r.t. each average function using the real dataset and the s@3 (*left*) and 2s@3 (*right*) evaluation metrics

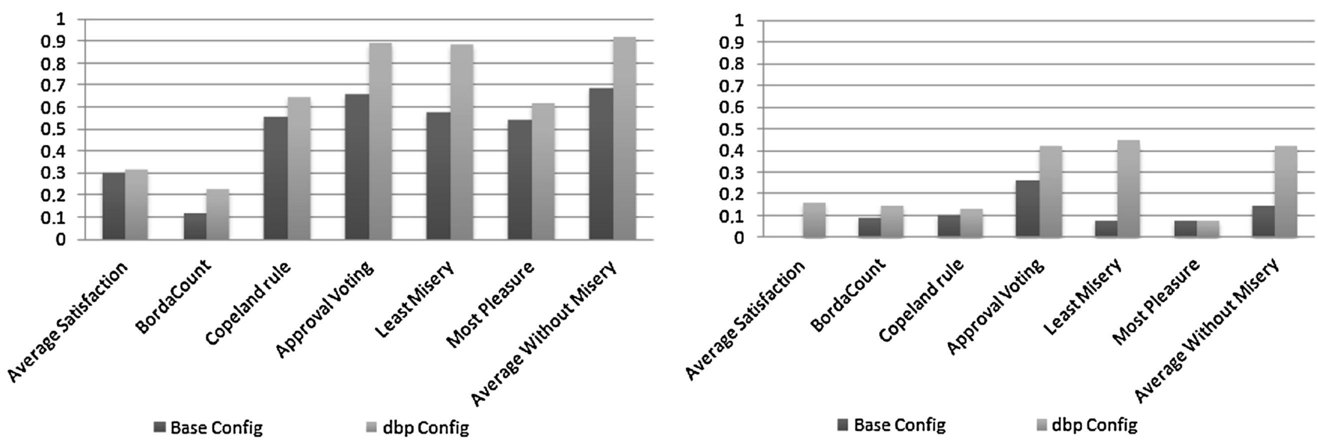


Fig. 7 Performance of the base and dbp configurations w.r.t. each average function using the synthetic dataset and the s@3 (*left*) and 2s@3 (*right*) evaluation metrics

baseline of the results and will be referred to as "Base configuration". Complementarily, configurations using the *delegation-based* prediction method will be labeled as "dbp configuration".

First, we analysed the improvement of the dbp configuration with respect to the baseline for each aggregation function. In Figs. 6 and 7 we can see the comparison of the results for the real data and the synthetic dataset. In average the improvement rates for the real dataset are 13.33 % with $s@3$ and 3.8 % with $2s@3$. For the synthetic dataset improvements are 15.22 % and 14.84 % for each evaluation measure respectively.

Next, we explored the correlation of these improvements to the social factors. Our delegation-based method integrates two different social factors: personality and trust. To figure out the impact of this social behaviour knowledge in the recommendation process we repeated the evaluation of the dbp configurations with three different flavours of the *dbp* method where each social factor was nullified: only personality dbp_p, only trust dbp_t and no social knowledge at all dbp_{null}. This last variant let us measure the impact of the collaborative approach followed by the *dbp* method where individual preferences are predicted by averaging other users' preferences. Results are presented in Fig. 8. As we can observe, these variants can achieve at the most the same results as the full dbr approach for several aggregation functions. However it is not possible to generalize these results because their accuracy varies depending on the aggregation strategy and the dataset being used. Therefore we cannot conclude that, in general, these variants can be used to maximize the performance of a global system that works with different group configurations. Nevertheless, Fig. 8 illustrates a relevant finding: the full dbr method always performs at least as well as the other variants. It is the best way to balance the social factors included in our recommendations model: personality and trust. The statistical significance of these results was confirmed using the Wilcoxon signed-rank test ($p < 0.05$). Consequently, we can conclude that our delegation-based prediction method significantly

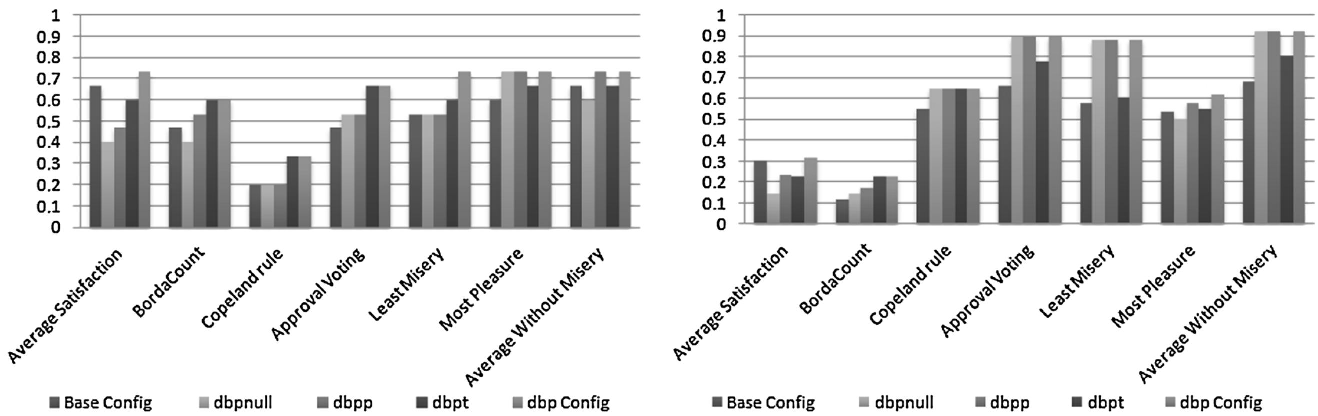


Fig. 8 Performance of the dbp variants w.r.t. each average function using both datasets real (*left*) and synthetic (*right*) and the s@3 evaluation metric

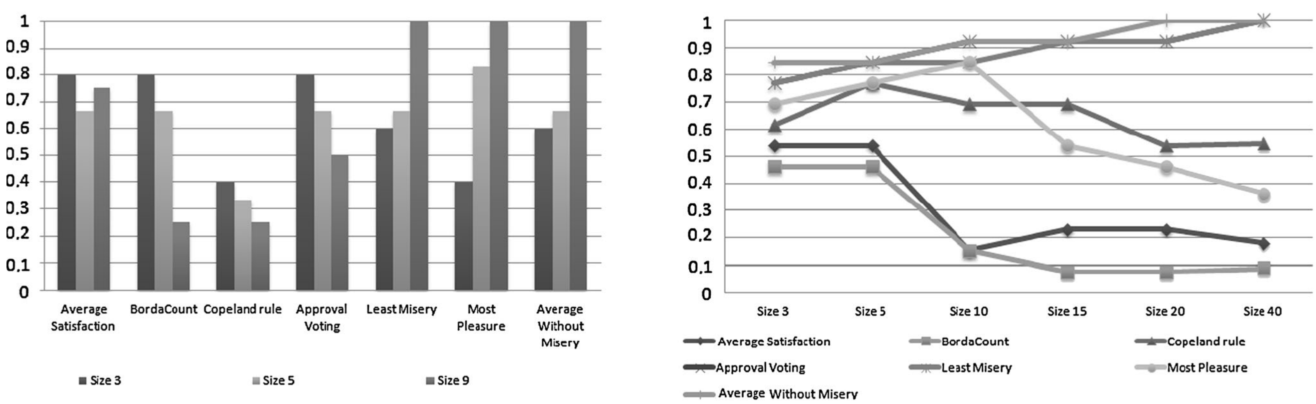


Fig. 9 Performance of the dbp configuration and group sizes w.r.t. each average function using the real dataset and the s@3 evaluation metric

improves standard aggregative recommendation approaches and performs at least as well as the other variants.

Finally, we analysed the performance of the delegation-based prediction method according to group size.

Figure 9 illustrates the results achieved by *dbp* applied to the real and synthetic datasets, for different group sizes and every aggregation function. Upon analysing the results from the real dataset (Fig. 9 (left)) we can observe that while some aggregation functions such as *average satisfaction* report better results for small groups (we consider groups of 10 or less as small), others like *least misery*, *most pleasure* or *average without misery* work the other way round and obtain better results for large groups. This is the reason why we needed synthetically generated data: to study the specific aggregation strategy that each possible group size requires. However, we can conclude that, on average, the best aggregation function for such small groups is the *average satisfaction* strategy.

Figure 9 (right) shows the results for the synthetic dataset. Here we can confirm that *least misery* and *average without misery* are the optimal aggregation functions for large groups. Both show a rise in their performance when applied to a larger group. We can also observe that the *average without misery* strategy has decreasing performance although it reported good results for 9-member groups in the experiment with real data. When analysing the behaviour of the *average satisfaction* strategy we can confirm its unsatisfactory performance with large groups.

A related work section will be introduced next explaining all the different works that have been done in the field of group recommenders and their differences with our system.

6 Related work

There are a lot of domains where group recommendation techniques can be applied. For example, in the music domain, the work presented in [30] uses an algorithm that personalizes the distance measure between different pieces of music based on user preferences. MusicFX [35] provides recommendations about the background music at a fitness centre based on the preferences provided in different musical genres by the users. We can also find FlyTrap [13], a group recommender which selects music to be played in a public room. In the movies domain, Polylens [38] is an extension of Movielens to generate recommendations to groups. Regarding recommendations of restaurants for groups, we can find an interesting recommender system, Pocket Restaurant Finder [34], which bases its strategy on users' locations and the culinary characteristics of the restaurant. To find the best TV program to watch we have YuTV [51], which uses a vector space model with features of the TV programs (such as genre or actors) to find relevant recommendations for groups. LET'S BROWSE [29] is another example of group recommendation; this one recommends web pages to a group of two or more persons who are browsing the web together.

What all these recommenders have in common is that they take into account personal preferences obtained from their users; however, they consider each user to be equal to the others. The recommendation is not influenced by their personality or the way each one behaves in a group when joining a decision-making process. In our approach we propose to study how people interact depending on their personality or their closeness in order to improve group recommendations.

6.1 More than preference aggregations

There are also other works, besides our own, that do take into account not only the preferences of every member but also the interaction among them; Travel Forum Decision [23] is an example of this. The goal of this application is to help groups of users plan their vacations together. The system provides a solution and allows group members to discuss. It acts as a mediator until they reach a solution. In our approach, we propose to simulate this discussion in order to relieve our users of the process of discussing about choosing a solution. This way, our proposal requires less interaction from the users and presents an immediate solution.

When considering more than just individual preferences, there are some systems, such as CATS [36] which is a conversational recommender for planning skiing holidays, that takes into account the attitudes and behaviour of other group members. Other systems that make a more detailed study of the group before making any recommendations are Intrigue [4], which plans visits for groups of tourists by weighting the preferences of different subgroups with special needs (like children or disabled people). Chen [11] proposes the use of genetic algorithms to learn group preferences by using the known preferences of the subgroups within a group. Although the results seem to be significant, they suppose that groups are fixed and they have previously rated some items together. In the case of CATS, the recommendation is defined as an incremental process where users collaboratively refine the suggested recommendation by critiquing its features or discarding it. They consider that the preferences of the current member partially depends on the preferences and/or the anticipated behaviour of other members. During the process of choosing a recommendation, users can see what other members have voted for, so they are conditioned by other members's opinions. CATS users need to read the information of other users in order to alter their initial opinion. Obviously this is only possible for users who vote later. Our approach simulates this conditioning more thoroughly, because it can simulate these alterations beforehand by taking into account the strength of the relationships between group members.

Other works focus on the integration of group disagreements in the recommendation process. One of the most recent systems is GRec-OC [25], a book recommender system for online communities. GRec-OC provides recommendations based on the books that other similar groups have purchased and tries to reduce the dissatisfaction of individual members. The work in [3] proposes a recommender that aggregates prior group member preferences to create the recommendation. Then, preference disagreements between pairs of individuals are collected and employed to score and rank the recommended items. Finally, Masthoff and Gatt [33] use individual satisfaction and emotional contagion in order to recommend a sequence of video clips for a group. The authors think that a member changes the selection of her/his best clip according to the clip selected during the previous selection step. This change can be reflected in the recommendation algorithm as an individual satisfaction function that computes the individual affective state. This state influences the affective state of the other members, producing an emotional contagion that should be taken into account during the recommendation process. Additionally, they point out a tendency in the influence of social status on the selection process.

Summing up, we conclude that there is a need to adapt the recommendation process to group composition [24, 32]. This is backed up by some recent works that have focused their studies on analyzing the effectiveness of group recommendations according to different aspects, such as group size and inner group similarity [5], or on studying different weighting models to combine the preferences of group members according to their activity or role within the group [8]. Additionally, it is also known that a user's preferences can be affected by the rest of the group [11, 32]. However, most of the aggregation strategies employed in previous works combine users' preferences without taking into account either the relationships between group members or the relevance of each member's preferences. The work dealing with these issues is limited. We observed that there was a need to modify those existing strategies that consider each user of the group to be equal to the others. So we focused our line of work on reflecting each user's individual aspects and how they interact with each other.

In [44] we presented an improvement of current group recommendation techniques by introducing a novel factor: the personality of each individual in the group when dealing with conflict situations. We use a personality test to obtain the different roles that people play when interacting in a decision-making process. Once we studied the individual characterization of people in a group, we decided to study other factors regarding the structure of the group itself and how users interact with each other. The inclusion of the individual personality factor wasn't enough to achieve this, due to the increasing importance of social networks and the trust connections that they imply. Therefore we needed to explore more social factors.

6.2 Social recommendations in social networks

In the last few years researchers have proved that the inclusion of social aspects in the recommendation processes improves the recommendation accuracy [14]. Social networks such as Facebook or Twitter can provide a rich mine of resources and the possibility of acquiring data about the user's circles of trust. These networks contain implicit information that can be used in a recommendation process [27]. This option, which is completely transparent to users, has as a main advantage that users are not required to provide explicit information about their trust in other users. This information is extracted implicitly from their daily interaction in the social network. However, it has the obvious drawback that every user involved in the group recommendation process must belong to the social network. Nevertheless, the rising popularity of this kind of web applications minimizes this risk. Even more, it is becoming usual to organize events (such as going to the cinema) through social networks, so group recommendation techniques could be integrated into these web sites. In [19] Golbeck proposes a methodology to infer relationships of trust within social networks. The computational problem of trust is to determine how much one person in the network should trust another. Certainly, trust inferences will not be as accurate as a direct rating. But the algorithm presented in this study, named TidalTrust, managed to improve the accuracy by 10 %.

It is a fact that people rely more on recommendations from people they trust (friends) than on recommendations based on anonymous ratings [47]. This is a very important factor in group recommendation strategies, when a decision for the whole group has to be made. This kind of recommendations usually follows an argumentation process, where each user defends her preferences and rebuts others' opinions. Here, the tie strength or trust between users is crucial because they must adjust their opinions in order to reach a common decision.

The generation of trust models has created a huge body of work. The emergence of the current collaborative web (Web 2.0) has boosted the idea of the Web Of Trust (WOT) [20, 39, 50]. The WOT represents the trust between users, modeled using an online network. There are specific approaches that use a custom trust network to recommend items. One example is FilmTrust [20], which exploits a custom network of trust between users regarding movie preferences. However, these specific trust networks are quite difficult to generate because they require explicit feedback from users, and this can generate rejection.

All these works take into account some of the different factors involved in our proposal: personality or trust. However, we have not found any work that integrates and evaluates these two factors in group recommendation processes. Therefore, we consider that our approach improves these works by making a more exact representation of how group argumentations take place in real life.

7 Conclusions

In this paper we have reviewed existing techniques in group recommender systems and contribute to the state of the art with a method of making group recommendations that includes social elements. The paper describes the ARISE architecture for the development of group recommenders that takes into account social factors like trust, conformity and fairness. The inclusion of these factors leads to a significant improvement in the performance of the recommendations. ARISE is a theoretical organization of the modules required to build such kind of enhanced recommenders, which has been instantiated in the *HappyMovie* application. *HappyMovie* is a real application that serves as a proof of concept and was developed to exploit Facebook in order to obtain social information about users.

The main focus in this paper is the study of how to apply the methods proposed in ARISE -exploiting information about the social relationships and behaviour of the users to provide better recommendations- to a wide range of different aggregation functions that help us combine all the information extracted from users in order to build the final group recommendation. We have tested the behaviour of our recommender with all its possible configurations in the movie recommendation domain using two test datasets. The first case study uses real users and the second one uses synthetically generated data to create simulated groups of people.

In both experiments we have used groups of different sizes and personal preferences, and different aggregation functions, where we have proved that by introducing the trust factor and personality awareness we improve the results of the recommendations. We have also studied several features of group composition to measure their impact on the accuracy of the group recommender. Regarding the influence of group size, the conclusion is that we obtain better results for small groups with *average satisfaction* and for big groups with *least misery* or *average without misery*.

These conclusions lead us to propose as future work an adaptive recommender that applies different aggregation functions depending on the group. This means that depending on the configuration of the group we will choose a different approach to compute the final group recommendation.

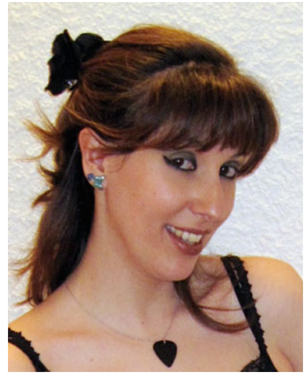
We are also working on the possibility of extending the group recommendation application for movies to other domains like music, recipes, trips and/or restaurants. This way we can validate our group recommendation method. To do so we would only have to improve the *Web Crawling Module* so that it searches the web for specific information on each domain and builds case bases. We will have to modify the *Individual Preferences Module* as well in order to ask users to rate items in the specific domain.

Acknowledgements Supported by Spanish Ministry of Science & Education (TIN2009-13692-C03-03) and Madrid Education Council and UCM (Group 910494) and Spanish Ministry of Economy and Competitiveness under grant IPT-2011-1890-430000.

References

1. Abramowitz M, Stegun IA (1964) Handbook of mathematical functions with formulas, graphs, and mathematical tables. Dover, New York. Ninth Dover printing, tenth GPO printing edition
2. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, pp 734–749
3. Amer-Yahia S, Roy SB, Chawlat A, Das G, Yu C (2009) Group recommendation: semantics and efficiency. Proc VLDB Endow 2(1):754–765
4. Ardissono L, Goy A, Petrone G, Segnan M, Torasso P (2003) Intrigue: personalized recommendation of tourist attractions for desktop and handset devices. Appl Artif Intell 17(8):687–714. http://dx.doi.org/10.1080/713827254
5. Baltrunas L, Makcinskas T, Ricci F (2010) Group recommendations with rank aggregation and collaborative filtering. In: Proceedings of the fourth ACM conference on recommender systems, RecSys'10. ACM, New York, pp 119–126. doi:10.1145/1864708.1864733
6. Barsade SG (2002) The ripple effect: emotional contagion and its influence on group behavior. Adm Sci Q 47(4):644–675
7. Bazire M, Brézillon P (2005) Understanding context before using it. In: Context, pp 29–40
8. Berkovsky S, Freyne J (2010) Group-based recipe recommendations: analysis of data aggregation strategies. In: Proceedings of the fourth ACM conference on recommender systems, RecSys'10. ACM, New York, pp 111–118. doi:10.1145/1864708.1864732
9. Bischoff K (2010) Exploiting social ties for search and recommendation in online social networks—challenges and chances. In: Grundlagen von Datenbanken
10. Bobadilla J, Serradilla F, Hernando A (2009) Collaborative filtering adapted to recommender systems of e-learning. Knowl-Based Syst 22(4):261–265
11. Chen YL, Cheng LC, Chuang CN (2008) A group recommendation system with consideration of interactions among group members. Expert Syst Appl 34(3):2082–2090. doi:10.1016/j.eswa.2007.02.008
12. Christakis NA, Fowler JH (2011) Social contagion theory: Examining dynamic social networks and human behavior. CoRR
13. Crossen A, Budzik J, Hammond KJ (2002) Flytrap: intelligent group music recommendation. In: IUI'02: proceedings of the 7th international conference on intelligent user interfaces. ACM, New York, pp 184–185. doi:10.1145/502716.502748
14. Davoodi E, Kianmehr K, Afsharchi M (2013) A semantic social network-based expert recommender system. Appl Intell 39(1):1–13
15. Díaz-Agudo B, González-Calero PA, Recio-García JA, Sánchez-Ruiz-Granados AA (2007) Building cbr systems with jcolibri. Sci Comput Program 69(1–3):68–75
16. Ekstrand MD, Riedl J, Konstan JA (2011) Collaborative filtering recommender systems. Found Trends Hum-Comput Interact 4(2):175–243
17. Erdélyi G, Nowak M, Rothe J (2009) Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. Math Log Q 55(4):425–443
18. Gilbert E, Karahalios K (2009) Predicting tie strength with social media. In: CHI'09: proceedings of the 27th international conference on human factors in computing systems. ACM, New York, pp 211–220. doi:10.1145/1518701.1518736
19. Golbeck J (2006) Combining provenance with trust in social networks for semantic web content filtering. In: Moreau L, Foster IT (eds) Provenance and annotation of data, international provenance and annotation workshop, IPAW 2006, Chicago, IL, May 3–5, 2006. Lecture notes in computer science, vol 4145. Springer, Berlin, pp 101–108. Revised Selected Papers
20. Golbeck J (2006) Generating predictive movie recommendations from trust in social networks. In: iTrust: 4th international conference on trust management, pp 93–104
21. Hatfield E, Cacioppo J, Rapson R (1994) Emotional contagion. Studies in emotion and social interaction. Cambridge University Press, Cambridge
22. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22(1):5–53. doi:10.1145/963770.963772
23. Jameson A (2004) More than the sum of its members: challenges for group recommender systems. In: AVI'04: proceedings of the working conference on advanced visual interfaces. ACM, New York, pp 48–54. doi:10.1145/989863.989869
24. Jameson A, Smyth B (2007) Recommendation to groups. In: Brusilovsky P, Kobsa A, Nejdl W (eds) The adaptive web, methods and strategies of web personalization. Lecture notes in computer science, vol 4321. Springer, Berlin, pp 596–627
25. Kim JK, Kim HK, Oh HY, Ryu YU (2010) A group recommendation system for online communities. Int J Inf Manag 30(3):212–219. http://www.sciencedirect.com/science/article/B6VB4-4XHJX9H-4/2/29b4c6d7fb829946dc84b9db4bcab5a2. doi:10.1016/j.ijinfomgt.2009.09.006
26. Klamler C (2005) The copeland rule and condorcet's principle. Econ Theory 25(3):745–749
27. Koohborfardhaghighi S, Kim J (2013) Using structural information for distributed recommendation in a social network. Appl Intell 38(2):255–266
28. Levin DZ, Cross R, Abrams LC (2004) The strength of weak ties you can trust: the mediating role of trust in effective knowledge transfer. Manag Sci 50:1477–1490
29. Lieberman H, Dyke NWV, Vivacqua AS (1999) Let's browse: a collaborative web browsing agent. In: IUI, pp 65–68
30. Liu NH (2013) Comparison of content-based music recommendation using different distance estimation methods. Appl Intell 38(2):160–174
31. Lops P, de Gemmis M, Semeraro G (2011) Content-based recommender systems: state of the art and trends. In: Ricci F, Rokach L, Shapira B, Kantor PB (eds) Recommender systems handbook. Springer, Berlin, pp 73–105

32. Masthoff J (2004) Group modeling: selecting a sequence of television items to suit a group of viewers. User Model User-Adapt Interact 14(1):37–85
33. Masthoff J, Gatt A (2006) In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. User Model User-Adapt Interact 16(3–4):281–319
34. McCarthy JF (2002) Pocket restaurant finder: a situated recommender systems for groups. In: Proceeding of workshop on mobile ad-hoc communication at the 2002 ACM conference on human factors in computer systems
35. McCarthy JF, Anagnost TD (1998) MusicFX: an arbiter of group preferences for computer supported collaborative workouts. In: CSCW'98: proceedings of the 1998 ACM conference on computer supported cooperative work. ACM, New York, pp 363–372. doi: 10.1145/289444.289511
36. McCarthy K, McGinty L, Smyth B, Salamó M (2006) The needs of the many: a case-based group recommender system. Adv Case-Based Reasoning 4106:196–210. doi:10.1007/11805816_16
37. Mika P (2011) Making things findable: semantics for web search and online media. In: WIMS, p 3
38. O'Connor M, Cosley D, Konstan JA, Riedl J (2001) Polylens: a recommender system for groups of users. In: ECSCW'01: proceedings of the seventh conference on European conference on computer supported cooperative work. Kluwer Academic, Norwell, pp 199–218
39. O'Donovan J, Smyth B (2005) Trust in recommender systems. In: IUI'05: proceedings of the 10th international conference on intelligent user interfaces. ACM, New York, pp 167–174. doi:10.1145/1040830.1040870
40. Pazzani MJ, Billsus D (2007) Content-based recommendation systems. In: The adaptive web, pp 325–341
41. Quijano-Sánchez L, Recio-García JA, Díaz-Agudo B (2009) Social based recommendations to groups. In: Procs of the 14th UK workshop on case-based reasoning. CMS Press, London, pp 46–57
42. Quijano-Sánchez L, Recio-García JA, Díaz-Agudo B (2010) Personality and social trust in group recommendations. In: Procs of the 22th international conference on tools with artificial intelligence, ICTAI'10. IEEE Comput Soc, Los Alamitos, pp 121–126
43. Quijano-Sánchez L, Recio-García JA, Díaz-Agudo B (2011) Happymovie: a facebook application for recommending movies to groups. In: Procs of the 23th international conference on tools with artificial intelligence, ICTAI'11
44. Recio-García JA, Jimenez-Diaz G, Sánchez-Ruiz AA, Díaz-Agudo B (2009) Personality aware recommendations to groups. In: Procs of the 2009 ACM conference on recommender systems. ACM, New York, pp 325–328
45. Ricci F, Rokach L, Shapira B, Kantor PB (eds) (2011) Recommender systems handbook. Springer, Berlin
46. Saari DG (2000) Mathematical structure of voting paradoxes. Econ Theory 15(1):1–53
47. Sinha RR, Swearingen K (2001) Comparing recommendations made by online systems and friends. In: DELOS workshop: personalisation and recommender systems in digital libraries
48. Thomas K, Kilmann R (1974) In: Thomas-Kilmann conflict mode instrument. Tuxedo, New York
49. Tomlinson S (2006) Comparing the robustness of expansion techniques and retrieval measures. In: CLEF, pp 129–136
50. Victor P, Cornelis C, Cock MD, Teredesai A (2008) Key figure impact in trust-enhanced recommender systems. AI Commun 21(2–3):127–143
51. Yu Z, Zhou X, Hao Y, Gu J (2006) TV program recommendation for multiple viewers based on user profile merging. User Model User-Adapt Interact 16(1):63–82. doi:10.1007/s11257-006-9005-6



Lara Quijano-Sanchez is a Ph.D. student at the Complutense Univeristy of Madrid. Her areas of research are Recommender Systems, Social Networks, Social Factors, Decision Support Systems, Applied Intelligent Systems, etc. She is presently working in the analysis and use of group recommender systems where she has developed several algorithms to reflect accurately argumentations between users in decision making processes. Her publications have appeared in various Journals and International Conferences.

Juan A. Recio-Garcia is Assistant Teacher at the Artificial Intelligence Department at the Computer Science Faculty at the Complutense University of Madrid, where he obtained a Ph.D. in Computer Science in 2008. His research has focused on the confluence of Software Engineering and Case-Based Reasoning, developing the COLIBRI framework for building CBR systems. Currently he is also working in the areas of Recommender Systems and Semantic Web. He has several publications at International Conferences and Journals.

Belen Diaz-Agudo is a professor at the Artificial Intelligence Department at the Computer Science Faculty at the Complutense University of Madrid. Her research interests include Case Based Reasoning, Semantic Web, Recommender Systems and their relationships to applied intelligent systems. She has more than 50 publications at International Conferences and Journals, and has participated in the organization of AI conferences.

Chapter 23

Development of a Group Recommender Application in a Social Network

23.1 Citation

Quijano-Sánchez, L., Recio-García, J. A., Díaz-Agudo, B., Development of a Group Recommender Application in a Social Network, Special Issue on Knowledge-Bases for Cognitive Infocommmunications, Knowledge-Based Systems, pages: 72-85. Elsevier, 2014. ISSN: 0950-7051. JCR (2013): 3.058. (Computer Science, Artificial Intelligence, 15 out of 121, Q1).

23.2 Contributions covered by this paper

In this paper we present a complete explanation about how our Facebook social group recommender application, *HappyMovie*, works. We have included all the implementation and designed details along with the explanations of why those decisions have been taken. Besides we have evaluated *HappyMovie* with a group of real users who have reflected their acceptance and happiness towards the recommendations presented.

Knowledge-Based Systems 71 (2014) 72–85

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Development of a group recommender application in a Social Network

Lara Quijano-Sánchez *, Belén Díaz-Agudo, Juan A. Recio-García

Department of Software Engineering and Artificial Intelligence, Facultad de Informatica, Universidad Complutense de Madrid, C/Prof. Jose Garcia Santesmases s/n, 28040 Madrid, Spain

ARTICLE INFO

Article history:
Received 17 January 2014
Received in revised form 7 May 2014
Accepted 7 May 2014
Available online 15 May 2014

Keywords:
Social Networks
Social knowledge
Group recommender systems
Social applications

ABSTRACT

In today's society, recommendations are becoming increasingly important. With the advent of the Social Web and the growing popularity of Social Networks, where users explicitly provide personal information and interact with others and the system, it is becoming clear that the key for the success of recommendations is to develop new strategies which focus on social recommendations leveraged by these new sources of knowledge. In our work, we focus on group recommender systems. These systems traditionally suffer from a number of shortcomings that hamper their effectiveness. In this paper we continue our research, that focuses on improving the overall quality of group recommendations through the addition of social knowledge to existing recommendation strategies. To do so, we use the information stored in Social Networks to elicit social factors following two approaches: the cognitive modeling approach, that studies how people's way of thinking predisposes their actions; and the social approach, that studies how people's relationships predispose their actions. We show the value of using models of social cognition extracted from Social Networks in group recommender systems through the instantiation of our model into a real-life Facebook movie recommender application.

1. Introduction

Recommender systems are knowledge-based systems which support human decision-making. In an era of overwhelming choice, they help us decide which products, services and information to consume. The focus of attention in recommender systems' research and development has been on making recommendations to individual consumers (e.g. see [1] for an overview). These places focus on the easier case, but ignore the fact that it is as common, if not more common, for us to consume items in groups such as couples, families and parties of friends. The choice of a date movie, a family holiday destination, or a restaurant for a celebration meal all require the balancing of the preferences of multiple consumers.

This kind of recommendations are provided by group recommender systems [2]. These systems commonly aggregate real or predicted ratings for group members [2–5]. The aggregation functions typically used are inspired by the social welfare functions developed by the Social Choice Theory research [6]. First, for each group member, an individual recommender system predicts a set of ratings for the candidate items. Next, the group recommender aggregates the ratings: for each candidate item, it might take the average of group members' ratings, or the minimum, or the maximum, for example. Finally, it recommends to the group the items with the highest aggregated ratings.

However, this widely accepted approach for group recommendations ignores the social factors that influence real group decision-making. In real-life group decision-making a variety of social-cognitive processes underlie the choices that people make [7]. For example, it has been proven that social media highly influences peoples' decisions, relationships, and education. Several researchers study the impact of social media in our lives [8]. These studies evaluate the social context, which refers to the immediate physical and social setting in which people live. It includes the culture that the individual was educated or lives in, and the people and institutions with whom they interact. Besides, circumstantial life events, influences, and surroundings can further change our social abilities [9]. In our research, we study how to model these social-cognitive processes in human decision-making processes [10,11]. Specifically, we have designed a method that is aware of the different personalities and social ties that group members present. These techniques and their associated algorithms have been compiled into a generic architecture named ARISE (Architecture for Recommendations Including Social Elements) that can be instantiated into any kind of social recommender system.[1] The common

* Corresponding author. Tel.: +34 91 394 75 99.
E-mail addresses: lara.quijano@fdi.ucm.es (L. Quijano-Sánchez), belend@sip.ucm.es (B. Díaz-Agudo), jareciog@fdi.ucm.es (J.A. Recio-García).

[1] Note that the verification of the generalisability of our approach is out of the scope of this paper. However, it has already been proven valid in two different domains: movies & clothing, as stated in [12].

http://dx.doi.org/10.1016/j.knosys.2014.05.013
0950-7051/

and key factor in all the different types of recommenders that can be built in all sort of domains using this generic architecture is the inclusion of social elements. These social elements define each person as a potentially influenced component of a social community or group determined by the environment, in most cases Social Networks, s/he belongs to. In our *social* method, we have simulated people's behavior based on the idea that the relationship between individuals and their networks of people directly influence their lives [8]. Besides, we have provided a software development methodology to ease the instantiation of the ARISE architecture into concrete applications. This methodology is based on templates [12] that formalize the functional behavior of social recommender systems and facilitate its configuration and deployment.

To illustrate and validate the capabilities of our *social recommendation* approach we have instantiated our model into a real-life recommender system: *HappyMovie*,[2] which is a particular application of our generic architecture ARISE for the movies recommendation domain in the Social Network Facebook. *HappyMovie* serves us as a use case and experimental environment where we are able to evaluate our ARISE architecture and our *Social Group Recommendation Method* with real data.

The main contribution of this paper is to measure the benefits of having a social group recommender application embedded in one of the biggest Social Networks in the world[3] following two goals:

(1) Testing in terms of users' acceptance, the benefits of using our *Social Recommendation* approach against a state-of-the-art approach that does not use social knowledge.[4] We provide a usability evaluation of the application *HappyMovie*. To do so, we have performed a functionality test where 60 users have evaluated different aspects of *HappyMovie*, proving that our application and its associated *Social Group Recommendation Method* indeed offer good and eligible group recommendations. With this evaluation we also want to measure users' response towards having the application in Facebook, if there is any resentment towards the different knowledge elicitation processes, and more importantly their satisfaction levels with the recommendations provided. This is an important evaluation as we want to test if *HappyMovie* is a viable and suitable platform to continue our research with.
(2) Testing if users will indeed use the application and therefore it is worth making it public. This evaluation will allow us to determine if *HappyMovie* is viable not only for giving good recommendations but also as a research tool that allows us to extract group related knowledge.[5]

Another contribution toward group recommender applications is a detailed explanation of our *Social Group Recommendation Method* in a social application like *HappyMovie*. This survey provides a self-contained full description to support the complete reproducibly of our system. In [13] we presented a functional vision of *HappyMovie*. This previous work introduced an initial version of *HappyMovie* where the focus was on presenting an experiment on the viability of making the application more easy going through an interactive game that measures users' personality instead of the long questionnaire that we previously used in [10,11]. In this paper we have refined, tested and justified the ideas and decisions made in our previous work. With it, we want to provide future group recommender developers a reference on how to build and exploit further *social group recommender* systems. The novelty presented in the paper lies in the explanation of how we now elicit and apply each social factor in the recommendation process and in the explanation of why these factors improve the overall quality of the recommendation.

The remainder of this paper is structured as follows: In the next section we introduce some of the state-of-the-art knowledge regarding recommender systems and study other recommendation approaches whose main goal is also the improvement in the performance of recommenders through the usage of social information. Next, Section 3 introduces the details of our Facebook application *HappyMovie*. The evaluation of *HappyMovie* is presented in Section 4. Finally Section 5 concludes the paper.

2. Literature review

There are a lot of domains where group recommendation techniques can be applied, for example in the music domain, MusicFX [14] and FlyTrap [15]. Polylens [16] in movies domain. Or, regarding recommendations of restaurants for groups, Pocket Restaurant Finder [17]. What all these recommenders have in common is that they take into account personal preferences obtained from their users; however, they consider each user to be equal to the others. The recommendation is not influenced by their personality or the way each user behaves in a group when joining a decision-making process. Some works, do take into account not only the preferences of every member but also the interaction among them; Travel Forum Decision [18] or the work performed by Masthoff and Gatt [6] are examples of this.

Social Networks have been one of the most important topics in the last few years, with nets like Facebook, Twitter and MySpace, among others. The use of Social Networks and trust when building recommender systems is not new [19]. Current research has pointed out that people tend to rely more on recommendations from people they trust (friends) than on recommendations based on anonymous ratings [20]. There is a huge body of work about the generation of trust models. There are specific approaches that use a custom trust network to recommend items. One example is FilmTrust [21], which exploits a custom network of trust among users according to movie preferences. However, these specific trust networks are quite difficult to generate because they require explicit feedback from users, and these can generate rejection.

The general opinion is that there is a need to adapt the recommendation process to group composition [2,22]. This is backed up by some recent works that have focused their studies on analyzing the effectiveness of group recommendations according to different aspects, such as group size and inner group similarity [3].

Avoiding repeated recommendations is also an important matter that some works give special importance to. For example [22] in the recommendation of TV programs domain or FlyTrap [15] where previous selections are also taken into consideration. Besides, some recommendations tend to be repeatedly detrimental to the same group members. Another system that takes previous selections into account is PoolCasting [23]. In our approach, we have given special treatment to the evaluation of previous recommendations in order to avoid this tendency and, similarly to Masthoff and Gatt's work [6], we have gone one step forward considering also users' satisfaction with past recommendations.

[2] http://gaia.fdi.ucm.es/research/happymovie.

[3] Facebook passes 1.19 billion monthly active users.

[4] Note that in [10,11] we proved in simulated environments that using social factors, i.e. personality and trust, improved the recommendation accuracy. However, now that we have embedded our approach directly in a Social Network, the elicitation of these factors has been adapted to this situation and is therefore different. Hence, new experiments need to be carried out.

[5] As a future goal we would like to obtain a big dataset with all the knowledge that this application is capable of acquiring, in terms of trust levels, users' personality, users' preferences and final group decisions. With this dataset we could be able to provide to the recommenders community a public group dataset (we are not aware of any similar public dataset), run Big Data experiments or study group similarity composition and behavior among others.

Regarding the extraction of information from Social Networks, in the last few years there has been a huge line of research that uses social information to improve recommender systems. Examples of this are [24] that use the social information stored in the music Social Network *last.fm*, and capture explicitly expressed bonds of friendship as well as social tags to improve recommendations' accuracy. Or [25], that use social tags to recommend the most suitable multimedia contents for users.

While recommender systems have been extensively researched since the mid-1990s [14,26], the study of social-based recommender systems is a new area [27]. One of the key factors that social-based recommendations use is the study of the multiple dimensions within a user's Social Network, including social relationship strength, interests, and user similarity. In [28] the authors seek to develop novel group recommender systems that leverage these dimensions.

Users in today's online Social Networks often post a profile, consisting of attributes such as geographic location, interests, etc. Such profile information is used on recommender systems as a basis for grouping users, for sharing content, and for suggesting users who may benefit from interaction. However, in practice, not all users provide these attributes and several researchers have focused on handling imprecise and incomplete information [29]. In [30] they use the information stored in online Social Networks to infer the attributes missing in some users' profiles.

As we can see, social relations provide an independent source for recommendation; various approaches are proposed to build social recommender systems such as trust relationships [31], trust propagation [32], or directly trust user based recommenders [33]. Besides, there is recent work reporting significant recommendation performance improvement for social recommender systems [21,34–39]. On the other hand, there are also unsuccessful attempts at applying social recommendation [40,41].

Summing up, social recommendation is still in the early stages of development, and there are many challenging issues needing further investigation. Following this reasoning we consider the necessity to discuss and propose new research directions that can improve social recommendation capabilities and make social recommendation applicable to an even broader range of applications. Besides, we have not found any work (aside from our own) that integrates an automatic elicitation of the knowledge stored in users' profiles to obtain a trust value that is later on used along with a personality value and a memory of past recommendations to improve the results of group recommender systems.

Next, we introduce our *Social Group Recommendation Method* and how we have modeled it in our *Social group recommender* application *HappyMovie*.

3. *HappyMovie*

Our group recommendation method is based on three major components: personality, social trust and memory of past recommendations. The integration of these social factors enables a realistic simulation of decision-making processes followed by groups of people when deciding a joint activity. With *HappyMovie* we try to mitigate certain limitations in existing group recommender systems, like obtaining users profiles or offering trading schemes in order to reach a final agreement [42]. It serves us as a use case and experimental environment where we can evaluate our *Social Group Recommendation Method* with real products. This way we can validate our previous results [10,11], where we concluded that personality and social trust factors indeed improved the performance of group recommendations. In addition, thanks to the inclusion of the system in a Social Network we can now continue our investigation with further experiments that include the study of users' response over time towards the recommendations given, a detailed analysis of their opinion towards *HappyMovie*'s recommendations and obtaining a large set of data which will enable us to better study group similarities and recurring behavior (this last experiment is out of the scope of this paper but would be now possible thanks to the social factors' automatic elicitation process now granted thanks to having the system embedded in a Social Network).

Our goal with *HappyMovie* is to evolve and integrate group recommender systems into the Social Web – concretely Facebook – where personal relations can be analyzed and exploited to enhance the process of making recommendations to groups. Within this environment, we are able to infer much of the information needed to perform *Social Group Recommendations* directly from Social Networks [21,43]. Previously, the acquisition of such social data had to be performed by means of tedious questionnaires. The integration into Social Networks eases this process and provides a lot of valuable feedback to evaluate and improve our proposal. In our social method, we have reproduced people's behavior based on the idea that the relationship between individuals and their networks of people directly influence their lives [8].

The architecture of *HappyMovie* is represented in Fig. 1. The application is organized in four different groups of modules: Social Knowledge Elicitation, Long Term Knowledge Management, Individual Recommendation and Group Recommendation. The next sections detail and analyse each of them.

3.1. Social Knowledge elicitation modules

When people face situations where their interests or preferences are incompatible with others *conflict situations* arise. Here conflict is understood as a difference that prevents agreement. More concretely, in group interactions it is defined as a competitive or opposing action of incompatibles: antagonistic state or action (as of divergent ideas, interests, or persons) [44]. Different people have different expectations and behavior in conflict situations [6]. Our research to improve group recommendation systems studies the different behaviors that people have in conflict situations according to their personality and inter-personal trust. Next we describe the modules in *HappyMovie*'s architecture in charge of eliciting users' social knowledge.

3.1.1. Personality modeling

Modeling human cognitive reactions through computer interfaces is not new [45,46]. As our goal is to estimate users' behavior in decision-making processes (users' personality value), *HappyMovie*'s users perform an adaptation of the Thomas–Kilmann Conflict Mode Instrument (TKI) test [47], which is a leading instrument used by individuals and businesses for identifying their ability to handle conflicts in decision-making processes. This test is commonly used in the human–machine interaction area, due to its efficiency, easy evaluation and easy usage for people not related to the psychology area. In comparison with other similar tests, such as *Ego Gram* [48], that measures personality according to three "Ego States" (Parent, Child, Adult), or *Pen Model* [49], that measures Psychoticism, Extraversion and Neuroticism (PEN), we believe that the TKI test is the most suitable choice as its main focus is on measuring people's reactions in conflict situations whereas other tests' focuses (even though they may perform a wider personality study) are not as specific for our purpose as TKI is. Besides, TKI provides a tangible and measurable value, easy to interpret and most importantly short and easy to answer. We think that this last characteristic is a key element in the success and acceptance of *HappyMovie* as users may not be willing to answer long personality tests in order to obtain a movie recommendation. For example, the NEO-PI-R [50] is a 240-item questionnaire (designed to operationalize the

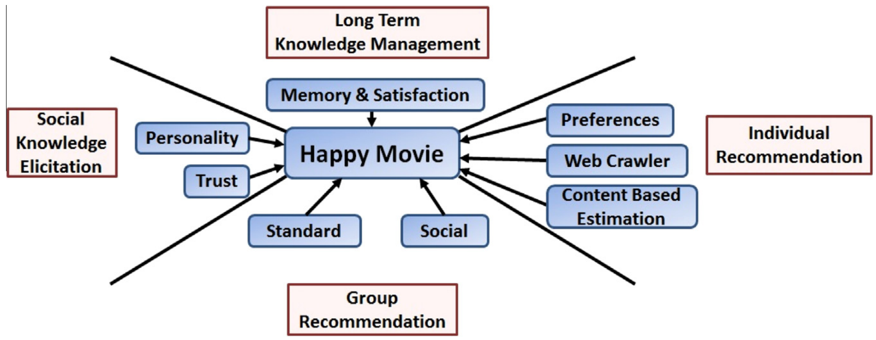


Fig. 1. *HappyMovie*'s modules.

five-factor model of personality [51]), which for our recommendation purposes is too long. The TKI test consists of 30 different situations with two possible answers. Depending on the answers, a score is assigned for 5 existing personality modes (see Fig. 2) organized according to two dimensions: *assertiveness* and *cooperativeness*.

Once we finished our experiments in simulated environments [10,11], users were asked to give their opinions about the test which they described as tedious and long. To make the application more easy going we studied the possibility of using a movie metaphor as an alternative personality test. Consequently we developed an alternative metaphor that lightens this activity. This interactive metaphor consists of displaying two movies characters with opposite personalities for each of the 5 existing conflict-handling modes. One character represents the essential characteristics of the mode, while the other one represents all the opposite ones. Users have to move an arrow showing their degree of similarity with the personality characteristics of the characters being presented (these personality characteristics along with examples of typical behavior patterns are presented under each character's image). In [13] we concluded that it is possible to replace the TKI personality test with the movie metaphor test because it provides an statistically confirmed accurate estimation of the personality mode. The results that the TKI test provided for the five different personality modes in comparison with the values that the movie metaphor test gathered had a Mean Absolute Error (MAE) [52] of 0.12. Hence, we believe that it is worth sacrificing a little accuracy in the test results (as they are not for psychology testing purposes) in exchange of enhancing significantly the usability and interest for the application. A screenshot of *HappyMovie*'s personality test is presented in Fig. 3.

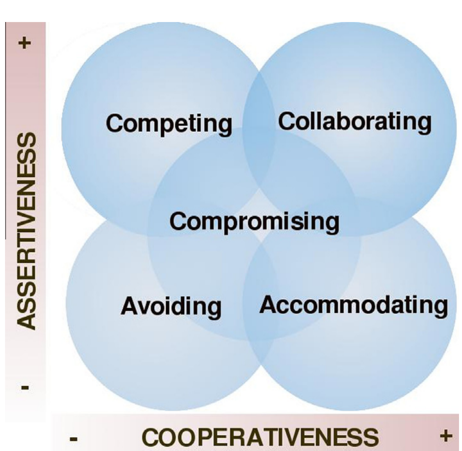


Fig. 2. TKI personality modes.

In this paper, we fully explain how to calculate the personality value p_u, which fits within a range of (0,1], 0 being the reflection of a very cooperative person and 1 the reflection of a very selfish one.

Firstly, users select the character they feel more identified with by moving the arrow. This process is repeated for the 5 personality modes with different representative and opposite characters. The arrow has positions (marked 1 to 10) that represent the percentage that users have in each category. This is done to measure users' resemblance degree on each personality mode.

The next step is to reduce these 5 different types of personality into TKI's two dimensions: *assertiveness* and *cooperativeness*. If one of the 5 scores, that have been now estimated, is below or above the 25 or 75 percentile according to the population, then users are classified as having a low or high personality mode in that category [53]. This way the test indicates if users have high or low degree of resemblance with each one of the existing modes. Following the schema shown in Fig. 2, if users have high competing and collaborative mode they are assigned a high assertiveness value. High avoiding and accommodating personality modes are considered as low assertiveness. Following the second dimension, high cooperativeness value is given to users if they have high accommodating and collaborating mode. The assertiveness and cooperativeness values are a weighted sum of the five personality modes. These weights are the coefficients shown in Fig. 4.[6] For example, users with a high percentile score in Competing mode add a weight of 0.4 in Assertiveness and −0.2 in Cooperativeness. Medium percentile scores are not included in the personality estimation.

For the sake of clarity we here present an example of the calculation of these values: let's say that a user u has a percentage of the personality modes as shown in Fig. 5, then, assertiveness and cooperativeness values are calculated as in Eqs. (1) and (2) respectively.

$$Assertiveness(u) = -0.2 - 0.1 - 0.2 - 0.4 + 0 = -0.9 \tag{1}$$

$$Cooperativeness(u) = 0 - 0.1 + 0.4 - 0.4 + 0 = -0.1 \tag{2}$$

Once the 5 personality modes are reduced to the *assertiveness* and *cooperativeness* dimensions, the personality value (p_u) is computed as the difference between both dimensions. P_u represents user u's predominant behavior according to her/his TKI evaluation, i.e., how assertive or cooperative s/he is. This is computed using the following equation:

$$p_u = \frac{1}{2}(1 + Assertiveness(u) - Cooperativeness(u)) \tag{3}$$

[6] Note that although personality computation is performed equally in [54,11], weights are slightly different as they now use TKI's updated normative sample [53].

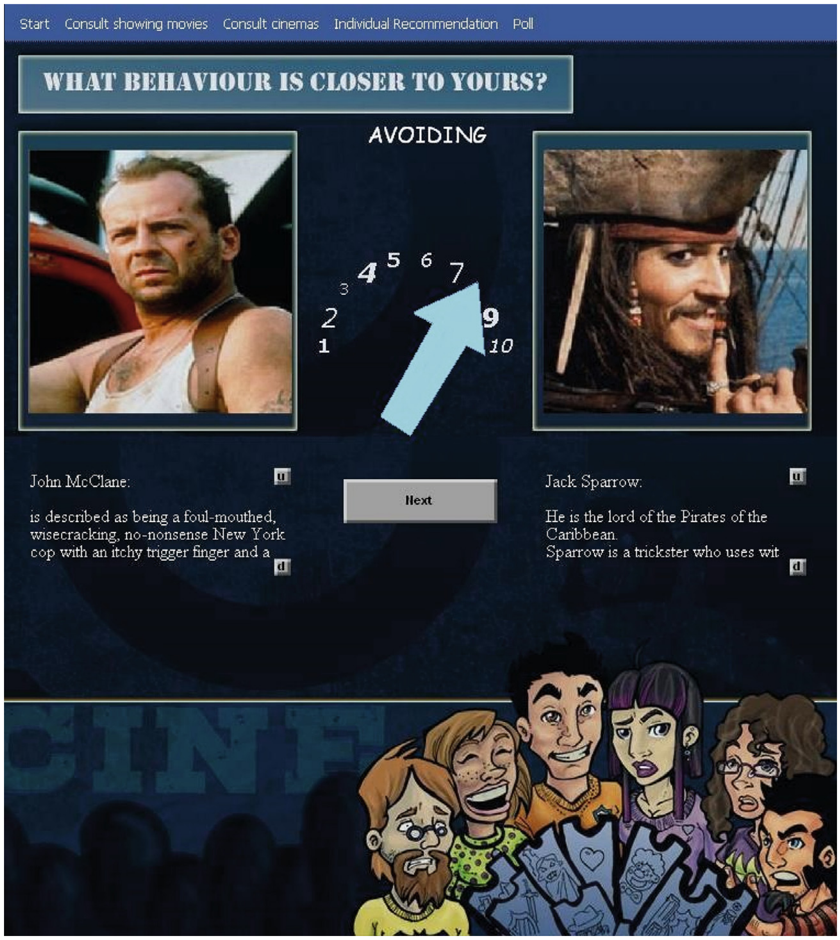


Fig. 3. *HappyMovie*'s personality test.

TKI Mode	Assertiveness		Cooperativeness	
	High	Low	High	Low
Competing	0.4	-0.2	-0.2	0
Collaborating	0.4	-0.1	0.2	-0.1
Accommodating	-0.2	0	0.4	-0.1
Avoiding	-0.4	0.2	-0.4	0.2
Compromising	0	0	0	0

Fig. 4. Weights used to obtain the TKI modes.

TKI Mode	Percentil	Classification
Competing	16.7%	Low
Collaborating	24.0%	Low
Accommodating	76.0%	High
Avoiding	83.3%	High
Compromising	58.3%	Not Relevant

Fig. 5. Example of the calculation of the TKI modes.

Next we explain how we compute the other social factor involved in our method, trust.[7]

3.1.2. Trust estimation

This module obtains the inter-personal trust or social tie between users. This factor can be estimated following different approaches, being most of them manual [19], task that users resented and found very tedious. Hence, we propose its elicitation from Social Networks. In this section we detail how the computation of the trust factor can be now automatically computed thanks to embedding the group recommender system in a Social Network application.[8] The process consists of calculating the inter-personal trust by analysing users' profiles and interactions in the Social Network. Users in Facebook post a huge amount of personal information that can be extracted to compute the trust with other users: likes and interests, personal preferences, pictures, games, etc.

The use of trust and other social knowledge obtained from Social Networks in the development of recommender systems is not new [21,55]. We have reviewed several existing works [56,19] that identify the factors to be analysed. In order to move from theory to practice it is important to note that these factors are not easy to quantify and are limited by the Social Network's API extraction power.

Previous works have reported that trust and tie strength are conceptually different but that there is a correlation between them [57]. [58] defines tie strength as a (probably linear) combination of four factors: the amount of time, the emotional intensity, the intimacy (mutual confiding), and the reciprocal services which

[7] Note that this trust factor measures the tie strength in people's relationships.

[8] Contrarily, for the experiments carried out in [10,11] the trust factor had to be differently computed and was done by friending on Facebook a small sample of users and extracting the information needed manually. Then we used 10 variables to estimate the trust factor, whereas as we will next see in here we only use 4. This change in the trust computation is due to the computed relevance of those factors (the weight in the trust computation of some of them was almost insignificant in [11]) and the extraction power that Facebook gives us (status was impossible to automatically extract reliably).

characterize the tie. The literature reviewed identifies these four factors as some of the major dimensions of predictive variables. With these dimensions as a guide, [56] identified 74 Facebook variables as potential predictors of tie strength. They presented a diagram showing percentages that indicate the predictive power of their top seven tie strength dimensions[9] and also the top three predictive variables for each dimension. From the predictive variables that [56] provided, we selected the ones that were more representative of each major dimension and which could also be extracted from users' Facebook profiles (as we have said before, we are limited to the elicitation power that Facebook grants us). The percentages that [56] presented for [58]'s four dimensions were:

1. **Intimacy** (32.8%), predictive variables: appearances together in photo, participant's appearances in photo, distance between hometowns, etc. In our estimation we use overlapping photo tags.
2. **Intensity** (19.7%), predictive variables: wall words exchanged, inbox messages exchanged, participant-initiated wall posts, etc. In our estimation we use all sorts of messages exchanged.
3. **Duration** (16.5%), referred to when users first met. We did not have access to the date when two people became friends. Hence, we were forced to compute it as the structural variable, number of mutual friends, where the more friends they have in common the longer they have interacted as more different groups of friends overlap.
4. **Reciprocal services** (21.7%[10]), top predictive variables: educational difference, occupational difference, links exchanged by wall post, applications in common, etc. In our estimation we use overlapping profile information.

Summing up, in *HappyMovie*, the trust factor that estimates tie strength between users u and v is computed as follows[11] (note that we understand that $t_{u,v}$ is a directed measure and that $t_{u,v} \neq t_{v,u}$):

$$t_{u,v} = 0.361 \cdot Intimacy_{u,v} + 0.239 \cdot ReciprocalServices_{u,v} + 0.219 \cdot Intensity_{u,v} + 0.181 \cdot Duration_{u,v} \quad (4)$$

where the predictive variables are computed as follows[12]:

$Intimacy_{u,v}$: Represents how much users interact outside the Social Network. To compute it we evaluate the percentage of pictures they appear together in the last year (denoted by $percentage_{tags}$). Note that we compute this percentage as a proportion of the pictures in u's profile that v is tagged on. This is done to contemplate the possibility where u has not posted a lot of pictures but v appears in most of them.

$$Intimacy_{u,v} = \begin{cases} 1.0, & \text{if } percentage_{tags} \text{ is} > 75\% \\ 0.7, & \text{if } percentage_{tags} \text{ is} > 50\% \\ 0.5, & \text{if } percentage_{tags} \text{ is} > 25\% \\ 0.3, & \text{if } percentage_{tags} \text{ is} > 10\% \\ 0.1, & \text{if } percentage_{tags} \text{ is} < 10\% \end{cases}$$

$Intensity_{u,v}$: Represents how much users interact inside the Social Network. To compute it we count the number of interactions in the last year (denoted by *interaction*). We understand that there has been interaction in the Social Network if users have exchanged messages of any kind (private messages, wall messages, etc).

$$Intensity_{u,v} = \begin{cases} 1.0, & \text{if } interaction > 3 \ days\ a\ week \\ 0.7, & \text{if } interaction > 1 \ day\ a\ week \\ 0.5, & \text{if } interaction > 1 \ day\ a\ month \\ 0.3, & \text{if } interaction > 1 \ day\ each\ 3\ months \\ 0.1, & \text{if } interaction < 1 \ day\ each\ 3\ months \\ 0.0, & \text{otherwise} \end{cases}$$

$Duration_{u,v}$: Represents how long they have known each other. We compute it as a structural variable that measures the number of common friends. We understand that the more friends they have in common the longer they have known each other because more different circles of friends overlap.

$$Duration_{u,v} = \begin{cases} 1.0, & \text{if } > 25 \ common\ friends \\ 0.7, & \text{if } > 15 \ common\ friends \\ 0.5, & \text{if } > 10 \ common\ friends \\ 0.3, & \text{if } > 5 \ common\ friends \\ 0.1, & \text{if } < 5 \ common\ friends \end{cases}$$

$ReciprocalServices_{u,v}$: Represents how similar their profiles are, in terms of common interests (music, movies, etc.), common schools, jobs, visited cities, etc. To compute it we evaluate the percentage of common posted information (denoted by $percentage_{info}$). Note that we compute this percentage as the proportion of information (interests, personal information, etc.) in u's profile that also appears in v's profile. This is done to contemplate the possibility where u has not posted a lot of information but v has also posted most of them.

$$ReciprocalServices_{u,v} = \begin{cases} 1.0, & \text{if } percentage_{info}\ is > 75\% \\ 0.7, & \text{if } percentage_{info}\ is > 50\% \\ 0.5, & \text{if } percentage_{info}\ is > 25\% \\ 0.3, & \text{if } percentage_{info}\ is > 10\% \\ 0.1, & \text{if } percentage_{info}\ is < 10\% \end{cases}$$

We have tested our trust estimation with a small group of users that indicated us their real trust factor, and obtained and estimated MAE error of 0.16. Hence, although we cannot conclude this section with a design prescription, we are comfortable enough in presenting it as a useful estimation of the tie strength between users.[13]

The trust calculation is done every time a user joins an event with the rest of users also attending to it. These values are not stored, but repetitively calculated as Facebook profiles keep changing and so does trust between two friends. Note that $t_{u,v} \in (0, 1]$, 0 being the reflection of a not very trusted person and 1 the reflection of a highly trusted one.

3.2. Long term knowledge management modules

HappyMovie stores all the recommendations that have been made for every user and every group. This feature avoids repeating past recommendations and also ensures a certain degree of fairness in the long run. Frequently, a group will expect to use the application several times, thereby getting a bigger sample of recommendations. However, our *Social Group Recommendation Method* tends to always favor the same users (because they have stronger personalities or because they are closer friends with other

[9] Note that [56]'s top four dimensions are the ones that [58] used as definition of tie strength and therefore the ones that we have adopted, as the literature has not resolved this issue, let alone specified how many discrete tie strength levels exist.

[10] This dimension was not represented by [56]'s variables, from their variable definition we understood it as profile distance (13.8%) + services (7.9%)

[11] The weights presented in this equation are a proportion of the percentages of the top 4 variables presented by [56] (as [56]'s top 4 only sum 90.7%).

[12] Note that the thresholds and specific values are different from the ones used in [10,11], as we have experimentally obtained them after analyzing average situations in the users' profiles that took part in our previous experiments.

[13] We are currently performing an experiment with *HappyMovie*'s users where, after obtaining users' real trust value through enquiries, we use a genetic algorithm to adjust the weights of each variable. Our goal is to achieve the minimum MAE when comparing the real trust value given by users to the estimation that *HappyMovie* provides. Unfortunately, these results are not available at the time of writing so we leave them for future work.

members). Therefore, we could end up with a situation where we have some dissatisfied users because we take their opinions less into account for the group's sake. In order to avoid a situation of high deviation in the satisfaction levels of the group, we must take into account users' satisfaction regarding past recommendations. It would be desirable that future recommendations favor dissatisfied users so that all of them reach a proper level of satisfaction.

To address this issue, we propose the use of a memory of past recommendations. This way, if one member accepts a proposal that s/he is not interested in, next time her/his preferences will be prioritized in the recommendation process. This means that her/his opinion will have a higher weight next time. These weights will also be influenced by the different personalities of each group member. For example, a user who dislikes the movie (gives it a low rating) may nevertheless be satisfied with the recommendation, especially if s/he appreciates that it has been necessary to balance conflicting interests. Her/his satisfaction might be all the greater if s/he has a more accommodating (less selfish) personality type, or if the recommendation better matches the tastes of group members with whom s/he has stronger connections through contagion and conformity [6]. This behavior is modeled by immediately compensating users who have been negatively affected and have strong personalities and bearing in mind that users with mild personalities might not mind giving in several times.

The satisfaction value s_u is the level of satisfaction of a user u. A user who is extremely happy with the recommendations will have this satisfaction value close to 1. However, the more dissatisfied with the recommendations s/he is, the more that this value will decrease, reaching down to 0 in the worst case. An important and interesting issue of this approach is the time scope of the memory of users' satisfaction. We can update the s_u value to reflect the satisfaction according to the last immediate group recommendation or take into account previous ones. Therefore, the satisfaction value for an execution t of the recommender may depend on the satisfaction of the user with the items recommended in t but also depends on her/his satisfaction with the previous recommendations $t-1, t-2, \ldots$. Hence, we manage two satisfaction values:

Instant satisfaction (is_u): reflects the immediate user's satisfaction with the last recommendation. This is, her/his conformance with the last item recommended to the group. We ask users to rate the items being recommended to the group in order to obtain the instant satisfaction value.[14]

Global satisfaction: (s_u): measures the average satisfaction of the user through time. It is updated every time a recommendation is made:

$$s_u(t) = (1-\delta) \cdot is_u(t) + \delta \cdot s_u(t-1) \tag{5}$$

In this equation we use the $\delta \in [0..1]$ threshold to adjust the impact of the previous satisfaction when updating that value. Somehow, this threshold measures the degree of forgetfulness about past (in) satisfaction. For example, some people could easily remember that they were not taken into account for the last recommendation when facing a new decision making process to select a similar item. On the other hand, other users would not ever take it into account. The measurement of this threshold belongs to the domain of the social sciences and is out of the scope of this paper. For the experiments presented in Section 4.4, we have configured a δ value of 0.5 to represent a balanced impact of previous satisfaction values.

[14] Note that for the experiments detailed in Section 4.4 and performed to justify the necessity of including *long term recommendations*, we computed is_u as the average of the three ratings that users give as feedback once a recommendation is presented to the group (see Fig. 8).

3.3. Individual recommendation modules

Our group recommendation method is based on preference aggregation approaches [59,60]. These approaches are based on the aggregation of users' individual ratings to obtain an estimated rating for the group. Hence, the basic building block of our group recommender is an individual recommender that computes the estimated preference of users for a given item. Individual recommendations in *HappyMovie* follow a content based approach [61]. This approach, schematized in Fig. 1, uses the descriptions of the products to be recommended (obtained with the *Web Crawler module*), compares them with the descriptions of products rated by the user (obtained with the *Preferences Elicitation module*), and predicts the rating for the aimed products (computed in the *Content Based Estimation module*) by computing the average of the most similar rated products.

3.3.1. Preferences elicitation

In the preferences elicitation test users indicate their taste in movies. The ratings here obtained are used by the individual recommender, that estimates the different movies to be recommended according to users' preferences in actors, genre, etc. For example, if a user has voted with 3 stars a certain movie, as we can see for example in Fig. 6, we could consider that s/he likes that type of movies, so later, the individual recommender will analyse the characteristics of this movie and try to find a similar one. In order to complete the test, users must rate at least 40 movies through an 6-point Likert scale (0 to 5). Users are allowed to run this test on demand to modify or increase their ratings. The more ratings users give the more accurate their personal profile will be, and therefore the individual recommender will perform better. This test returns a set of real ratings $r_{u,i}$ for every user u in group G_a and item i in the test set T_s.

The test always presents the same 70 movies, which have been carefully chosen to cover a wide spectrum of movie tastes. They are the most popular movies of the last 3 years in all the different studied genres.[15] After testing different approaches Fig. 6 shows how it has finally been implemented.

3.3.2. Web Crawler

We have built a Web Crawler that searches the web and retrieves all the movies and movie sessions being displayed in Spain's cinemas. This Web Crawler obtains a full technical datasheet for each of the movies being displayed. Each specific characteristic of the movie is a field that the individual recommender compares. For example, in our particular case study these characteristics are main actors, director and synopsis, between others. The retrieved set of movies, with all their specific information, is the target movie listing T_a containing the items i sent to the individual and group recommenders.

3.3.3. Content based estimation

We have chosen a content-based approach to estimate the rating users would assign to a new movie [61]. An alternative approach is a collaborative filtering approach [62]. However, we have chosen the first option because the movies to be recommended are the most recent movies on cinemas, so there are too new to have user ratings. Hence, we could not use those ratings as collaborative recommenders do. This section produces for every

[15] The first time that we implemented this test [13], we chose well-known classic movies, however they were no good for the recommender as it compares actors and directors, besides genre, etc., from current movies on cinemas. So, if for example we selected *Marilyn Monroe* movies the actor's field would be useless as she is no longer making movies and there would not be any possible comparison between fields.

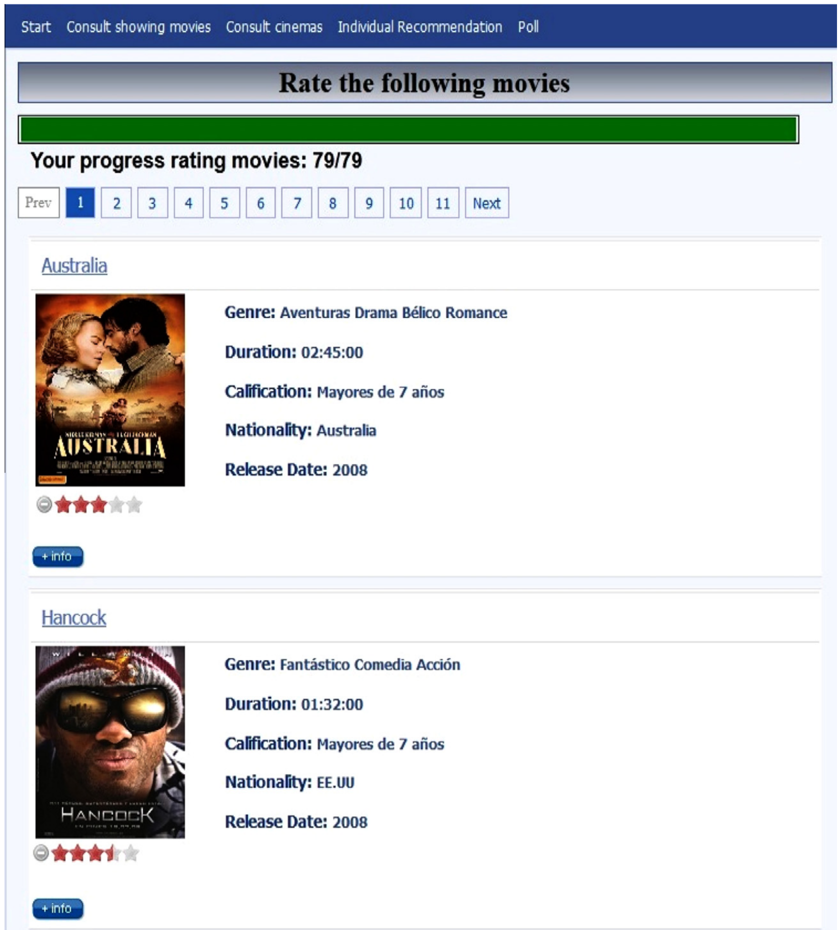


Fig. 6. *HappyMovie*'s preferences test.

user u in the active group G_a a set $\{\hat{r}_{u,i} : u \epsilon G_a, i \epsilon T_a\}$ with the individual predicted ratings for all the target movies.

Our content-based method applies a weighted average of the similarity of the following fields that describe each movie: duration ($w_1 = 0.01$), recommended age ($w_2 = 0.03$), nationality ($w_3 = 0.11$), actors ($w_4 = 0.17$), directors ($w_5 = 0.178$), percentage of action ($w_6 = 0.042$), percentage of animation ($w_7 = 0.045$), percentage of adventures ($w_8 = 0.043$), percentage of comedy ($w_9 = 0.045$), percentage of documental ($w_{10} = 0.02$), percentage of drama ($w_{11} = 0.025$), percentage of fantasy ($w_{12} = 0.034$), percentage of romantic ($w_{13} = 0.044$), percentage of terror ($w_{14} = 0.04$), percentage of thriller ($w_{14} = 0.038$), percentage of science fiction ($w_{15} = 0.046$), and synopsis ($w_{16} = 0.08$).[16]

Note that the weights shown in brackets for each category have been experimentally obtained using our recommendation algorithm combined with a genetic algorithm (GA). We have performed and experiment where 6 people answered the 70 movies of the preferences test.[17] After extracting the results of the test, we used the data to run an experiment using 60% of the data to train the GA and the 40% left to test the results. Our GA manages a population of vectors of weights (w_k). These vectors are combined and mutated in order to maximize the fitness function. Our fitness function is the Mean Average Error (MAE) where we compare the real rating given by our users to the prediction that the recommender system has given.

To compute the percentage that each movie has of each genre we apply Information Extraction (IE) techniques [63] to the textual synopsis. The IE algorithm searches for key terms that are associated to each genre (after text normalization). Finally the textual synopsis is compared by a cosine distance metric.

3.4. Group recommendation module

Suppose there are n users, $U = \{u : 1 \ldots n\}$, let $G_a \subseteq U$ be an active group of users, in our case a group who intend going to see a movie together. The goal is to recommend k items from a set $\{i : 1 \ldots k\}$ of target T_a items. We do this by computing a predicted rating $\hat{r}_{G_a,i}$ for active group G_a and each target item $i \in T_a$, and then recommending the k items in T_a that have the highest predicted ratings.

To obtain a prediction for the group we aggregate the predicted ratings of the members, $\hat{r}_{u,i}$ for each $u \in G_a$ for the various i in T_a. Possible aggregation functions include *least misery* (where the minimum is taken), *most pleasure* (where the maximum is taken) or *average satisfaction* (where the average of the predicted ratings of each group member is taken). With the data retrieved in our experiments in simulated environments [10,11] we have performed a conscientious experimentation comparing the recommendation results of the state-of-the-art aggregation functions presented by [22] when applying them to what we will next define as *standard* and *social* recommendations approaches. The results of these experiments are out of the scope of this paper but can be found

[16] Extracted genres correspond to the genres used by the Web Crawler searches (*http://www.guiadelocio.com*).

[17] We are aware of the limitations of this experiment given the low number of respondents and intend to make further analysis now that we have gathered more information as a result of the experiment carried out in this paper.

in [64]. During this experimentation we have found that *average satisfaction* reported better results for small groups (we consider groups of 10 or less as small) than the other studied aggregation functions, and therefore it is the strategy adopted in *HappyMovie* (as for the moment we do not expect to have large groups using the application):

$$\hat{r}_{G_a,i} \triangleq \frac{1}{G_a}\sum_{u\in G_a}\hat{r}_{u,i} \quad (6)$$

We will designate this baseline recommender by *Standard Group Recommender* which will be our state-of-the-art recommender to compare with.

3.4.1. Social group recommender

As we have previously explained, our approach provides an improvement in the accuracy of predicted group ratings by taking into account users' personality and the strength of their connections (which we refer to as their trust). The prediction strategy that takes this extra social knowledge into account is called the *delegation-based rating* method (*dbr*). Using the *average satisfaction* principle again, *Social Recommenders* that use the *dbr* method are defined as:

$$\hat{r}_{G_a,i} \triangleq \frac{1}{G_a}\sum_{u\in G_a} dbr(\hat{r}_{u,i}, G_a) \quad (7)$$

Here the *average satisfaction* principle is not applied directly to individual predicted ratings, $\hat{r}_{u,i}$. The ratings are modified by the *dbr* function, which takes into account personality and trust values within the group G_a to compute what we call a *delegation-based rating*. The delegation-based method recognizes that a person's opinions may be based in part on the opinions of other group members. Basically, in each user's turn the user's opinion is not taken into account but it is considered in the other $(n-1)$ turns that is when the user influences others. We know that this is not an intuitive idea. Basically, instead of taking users' opinion once into account, the method takes it several times into account, once for each other user in the group. In our previous work [11,64], when testing our method in simulated environments, we showed that our delegation-based method improves the accuracy of predicted group ratings more than any other *standard* or *social* approach that we have studied. We here present a refined version of the original formula presented in [11,64]. The formula, which we explain below, is as follows:

$$dbr(\hat{r}_{u,i}, G_a) = \frac{1}{T}\sum_{v\neq u\in G_a} t_{u,v}[\hat{r}_{v,i} + \theta_{r_{v,i}} \cdot \Delta p_{u,v}] + m_v \quad (8)$$

where

$$T = \sum_{v\neq u\in G} t_{u,v}$$

$$\Delta p_{u,v} = p_v - p_u$$

$$m_v = \alpha(1 - s_v)p_v;$$

In Eq. (8), $t_{u,v}$ denotes the trust between u and v, which is a real number between 0.0 (no connection) and 1.0 (strong connection). For a given user u in group G_a, we take into account the predicted ratings, $\hat{r}_{v,i}$, for the rest of the group members, $v \in G_a$, $v \neq u$, weighted by the trust between the two users, $t_{u,v}$. This follows [21], where a method for group recommendations using trust is proposed.

Variable p_u denotes user u's personality, also a real number between 0.0 (very cooperative) and 1.0 (very selfish). The predicted rating of the other group members $\hat{r}_{v,i}$ is increased or decreased depending on the difference in personalities, $\Delta p_{u,v}$. This way, users with stronger personalities will contribute more to the final score.

In this paper (differently to [11,64]) we have included a $\theta_{r_{v,i}}$ factor. We believe that when modifying a user's predicted preference, $\hat{r}_{u,i}$, for an item i according to trusted friends' preferences, $\hat{r}_{v,i}$, (this is *dbr*'s goal) it is necessary to acknowledge whether the preference of the trusted friend, $\hat{r}_{v,i}$, is positive or negative with respect to the questioned item i. Meaning that a user v with a positive opinion of i, i.e. where $\hat{r}_{v,i}$ is greater than the mid-point of the ratings scale, will want to increase u's opinion of i; but if v has a negative opinion, i.e. where $\hat{r}_{v,i}$ is less than the mid-point of the scale, then v will want to decrease u's opinion. We now model this through a function θ:

$$\theta_{r_{v,i}} \triangleq \begin{cases} 5 & \text{if } \hat{r}_{v,i} \geqslant mid \\ -5 & \text{otherwise} \end{cases} \quad (9)$$

where *mid* is the mid-point of the ratings scale, e.g. 3 on a five-point Likert scale. We have chosen constants 5 and -5 because after several studies in group personality composition [13,11] we have observed that the mean difference in group personality composition is 0.2 and therefore the impact of $\theta_{r_{v,i}} \cdot \Delta p_{u,v}$ in Eq. (8) will typically be 1 or -1, which in comparison with other tested ranges has proven to be the most adequate.

Finally, we now include m_v, that represents the memory of past recommendations. The satisfaction value s_v is the level of satisfaction of user v, as explained in Section 3.2. Note that initially all users are assigned a $s_v = 1$. Therefore, the first time that a group receives a recommendation the memory factor is nullified in the formula as it is not necessary because there are not previous recommendations. Parameter α is used to modify the impact of memory in *dbr*. It has a positive or negative value according to $\hat{r}_{v,i}$ in the same way that $\theta_{r_{v,i}}$ has. In the experiments carried out in Section 4.4 we have considered $\alpha = 1$ to get a clearer picture of the impact of the satisfaction value. It is important to note that this satisfaction value is also weighted depending on user v's personality to reflect the importance of satisfying that concrete user. Once the recommendation process has finished the s_v value is updated for every user. Note that here we have intentionally omitted the time-stamp (t) for the sake of readability.

The recommender recommends the k items i from T_a for which $\hat{r}_{G_a,i}$ is highest. We will designate this recommender by *Social Group Recommender*.

4. Experimental evaluation

In order to verify our *Social Group Recommender* method and *HappyMovie*'s usability we have run an experiment with real users testing our application. Concretely, we have performed four different evaluations: E_1) a functional evaluation of the application to validate its performance from the users' point of view, E_2) a conceptual evaluation to validate the improvement of using *social recommenders* versus *standard recommenders*, E_3) a conceptual evaluation to validate the necessity of using recommendations that take into account users' satisfaction with past recommendations and E_4) a descriptive analysis of the social factors that enhance the recommendations.

We managed to gather 60 users (25 females and 35 males) that completed the whole experiment. Users are students in their twenties from an AI course. All participants used Facebook regularly and had been members for at least one year.

In order to make a further analysis of the results we have considered three different stratified analysis according to personality (strong or mild), genre (male or female) and trust (high or low). However, this last analysis (trust) cannot be performed because it is not an individual feature such as personality and it changes

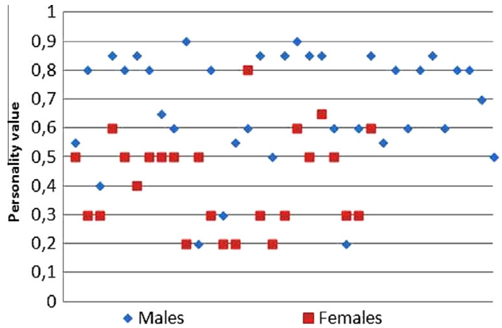


Fig. 7. Distribution of users's personality according to their genre.

in relation to each pair of group members.[18] Therefore we set-up the comparison of the following subpopulations: users whose personality score (p_u) is higher than 0.6 and are therefore considered as "Strong Personality" users (32 users fell into this fold) compared to users whose personality score (p_u) is lower than 0.6 and are therefore considered as "Mild Personality" users (28 users fell into this fold) and males compared to females.

Doing this stratified analysis we have found a correlation between genre and personality. We have observed (as shown in Fig. 7) that male students tend to have higher personality values (average 0.677), implying more selfish personalities (74.28% of the males fell into the strong personality fold), whereas female students have a lower average personality value (0.422), implying less selfish personalities (80% of the females fell into the mild personality fold). We have performed the *ANOVA* test [65] to study the effect that genre has on the personality value and found by refuting the null hypothesis that there is indeed a relation between these two variables.[19] However, as we are not psychological experts we will not draw firm conclusions on this matter, as there might be something in the personality test that we are unaware of, maybe in the way that questions are formulated that leads to these gender differences. Therefore we do not want to extrapolate it to a general population fact and remark it as an intrinsic aspect of this sample.

4.1. E_1) functional evaluation

Users were asked to test the functionality offered by the application and answer a questionnaire. More precisely they were asked to perform the following steps:

Step 1. Answer the personality test through the movie metaphor (Fig. 3). **Step 2.** Answer the preferences test (Fig. 6). Users were asked to rate as many movies as possible, if they had not watched the movie they could skip it. In the end we have gathered 3863 ratings, with an average of 64 ratings per user. Our preferences test allows to distinguish if users don't know a movie, or if they know the movie but have not watched it on purpose because they totally despise it. For example, a user might have seen some *Almodovar*'s movies, and do not intend to see any more because s/he does not like them, giving a score of 0 to the new *Almodovar*'s movie. **Step 3.** Check the accuracy of the recommended movies presented by the "Individual Recommendation" tab. **Step 4.** Meet together grouping themselves in groups of 3 people and create an event to go to the movies together. **Step 5.** Look at the complete current movie listing at local cinemas and debate until there is an agreement which 3 movies they would like to see in a movie outing (users were also asked to individually think which 3 movies they would like to see before performing step 6). **Step 6.** Check the 3 best movies that the application has found for the group.[20] Internally debate whether they would follow or not the recommendation and how satisfied they are with it. Individually rate the presented movies through the 5-star system presented in the event's page (see Fig. 8).

Next they answered individually the following questions, with a five star Likert scale[21]:

Q1. Usefulness (u): "*I find the application useful (being 0 not useful at all and 5 very useful)*".
Q2. Decision process (dP): "*It is useful because it speeds up the group decision process (being 0 very little and 5 a lot)*".
Q3. Reusability (r): "*I will use the application to go to the movies with my friends (being 0 very little and 5 a lot)*".
Q4. Usability (i): "*The application is intuitive and easy to use (being 0 not at all intuitive and 5 very intuitive)*".
Q5. Individual Recommendation (iR): "*I like the individual recommendation of the system (being 0 barely and 5 a lot)*".
Q6. Individual Group Recommendation (iGR): "*I individually like the group recommendation of the system (being 0 barely and 5 a lot)*".
Q7. Group Recommendation (gR): "*As a group we like the group recommendation of the system (being 0 barely and 5 a lot)*".
Q8. Personality Test (perT): "*Was it easy to answer to the personality test? (being 0 very easy and 5 not easy at all)*".
Q9. Preferences Test (preT): "*Was it easy to answer to the preferences test? (being 0 very easy and 5 not easy at all)*".
Q10. Social Network (sN): "*Do you like having the application in a Social Network? (being 0 not at all positive and 5 very positive)*".

Fig. 9 shows the test's general results and Fig. 10 shows the average of the results when analyzing the stratified data. Note that results for users with more selfish personality values are very similar to results for male users; and results for users with less selfish personality values are very similar to results for female users. This follows from the background observation we made, that male students had on average more selfish personalities than female students. When comparing "Strong Personality" users' answers with the "Mild Personality" ones we do not find significant differences. Both subpopulations seem equally (dis) satisfied. We now evaluate users' answers to each question:

Q1. Usefulness (u) [$\overline{u} = 3.86,\ s_u = 0.911$]: reflects that users' opinion about the application usefulness is very high (being 5 the top value).
Q2. Decision process (dP) Average score [$\overline{dP} = 3.69,\ s_{dP} = 1.20$]: supports a positive opinion about the speed up of the decision process given by the application.
Q3. Reusability (r) Average score [$\overline{r} = 3.56,\ s_r = 1.23$]: reflects users' good predisposition to use frequently the application. This inclination is probably motivated by the two previous answers where users expressed that they consider the application useful and that it speeds up their group decision, therefore they intend to return and use it again.
Q4. Usability (i) Average score [$\overline{i} = 3.89,\ s_i = 1.08$]: this high response towards the usability question reflects that users think that the application is intuitive and easy to use without further instructions.

---

[18] We could have used the average trust, however, we did not consider that it would be a representative variable as it always tends to similar values (around 0.4) as explained in Section 4.4.
[19] The obtained *F*-value and *p*-value are 28.6828 and <0.0001 respectively.
[20] These 3 movies are presented in *HappyMovie*'s event page (see Fig. 8), where apart from event related information (guests, celebration place, date and time, etc.) the application displays the best 3 retrieved movies for the current group along with a 5-star voting system that allows *HappyMovie* to update users' satisfaction value (s_u).
[21] We ran the experiment with students whose first language was Spanish. The questions that we show here are paraphrases into English of the Spanish questionnaire.

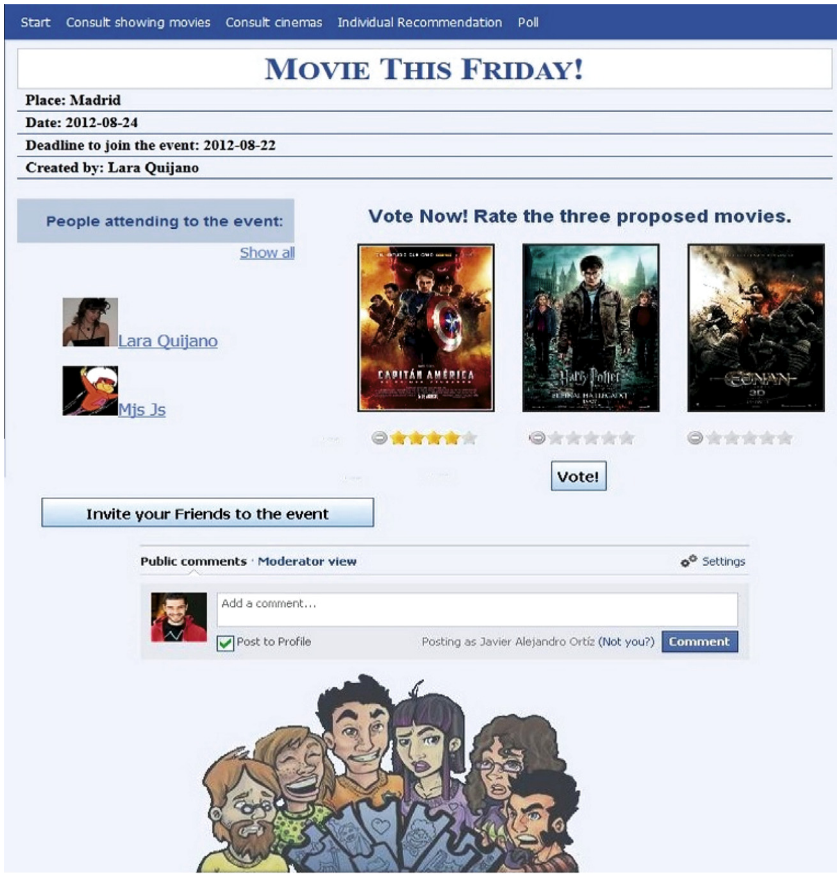


Fig. 8. *HappyMovie*'s event page.

Q5. Individual Recommendation (iR) Average score [$\overline{iR} = 3.89$, $s_{iR} = 1.01$]: reflects users' happiness with their individual recommendations.

Q6. Individual Group Recommendation (iGR) Average score [$\overline{iGR} = 3.93$, $s_{iGR} = 1.06$]: reflects that users are individually satisfied with the group recommendations.

Q7. Group Recommendation (gR) Average score [$\overline{gR} = 3.94$, $s_{gR} = 0.921$]: shows that users think that the application made a good group recommendation. We consider that this is a good result towards our *Social Group Recommendation Method* as this is one of the questions with the highest scoring.

Q8. Personality Test (perT) Average score [$\overline{perT} = 1.33$, $s_{perT} = 1.07$]: this value is on average not very high which is good, because it means that users do not resent doing the personality test. With the interactive metaphor we have managed to make the application more usable and entertaining.

Q9. Preferences Test (preT) Average score [$\overline{preT} = 2.24$, $s_{preT} = 1.33$]: this value is by far the worst result in the questionnaire. We consider that although it is not high enough (more than 3) to represent that users resent answering the preferences test it is quite a high value. However, we decided that it was worth sacrificing this question's results, asking users to rate a lot of movies in the preferences test (70 movies), in order to offer better recommendations (as we have explained in Section 3.3.1 the more precise the individual profile is the better the recommenders perform).

Q10. Social Network (sN) Average score [$\overline{sN} = 4.41$, $s_{sN} = 0.9$]: from this answer we can conclude that users totally approve having the application in a Social Network.

4.2. E$_2$) Social recommender vs. standard recommender

We have also tested whether social factors improve the performance of group recommendations. This premise, that was proven to be true in our experiments with simulated environments [10,11,64], has been now also confirmed when using our *Social Group Recommendation Method* through *HappyMovie*.[22]

Once every group had answered the test, we replaced the *Social Group Recommender* (Eq. (7)) by the *Standard Group Recommender* (Eq. (6)). Users were asked to repeat steps 5 and 6 of the experiment and answer again questions *Q6. Individual Group Recommendation* and *Q7. Group Recommendation*.[23] Fig. 11 shows the comparison between answers to both questions with and without the inclusion of social factors in the group recommendation. As we can see users' opinion about the group recommendation, individually and as a group, is far higher when *HappyMovie* has a *Social Group Recommender* configuration than when it has a *Standard Group Recommender* configuration. Statistical significance tests have been carried out to prove that these differences are significant. We have used Wilcoxon test [66] as our sample is not normally distributed. Firstly we have proven that there are no dependencies

[22] Note that in our previous experiments we performed severe testing comparing our *Social Group Recommendation Method* with several state-of-the-art group recommenders that did not use social factors. In this paper we will just limit to confirm our previous conclusions by comparing our results with a state-of-the-art recommender that uses an average satisfaction approach as explained in Eq. (6).

[23] Note that users were not explained at any moment the concepts of social or standard recommenders, the purpose of our experiment or that we had changed the recommendation methods. Users were just asked to check again for the movies that the recommender proposed and answer whether they liked this proposal better or worse than the previous one by answering again to questions Q6 and Q7.

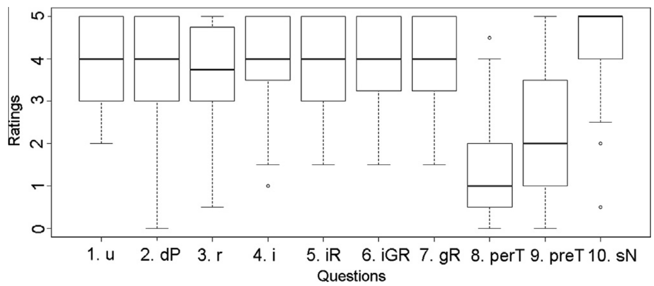


Fig. 9. Users answers to *HappyMovie*'s questionnaire.

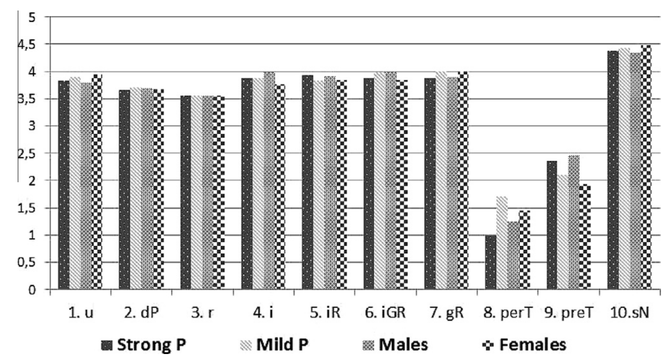


Fig. 10. Average of user's answers to *HappyMovie*'s questionnaire. Data analysis comparison.

between both answers and that they do not follow a random pattern by choosing as null hypothesis that "the ordinal relationships among the measures are equal to". We have obtained *p*-values < 0.05 (6.886e−06 for Social vs. standard comparing question's *Q6* results and 6.251e−06 for Social vs. standard comparing question's *Q7* results). This proves (as Wilcoxon test assures) that when we compare two rated samples and the *p*-value is lower than 0.05 we can assure that the two series are different, are not a result of chance and are said to be statistically significant. We have performed a variation of Wilcoxon test, this time, proving by choosing the null hypothesis as "less than" that *Standard Group Recommendation* results are smaller than the *Social Group Recommendation* ones. With this test we have also obtained *p*-values < 0.05 (3.443e−06 for Social vs. standard comparing question's *Q6* results and 3.125e−06 for Social vs. standard comparing question's *Q7* results). Therefore we can conclude that indeed our *Social* method improves the performance of other group recommendations that do not use social factors.

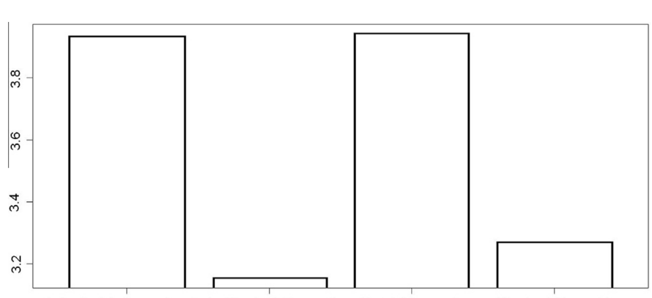


Fig. 11. Users reaction to *Standard* and *Social Group Recommenders*.

4.3. E_3) conceptual evaluation of the long term recommendation module

Next, we have tested the impact of users' opinion through time. This was motivated by the idea that, as reflected in answer *Q3. Reusability* from our questionnaire, users would like to use regularly our application when going to the movies. To do so, we asked users to return one month later and use the application again with the same group configuration. Consequently, cinemas' movie listings had changed. As we explained in Sections 3.2 and 3.4 our group recommendation method keeps a memory of past recommendations and tries to ensure a balanced user satisfaction. We asked users to repeat again steps 5 and 6 of the experiment and answer to questions *Q6. Individual Group Recommendation* and *Q7. Group Recommendation*, that were now slightly modified for its better adjustment to this part of the experiment as follows:

Q6. Individual Group Recommendation (iGR): "*I am individually satisfied with the group recommendation of the system (being 0 barely and 5 a lot)*".

Q7. Group Recommendation (gR): "*We are satisfied as a group with the group recommendation of the system (being 0 barely and 5 a lot)*".

One month later users were asked to return again (so that cinemas' movie listings had changed again), repeat the 5th and 6th step of the experiment and answer for the third time questions *Q6* and *Q7* from our questionnaire. Fig. 12 shows users' satisfaction with our group recommendation individually and as a group. Looking at users' answers, both individually and as a group, users' satisfaction with the recommendations keeps improving reaching out in the third time values higher than 4.5. In the 2nd and 3rd time users' average individual satisfaction is higher than group satisfaction. This can be due to a feeling that the recommender has favored them, or that they have "won" in the decision about which movie the group will watch. This is not a surprising result as it was our goal (as explained in Section 3.2) to favor those users less satisfied with the previous recommendation through the satisfaction value

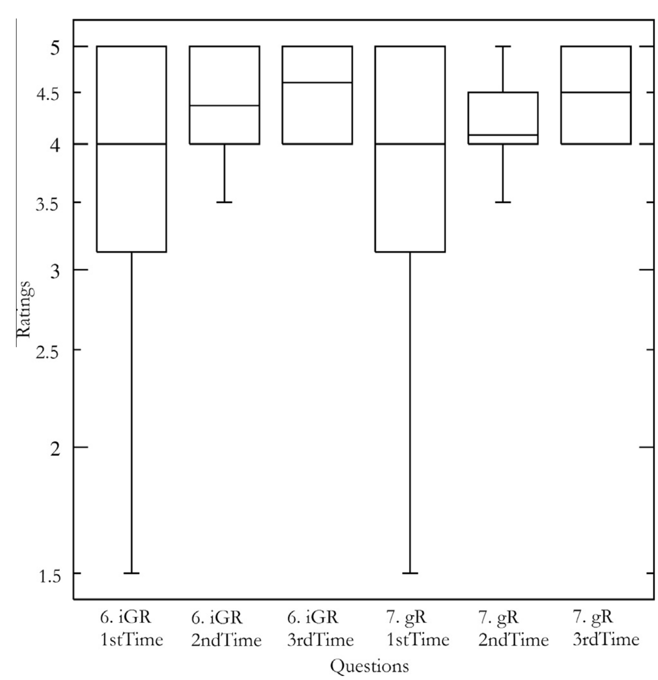


Fig. 12. Users reaction to our group recommendations over time.

Table 1
p-value results for Wilcoxon test.

Results	*p*-value "equal to" n.h	*p*-value "less than" n.h
1st time vs. 2nd time Indv	0.04654	0.02327
1st time vs. 2nd time group	0.00135	0.0006751
2nd time vs. 3rd time Indv	0.001715	0.0008573
2nd time vs. 3rd time group	0.001715	0.0008573

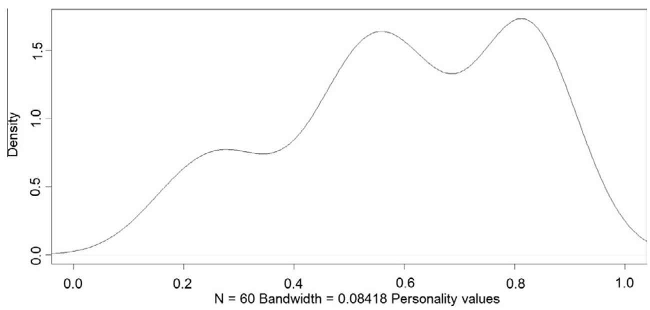


Fig. 13. Density distribution of the personality factor.

(s_u). Regarding this fact, the most important conclusion is that we can appreciate that there are no dissatisfied users the 2nd time and that in the 3rd time not only there are no dissatisfied users but also only highly satisfied ones. Hence, we can conclude that it is worth having a system that contemplates users' satisfaction over time and tries to ensure a general global satisfaction as results show users' increasing well-fare. These results have been statistically confirmed through Wilcoxon test [66] obtaining *p*-value results less than 0.05, as shown in Table 1. As it can be seen, we have tested the two different null hypothesis (n.h) "equal to" (that proves that results are different and are not a result of chance and therefore statistically significant) and "less than" (that proves that the differences in the results are statistically significant).

Moreover we have performed numerical analysis of the personality and trust factors to find correlations and study their impact on the recommendation method.

4.4. E_4) descriptive analysis of the social factors

Analysis of the personality factor: We have analyzed users' answers to the given personality test. The average p_u is 0.57 and the p_u values distribution is reflected in Fig. 13 where the standard deviation is 0.221. We can conclude from this analysis than on average users have high personality values, as the majority of them are comprehended in the [0.5, 0.8] range. However the variance in users' personality is high, we have users with very strong personalities (0.9) and very mild personalities (0.2).

Analysis of the trust factor: Once our experiment was over, we analyzed the trust between each group member and found a lot of diversity in the data. The average trust is 0.41 and the standard deviation is 0.259. Different users had completely opposite levels of trust with values of 0.86 or 0.10. This means that our sample has a varied representation of relationships, some of them are just classmates and others are close friends.[24] On average trust is not very high, we consider this aspect to be predictable due to the sample of people taken. For example, when testing the obtained improvement in the recommendation accuracy when using our *Social Group Recommender* method in simulated environments [10,11] our data was formed by our friends in two different Social Networks thus the average trust was higher (0.597). We have also studied if trust was related to personality, and for example groups with strong or mild personality on average had strong or weak trust between them. However, we have not found any evidence of this. From this study we can conclude that trust between users is related to each person individually and has nothing to do with the personality of each individual. We have found very "trusting" users with strong and mild personalities and the other way round.

[24] This kind of limitation when having to find a sample of objective users is a problem that most researchers find when testing group recommenders [6].

5. Conclusions and future work

This paper extends our previous work regarding *Social Group Recommenders* [10,11,64] and presents a comprehensive description of *HappyMovie*, our social recommender application. We provide a complete description of the system that may serve to reproduce the proposed techniques in other recommender systems. To illustrate the advantages of the approach, a functional description and evaluation is presented.

Through the inclusion of social factors – namely personality and trust – the *HappyMovie* system can ease the real decision making process performed by groups of people when choosing a movie to watch together. The simulation of this process is implemented through different modules that obtain and provide social knowledge, estimate the individual and group preferences, and include a long term knowledge management regarding users satisfaction with previous recommendations. Throughout the different experiments presented in this paper we have proven users' acceptance towards the system and tested the higher acceptance of our *Social Group Recommendation Method* proposals compared to the ones provided by the *Standard Group Recommender*. We have also justified the need for considering a system that takes into account previous group recommendations events by evaluating the global welfare and satisfaction of users through time.

Users' answers to our different questionnaires have reflected that they are willing to use *HappyMovie*. Some of the reasons for this positive response are that users believe that *HappyMovie* eases group decisions and that it is easy to use. But the most interesting and important feedback that users have given us is that they like the individual and group recommendations that *HappyMovie* offers and that this positive opinion increases the more they use the application. Therefore we believe that it is worth making the application public.

One extra advantage of building *HappyMovie* has been obtaining data, such as ratings, personality values, etc. This was a very difficult and costly matter when we had our recommendation method embedded in a standalone system ([10,11]). With *HappyMovie* we have been able to extract automatically most of the data required by our system (for example the trust factor) and also to obtain a bigger sample of data (we now count with a database of 3863 ratings). We will now be able to conduct further experiments using the data obtained and expect to obtain more if users start using *HappyMovie* every time they need a movie recommendation.

Although our *social recommendation* approach has been applied to the movies domain, it can be reproduced in other domains as *HappyMovie* follows a generic architecture called ARISE and a development process based on software templates has also been provided [12]. As future work we would like to obtain data enough to carry Big Data experiments, provide a public group recommender dataset, and also perform further group analysis, like measuring how dissimilar or similar are preferences within a group, diversity, serendipity, etc. Another interesting on-going work is the inclusion of explanations to users, through them the system will try to justify the proposed items and increase users' level of acceptance by displaying others' needs.

References

[1] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.
[2] A. Jameson, B. Smyth, Recommendation to groups, in: The Adaptive Web, Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Springer, 2007, pp. 596–627. ISBN 978-3-540-72078-2.
[3] L. Baltrunas, T. Makcinskas, F. Ricci, Group recommendations with rank aggregation and collaborative filtering, in: RecSys '10, ACM, New York, NY, USA, 2010, pp. 119–126.
[4] S. Berkovsky, J. Freyne, Group-based recipe recommendations: analysis of data aggregation strategies, in: RecSys, 2010, pp. 111–118.
[5] T.D. Pessemier, S. Dooms, L. Martens, Comparison of group recommendation algorithms, in: Multimedia Tools an Applications, 2013.
[6] J. Masthoff, A. Gatt, In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems, User Model. User-Adapt. Interact. (2006) 281–319.
[7] C.A. Erdley, M. Rivera, E. Shepherd, L.J. Holleb, Social-cognitive models and skills, in: Practitioners Guide to Empirically Based Measures of Social Skills, Springer, 2010, pp. 21–35.
[8] N.A. Christakis, J.H. Fowler, Social Contagion Theory: Examining Dynamic Social Networks and Human Behavior, CoRR abs/1109.5235, 2011.
[9] M. Bazire, P. Brézillon, Understanding context before using it, in: Context, 2005, pp. 29–40.
[10] L. Quijano-Sánchez, J.A. Recio-García, B. Díaz-Agudo, Personality and social trust in group recommendations, in: ICTAI'10, IEEE Computing Society, 2010, pp. 121–126.
[11] L. Quijano-Sánchez, J.A. Recio-García, B. Díaz-Agudo, G. Jiménez-Díaz, Social factors in group recommender systems, in: ACM TIST, vol. 4, 2013, p. 8.
[12] L. Quijano-Sánchez, J.A. Recio-García, B. Díaz-Agudo, A reusable methodology for the instantiation of social recommender systems, in: 25th IEEE International Conference on Tools with Artificial Intelligence, ICTAI'13, 2013, pp. 777–782.
[13] L. Quijano-Sánchez, J.A. Recio-García, B. Díaz-Agudo, HappyMovie: a facebook application for recommending movies to groups, in: ICTAI'11, 2011, pp. 239–244.
[14] J. McCarthy, T. Anagnost, MusicFX: an arbiter of group preferences for computer supported collaborative workouts, in: CSCW '98, ACM, 1998, pp. 363–372.
[15] A. Crossen, J. Budzik, K. Hammond, Flytrap: intelligent group music recommendation, in: IUI '02, ACM, 2002, pp. 184–185.
[16] M. O'Connor, D. Cosley, J. Konstan, J. Riedl, PolyLens: a recommender system for groups of users, in: ECSCW'01, ISBN 0-7923-7162-3, 2001, pp. 199–218.
[17] J. McCarthy, Pocket restaurant finder: a situated recommender systems for groups, in: Proc. in Workshop on Mobile Ad-Hoc Communication at CHI'02, ACM, 2002.
[18] A. Jameson, More than the sum of its members: challenges for group recommender systems, in: AVI '04, ACM, 2004, pp. 48–54.
[19] J. Golbeck, Combining provenance with trust in social networks for semantic web content filtering, in: IPAW'06, Lecture Notes in Computer Science, Springer, 2006, pp. 101–108. ISBN 3-540-46302-X.
[20] R. Sinha, K. Swearingen, Comparing recommendations made by online systems and friends, in: DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries, 2001.
[21] J. Golbeck, Generating predictive movie recommendations from trust in social networks, in: iTrust: 4th International Conference on Trust Management, 2006, pp. 93–104.
[22] J. Masthoff, Group modeling: selecting a sequence of television items to suit a group of viewers, User Model. User-Adapt. Interact. 14 (1) (2004) 37–85.
[23] C. Baccigalupo, E. Plaza, A case-based song scheduler for group customised radio, in: ICCBR, Belfast, 2007, Proceedings, Lecture Notes in Computer Science, vol. 4626, Springer, 2007, pp. 433–448. ISBN 978-3-540-74138-1.
[24] I. Konstas, V. Stathopoulos, J.M. Jose, On social networks and collaborative recommendation, in: SIGIR, 2009, pp. 195–202.
[25] W.-P. Lee, C. Kaoli, J.-Y. Huang, A smart TV system with body-gesture control, tag-based rating and context-aware recommendation, Knowl.-Based Syst. 56 (2014) 167–178.
[26] H. Lieberman, N.W.V. Dyke, A.S. Vivacqua, Let's browse: a collaborative web browsing agent, in: IUI, 1999, pp. 65–68.
[27] X.H. Jiliang Tang, H. Liu, Social recommendation: a review, in: Social Network Analysis and Mining, Springer, 2013.
[28] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, K. Seada, Enhancing group recommendation by incorporating social relationship interactions, in: GROUP, 2010, pp. 97–106.
[29] S. Huanga, X. Sub, Y. Huc, S. Mahadevand, Y. Deng, A new decision-making method by incomplete preferences based on evidence distance, Knowl.-Based Syst. 56 (2014) 264–272.
[30] A. Mislove, B. Viswanath, P.K. Gummadi, P. Druschel, You are who you know: inferring user profiles in online social networks, in: WSDM, 2010, pp. 251–260.
[31] N. Ma, E.-P. Lim, V.-A. Nguyen, A. Sun, H. Liu, Trust relationship prediction using online product review data, in: CIKM-CNIKM, 2009, pp. 47–54.
[32] M. Jamali, M. Ester, TrustWalker: a random walk model for combining trust-based and item-based recommendation, in: KDD, 2009, pp. 397–406.
[33] F. Zhang, L. Bai, F. Gao, A user trust-based collaborative filtering recommendation algorithm, in: ICICS, 2009, pp. 411–424.
[34] S. Nepal, C. Paris, S.K. Bista, SRec: a social behaviour based recommender for online communities, in: UMAP Workshops, 2012.
[35] M.S. Pera, N. Condie, Y.-K. Ng, Personalized book recommendations created by using social media data, in: WISE Workshops, 2010, pp. 390–403.
[36] X. Yang, H. Steck, Y. Guo, Y. Liu, On top-k recommendation using social networks, in: RecSys, 2012, pp. 67–74.
[37] J. Bao, Y. Zheng, M.F. Mokbel, Location-based and preference-aware recommendation using sparse geo-social networking data, in: SIGSPATIAL/GIS, 2012, pp. 199–208.
[38] X. Hu, C. Chen, X. Chen, Z.-K. Zhang, Social Recommender Systems Based on Coupling Network Structure Analysis, CoRR abs/1204.1949, 2012.
[39] U. Shardanand, P. Maes, Social information filtering: algorithms for automating word of mouth, in: CHI, 1995, pp. 210–217.
[40] IBM, Ibm's Black Friday Report Says Twitter Delivered 0 Percent of Referral Traffic and Facebook Sent Just 0.68 Percent, <https://strme.wordpress.com/2012/11/27/ibmsblack-friday-report-says-twitter-delivered-0-percent-of-referral-traffic-and-facebook-sentjust-0-68-percent/>, 2012.
[41] Quora, Why Does the Startup Idea of Social Recommendations Consistently Fail?, <http://www.quora.com/Why-does-the-startup-idea-of-social-recommendationsconsistently-fail>, 2012.
[42] Y. Chen, L. Cheng, C. Chuang, A group recommendation system with consideration of interactions among group members, Expert Syst. Appl. (2008) 2082–2090. ISSN 0957-4174, doi: < http://dx.doi.org/10.1016/j.eswa.2007.02.008>.
[43] K. Bischoff, Exploiting social ties for search and recommendation in online social networks – challenges and chances, in: Grundlagen von Datenbanken, 2010.
[44] Merriam-Webster's Collegiate Dictionary, tenth ed., 2002.
[45] H. Fujita, J. Hakura, M. Kurematsu, Intelligent human interface based on mental cloning-based software, Knowl.-Based Syst. 22 (3) (2009) 216–234.
[46] A. Minamikawa, H. Fujita, J. Hakura, M. Kurematsu, Personality estimation application for social media, in: SoMeT, 2012, pp. 327–335.
[47] K. Thomas, R. Kilmann, Thomas–Kilmann Conflict Mode Instrument, Tuxedo, N.Y., 1974.
[48] E. Berne, Games People Play The Basic Hand Book of Transactional Analysis, Ballantine Books, New York, 1964.
[49] P.T. Barrett, K. Petrides, S.B. Eysenck, H.J. Eysenck, The Eysenck personality questionnaire: an examination of the factorial similarity of P, E, N, and L across 34 countries, Person. Individual Differences 25 (5) (1998) 805–819.
[50] J. Paul, T. Costa, R.R. McCrae, Domains and facets: hierarchical personality assessment using the revised NEO personality inventory, J. Person. Assess. 64 (1) (1995) 21–50.
[51] R.R. McCrae, Oliver P. John, An introduction to the five-factor model and its applications, J. Person. 60 (1992) 175–215.
[52] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. 22 (1) (2004) 5–53. ISSN 1046-8188, doi: <http://doi.acm.org/10.1145/963770.963772>.
[53] N.A. Schaubhut, Technical Brief for the Thomas–Kilmann Conflict Mode Instrument, CPP Research Department, 2007.
[54] J.A. Recio-García, G. Jimenez-Diaz, A.A. Sánchez-Ruiz, B. Díaz-Agudo, Personality aware recommendations to groups, in: Proc. of the 2009 ACM Conference on Recommender Systems, ACM, 2009, pp. 325–328.
[55] P. Avesani, P. Massa, R. Tiella, A trust-enhanced recommender system application: Moleskiing, in: SAC '05, ACM, NY, USA, 2005, pp. 1589–1593.
[56] E. Gilbert, K. Karahalios, Predicting tie strength with social media, in: CHI '09, ACM, 2009, pp. 211–220.
[57] D.Z. Levin, R. Cross, L.C. Abrams, The strength of weak ties you can trust: the mediating role of trust in effective knowledge transfer, Manage. Sci. 50 (2004) 1477–1490.
[58] M. Granovetter, The strength of weak ties, Am. J. Sociol. 78 (6) (1973) 1360–1380.
[59] Y.-G. Cho, K.-T. Cho, A loss function approach to group preference aggregation in the AHP, Comput. OR 35 (3) (2008) 884–892.
[60] M. Volkovs, R.S. Zemel, A flexible generative model for preference aggregation, in: WWW, 2012, pp. 479–488.
[61] P. Lops, M. de Gemmis, G. Semeraro, Content-based recommender systems: state of the art and trends, in: Recommender Systems Handbook, 2011, pp. 73–105.
[62] M. Ekstrand, J. Riedl, J. Konstan, Collaborative filtering recommender systems, Foundat. Trends Human–Comput. Interact. 4 (2) (2011) 175–243.
[63] S. Sarawagi, Information extraction, Foundat. Trends Databases 1 (3) (2008) 261–377.
[64] L. Quijano-Sánchez, J.A. Recio-García, B. Díaz-Agudo, Group recommendation methods for social network environments, in: 3rd Workshop on Recommender Systems and the Social Web, RecSys, 2011, pp. 383–384.
[65] D.C. Howell, Statistical Methods for Psychology, fifth ed., Pacific Grove, CA: Duxbury/Thomson Learning, ISBN 0-534-37770-X, 2002.
[66] F. Wilcoxon, Individual comparisons by ranking methods, in: Biometrics Bulletin, 1, 1945, pp. 80–83.

Bibliography

Merriam-Webster's Collegiate Dictionary 10th Ed. 2002.

New Oxford American Dictionary 3rd Ed. 2010.

Aamodt, A. and Plaza, E. Case-based reasoning: Foundational issues,methodological variants, and system approaches. *Artificial Intelligence Communications*, vol. 7(1), pages 39–59, 1994.

Abramowitz, M. and Stegun, I. A. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, ninth dover printing, tenth gpo printing edition, 1964.

Adomavicius, G. and Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, vol. 17(6), pages 734–749, 2005.

Anderson, C. Calculating latent demand in the long tail. In *KDD*, page 1. 2007.

Ardissono, L., Gena, C., Torasso, P., Bellifemine, F., Difino, A. and Negro, B. User modeling and recommendation techniques for personalized electronic program guides. In *Personalized Digital Television. Targeting Programs to Individual Viewers, volume 6 of Human-Computer Interaction Series, chapter 1*, pages 3–26. Kluwer Academic Publishers, 2004.

Avesani, P., Massa, P. and Tiella, R. A trust-enhanced recommender system application: Moleskiing. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 1589–1593. ACM, NY, USA, 2005. ISBN 1-58113-964-0.

Baltrunas, L., Makcinskas, T. and Ricci, F. Group recommendations with rank aggregation and collaborative filtering. In *RecSys '10*, pages 119–126. ACM, New York, NY, USA, 2010.

Bao, J., Zheng, Y. and Mokbel, M. F. Location-based and preference-aware recommendation using sparse geo-social networking data. In *SIGSPATIAL/GIS*, pages 199–208. 2012.

BARRETT, P. T., PETRIDES, K., EYSENCK, S. B. and EYSENCK, H. J. The eysenck personality questionnaire: An examination of the factorial similarity of p, e, n, and l across 34 countries. *Personality and Individual Differences*, vol. 25(5), pages 805–819, 1998.

BARSADE, S. G. The ripple effect: Emotional contagion and its influence on group behavior. *Administrative Science Quarterly*, vol. 47(4), pages pp. 644–675, 2002. ISSN 00018392.

BATET, M., MORENO, A., SÁNCHEZ, D., ISERN, D. and VALLS, A. Turist@: Agent-based personalised recommendation of tourist activities. *Expert Syst. Appl.*, vol. 39(8), pages 7319–7329, 2012.

BERKOVSKY, S. and FREYNE, J. Group-based recipe recommendations: analysis of data aggregation strategies. In *RecSys*, pages 111–118. 2010.

BERNE, E. *Games People Play. The Basic Hand Book of Transactional Analysis*. New York: Ballantine Books, 1964.

BILLSUS, D. and PAZZANI, M. J. Learning collaborative information filters. In *ICML*, pages 46–54. 1998.

BILLSUS, D. and PAZZANI, M. J. A personal news agent that talks, learns and explains. In *Agents*, pages 268–275. 1999.

BISCHOFF, K. Exploiting social ties for search and recommendation in online social networks - challenges and chances. In *Grundlagen von Datenbanken*. 2010.

BOBADILLA, J., SERRADILLA, F. and HERNANDO, A. Collaborative filtering adapted to recommender systems of e-learning. *Knowl.-Based Syst.*, vol. 22(4), pages 261–265, 2009.

BREESE, J. S., HECKERMAN, D. and KADIE, C. M. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI*, pages 43–52. 1998.

BRIDGE, D. G., GÖKER, M. H., MCGINTY, L. and SMYTH, B. Case-based recommender systems. *Knowledge Eng. Review*, vol. 20(3), pages 315–320, 2005.

BURKE, R. D. Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact.*, vol. 12(4), pages 331–370, 2002.

BURT, R. S. Toward a structural theory of action: Network models of social structure, perception and action. pages 1336–1338. 1982.

CANDILLIER, L., MEYER, F. and BOULLÉ, M. Comparing state-of-the-art collaborative filtering systems. In *MLDM*, pages 548–562. 2007.

Chao, D. L., Balthrop, J. and Forrest, S. Adaptive radio: achieving consensus using negative preferences. In *GROUP*, pages 120–123. 2005.

Chen, Y.-L., Cheng, L.-C. and Chuang, C.-N. A group recommendation system with consideration of interactions among group members. *Expert Syst. Appl.*, vol. 34(3), pages 2082–2090, 2008. ISSN 0957-4174.

Choudhury, M. D., Sundaram, H., John, A. and Seligmann, D. D. Analyzing the dynamics of communication in online social networks. In *Handbook of Social Network Technologies and Applications*, pages 59–94. 2010a.

Choudhury, M. D., Sundaram, H., John, A., Seligmann, D. D. and Kelliher, A. "birds of a feather": Does user homophily impact information diffusion in social media? *CoRR*, vol. abs/1006.1702, 2010b.

Christakis, N. A. and Fowler, J. H. Social contagion theory: Examining dynamic social networks and human behavior. *CoRR*, 2011.

Christensen, I. A. and Schiaffino, S. Social influence in group recommender systems. *Online Information Review*, vol. 38(4), pages 5–5, 2014.

Connor, M. J. and Herlocker, J. L. Clustering items for collaborative filtering. In *ACM SIGIR Workshop on Recommender Systems*. 2001.

Cordier, A., Gaillard, E. and Nauer, E. Man-machine collaboration to acquire cooking adaptation knowledge for the taaable case-based reasoning system. In *WWW (Companion Volume)*, pages 1113–1120. 2012.

Costello, E., Doody, J., McGinty, L. and Smyth, B. *i*care: intelligent customer assistance for recommending eyewear. In *IUI*, pages 282–284. 2006.

Crossen, A., Budzik, J. and Hammond, K. J. Flytrap: intelligent group music recommendation. In *IUI*, pages 184–185. 2002.

Deutsch, M. and Gerard, H. B. A study of normative and informational social influences upon individual judgement. *Journal of Abnormal & Social Psychology*, vol. 51(3), pages 629–36, 1955.

Díaz-Agudo, B., González-Calero, P. A., Recio-García, J. A. and Sánchez, A. Building cbr systems with jcolibri. *Special Issue on Experimental Software and Toolkits of the Journal Science of Computer Programming*, vol. 69(1-3), pages 68–75, 2007. ISSN 0167-6423.

Ekstrand, M. D., Riedl, J. and Konstan, J. A. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, vol. 4(2), pages 175–243, 2011.

EL-SHINNAWY, M. M. and VINZE, A. S. Polarization and persuasive argumentation: A study of decision making in group settings. *MIS Quarterly*, vol. 22(2), pages 165–198, 1998.

FOWLER, J. H. and CHRISTAKIS, N. A. Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study. *Brithis Medical Journal*, 2008.

GAILLARD, E., LIEBER, J., NAUER, E. and CORDIER, A. How case-based reasoning on e-community knowledge can be improved thanks to knowledge reliability. In *Case-Based Reasoning Research and Development - 22nd International Conference, ICCBR 2014, Cork, Ireland, September 29, 2014 - October 1, 2014. Proceedings*, pages 155–169. 2014.

GARTRELL, M., XING, X., LV, Q., ANDREST RICHARD HAN, A. B., MISHRA, S. and SEADA, K. Enhancing group recommendation by incorporating social relationship interactions. In *GROUP*, pages 97–106. 2010.

DE GEMMIS, M., LUCA, E. W. D., NOIA, T. D., GANGEMI, A., HAUSENBLAS, M., LOPS, P., LUKASIEWICZ, T., PLUMBAUM, T. and SEMERARO, G., editors. *Proceedings of the second Workshop on Semantic Personalized Information Management: Retrieval and Recommendation 2011, Bonn, Germany, October 24, 2011*, vol. 781 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011.

GILBERT, E. and KARAHALIOS, K. Predicting tie strength with social media. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 211–220. ACM, New York, NY, USA, 2009. ISBN 978-1-60558-246-7.

GLADWELL, M. The tipping point: How little things canmake a big difference. Little Brown, Boston, 2000.

GOLBECK, J. Combining provenance with trust in social networks for semantic web content filtering. In *Provenance and Annotation of Data, International Provenance and Annotation Workshop, IPAW 2006, Chicago, IL, USA, May 3-5, 2006, Revised Selected Papers* (ed. by L. Moreau and I. T. Foster), vol. 4145 of *Lecture Notes in Computer Science*, pages 101–108. Springer, 2006a. ISBN 3-540-46302-X.

GOLBECK, J. Generating predictive movie recommendations from trust in social networks. In *iTrust: 4th International Conference on Trust Management*, pages 93–104. 2006b.

GOODE, E. How culture molds habits of thought. *New York Times*, 2000.

GRANOVETTER, M. The Strength of Weak Ties. *The American Journal of Sociology*, vol. 78(6), pages 1360–1380, 1973.

GRUBER, T. Collective knowledge systems: Where the social web meets the semantic web. *J. Web Sem.*, vol. 6(1), pages 4–13, 2008.

HATFIELD, E., CACIOPPO, J. and RAPSON, R. *Emotional Contagion*. Studies in Emotion and Social Interaction. Cambridge University Press, 1994.

HERLOCKER, J. L. *Understanding and Improving Automated Collaborative Filtering Systems*. PhD Thesis, University of Minnesota, 2000.

HERLOCKER, J. L., KONSTAN, J. A. and RIEDL, J. Explaining collaborative filtering recommendations. In *CSCW*, pages 241–250. 2000.

HERLOCKER, J. L., KONSTAN, J. A. and RIEDL, J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Inf. Retr.*, vol. 5(4), pages 287–310, 2002.

HERLOCKER, J. L., KONSTAN, J. A., TERVEEN, L. G. and RIEDL, J. T. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, vol. 22(1), pages 5–53, 2004. ISSN 1046-8188.

HU, X., CHEN, C., CHEN, X. and ZHANG, Z.-K. Social recommender systems based on coupling network structure analysis. *CoRR*, vol. abs/1204.1949, 2012.

IBM. Ibm's black friday report says twitter delivered 0 percent of referral traffic and facebook sent just 0.68 percent. In *https://strme.wordpress.com/2012/11/27/ibmsblack-friday-report-says-twitter-delivered-0-percent-of-referral-traffic-and-facebook-sentjust-0-68-percent/*. 2012.

JAMALI, M. and ESTER, M. *TrustWalker*: a random walk model for combining trust-based and item-based recommendation. In *KDD*, pages 397–406. 2009.

JAMESON, A. More than the sum of its members: challenges for group recommender systems. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 48–54. ACM, New York, NY, USA, 2004. ISBN 1-58113-867-9.

JAMESON, A., BALDES, S. and KLEINBAUER, T. Two methods for enhancing mutual awareness in a group recommender system. In *AVI*, pages 447–449. 2004.

JAMESON, A. and SMYTH, B. Recommendation to groups. In *The Adaptive Web, Methods and Strategies of Web Personalization* (ed. by

P. Brusilovsky, A. Kobsa and W. Nejdl), vol. 4321 of *Lecture Notes in Computer Science*, pages 596–627. Springer, 2007. ISBN 978-3-540-72078-2.

Jiliang Tang, X. H. and Liu, H. Social recommendation: a review. In *Social Network Analysis and Mining*. Springer, 2013.

Jung, J. J. Attribute selection-based recommendation framework for short-head user group: An empirical study by movielens and imdb. *Expert Syst. Appl.*, vol. 39(4), pages 4049–4054, 2012.

Kaptein, M. and van Halteren, A. Adaptive persuasive messaging to increase service retention: using persuasion profiles to increase the effectiveness of email reminders. *Personal and Ubiquitous Computing*, vol. 17(6), pages 1173–1185, 2013.

Karatzoglou, A. and Weimer, M. Quantile matrix factorization for collaborative filtering. In *EC-Web*, pages 253–264. 2010.

Kompan, M. and Bieliková, M. Social structure and personality enhanced group recommendation. In *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 22nd Conference on User Modeling, Adaptation, and Personalization co-located with the 22nd Conference on User Modeling, Adaptation, and Personalization (UMAP2014), Aalborg, Denmark, July 7-11, 2014.*. 2014.

Konstas, I., Stathopoulos, V. and Jose, J. M. On social networks and collaborative recommendation. In *SIGIR*, pages 195–202. 2009.

Koren, Y. and Bell, R. M. Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 145–186. 2011.

Lawrence, N. D. and Urtasun, R. Non-linear matrix factorization with gaussian processes. In *ICML*, page 76. 2009.

Lazarsfeld, P. and Merton, R. Friendship as a social process: A substantive and methodological analysis. In *Freedom and Control in Modern Society* (ed. by T. A. M. Berger and C. Page), pages 18–66. Van Nostrand, New York, 1954.

Leake, e., D. B. Case-based reasoning: Experiences, lessons, and future directions. In *Menlo Park, CA: AAAI Press/MIT Press, Menlo Park, CA*. 1996.

Lee, D. D. and Seung, H. S. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562. 2000.

Lee, J., Sun, M. and Lebanon, G. A comparative study of collaborative filtering algorithms. *CoRR*, vol. abs/1205.3193, 2012.

Lemire, D. and Maclachlan, A. Slope one predictors for online rating-based collaborative filtering. *CoRR*, vol. abs/cs/0702144, 2007.

Leonard, J. S. *PerTrust: Leveraging personality and trust for group recommendations*. PhD Thesis, 2014.

Levin, D. Z., Cross, R. and Abrams, L. C. The strength of weak ties you can trust: the mediating role of trust in effective knowledge transfer. *Management Science*, vol. 50, pages 1477–1490, 2004.

Li, J., Sun, L. and Wang, J. A slope one collaborative filtering recommendation algorithm using uncertain neighbors optimizing. In *WAIM Workshops*, pages 160–166. 2011.

Liang, T.-P., Lai, H.-J. and Ku, Y.-C. Personalized content recommendation and user satisfaction: Theoretical synthesis and empirical findings. *Journal of Management Information Systems*, vol. 23(3), pages 45–70, 2007.

Lieberman, H., Dyke, N. W. V. and Vivacqua, A. S. Let's browse: a collaborative browsing agent. *Knowl.-Based Syst.*, vol. 12(8), pages 427–431, 1999.

Linden, G., Smith, B. and York, J. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, vol. 7(1), pages 76–80, 2003.

Lops, P., de Gemmis, M. and Semeraro, G. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, pages 73–105. 2011.

Ma, N., Lim, E.-P., Nguyen, V.-A., Sun, A. and Liu, H. Trust relationship prediction using online product review data. In *CIKM-CNIKM*, pages 47–54. 2009.

Masthoff, J. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, vol. 14(1), pages 37–85, 2004. ISSN 0924-1868.

Masthoff, J. and Gatt, A. In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. *User Modeling and User-Adapted Interaction*, vol. 16(3-4), pages 281–319, 2006. ISSN 0924-1868.

Maybury, M. T., Greiff, W. R., Boykin, S., Ponte, J. M., McHenry, C. and Ferro, L. Personalcasting: Tailored broadcast news. *User Model. User-Adapt. Interact.*, vol. 14(1), pages 119–144, 2004.

McCarthy, J. F. Pocket restaurant finder: A situated recommender systems for groups. In *Proceeding of Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems*. 2002.

McCarthy, J. F. and Anagnost, T. D. MusicFX: An arbiter of group preferences for computer aupported collaborative workouts. In *CSCW '98: Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 363–372. ACM, 1998.

Mccarthy, K., Reilly, J., Mcginty, L. and Smyth, B. Thinking positively - explanatory feedback for conversational recommender systems. Informe técnico, In Proceedings of the ECCBR 2004 Workshops, 2004.

McCarthy, K., Salamó, M., Coyle, L., McGinty, L., Smyth, B. and Nixon, P. Cats: A synchronous approach to collaborative group recommendation. In *FLAIRS Conference*, pages 86–91. 2006.

McDonald, D. W. Recommending collaboration with social networks: a comparative evaluation. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 593–600. ACM, New York, NY, USA, 2003. ISBN 1-58113-630-7.

McGinty, L. and Smyth, B. Improving the performance of recommender systems that use critiquing. In *ITWP*, pages 114–132. 2003a.

McGinty, L. and Smyth, B. On the role of diversity in conversational recommender systems. In *ICCBR*, pages 276–290. 2003b.

McPherson, M. and Smith-Lovin, L. Homophily in voluntary organizations: Status distance and the composition of face-to-face groups. pages 370–379. 1987.

Mika, P. Making things findable: semantics for web search and online media. In *WIMS*, page 3. 2011.

Miller Mcpherson, L. S. L. and Cook, J. M. Birds of a feather: Homophily in social networks. pages 415–444. 2001.

Mislove, A., Viswanath, B., Gummadi, P. K. and Druschel, P. You are who you know: inferring user profiles in online social networks. In *WSDM*, pages 251–260. 2010.

Miyahara, K. and Pazzani, M. J. Collaborative filtering with the simple bayesian classifier. In *PRICAI*, pages 679–689. 2000.

Nachawati, M. O., Rabbi, R., Yu, G. E., Kerschberg, L. and Brodsky, A. Social sifter: An agent-based recommender system to mine the social web. In *STIDS*, pages 125–128. 2012.

Nepal, S., Paris, C. and Bista, S. K. Srec: a social behaviour based recommender for online communities. In *UMAP Workshops*. 2012.

O'Connor, M., Cosley, D., Konstan, J. A. and Riedl, J. Polylens: a recommender system for groups of users. In *ECSCW'01: Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work*, pages 199–218. Kluwer Academic Publishers, Norwell, MA, USA, 2001. ISBN 0-7923-7162-3.

O'Donovan, J. and Smyth, B. Trust in recommender systems. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pages 167–174. ACM, New York, NY, USA, 2005. ISBN 1-58113-894-6.

Paul T. Costa, J. and McCrae, R. R. Domains and facets: Hierarchical personality assessment using the revised neo personality inventory. *Journal of Personality Assesment*, vol. 64(1), pages 21–50, 1995.

Pazzani, M. J. and Billsus, D. Content-based recommendation systems. In *The Adaptive Web*, pages 325–341. 2007.

Penczynski, S. P. Persuasion: An experimental study of team decision making. *University of Mannheim, Department of Economics*, 2014.

Pera, M. S., Condie, N. and Ng, Y.-K. Personalized book recommendations created by using social media data. In *WISE Workshops*, pages 390–403. 2010.

Pessemier, T. D., Dooms, S. and Martens, L. Comparison of group recommendation algorithms. In *Multimedia Tools an Applications*. 2013.

Pham, M. C., Cao, Y., Klamma, R. and Jarke, M. A clustering approach for collaborative filtering recommendation using social network analysis. *J. UCS*, vol. 17(4), pages 583–604, 2011.

Posse, C. Key lessons learned building recommender systems for large-scale social networks. In *KDD*, page 587. 2012.

Quijano-Sánchez, L. and Bridge, D. Feedback on group recommendations. In *Procs. of the Workshop on Case-Based Reasoning in Social Web Applications (Workshop Programme of the 21st ICCBR)* (ed. by L. Quijano-Sánchez and D. Bridge), pages 17–26. 2013.

Quijano-Sánchez, L., Bridge, D., Díaz-Agudo, B. and Recio-García, J. A. A case-based solution to the cold-start problem in group recommender. In *Procs. of the Best Papers in Sister Conferences Track at the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI* (ed. by F. Rossi), pages 3042–3046. AAAI Press, 2013a.

Quijano-Sánchez, L., Bridge, D. G., Díaz-Agudo, B. and Recio-García, J. A. Case-based aggregation of preferences for group recommenders. In *20th International Conference on Case-Based Reasoning, ICCBR*, pages 327–341. 2012a.

Quijano-Sánchez, L., Bridge, D. G., Díaz-Agudo, B. and Recio-García, J. A. A case-based solution to the cold-start problem in group recommenders. In *20th International Conference on Case-Based Reasoning,ICCBR*, pages 342–356. 2012b.

Quijano-Sánchez, L., Recio-García, J. A. and Díaz-Agudo, B. Social based recommendations to groups. In *Procs. of the 14th UK Workshop on Case-Based Reasoning*, pages 46–57. CMS Press, University of Greenwich, 2009. ISBN 978-1-904521-64-8.

Quijano-Sánchez, L., Recio-García, J. A. and Díaz-Agudo, B. Personality and social trust in group recommendations. In *22nd IEEE International Conference on Tools with Artificial Intelligence,ICTAI*, pages 121–126. 2010.

Quijano-Sánchez, L., Recio-García, J. A. and Díaz-Agudo, B. Group recommendation methods for social network environments. In *3rd Workshop on Recommender Systems and the Social Web, RecSys*, pages 383–384. 2011a.

Quijano-Sánchez, L., Recio-García, J. A. and Díaz-Agudo, B. Happymovie: A facebook application for recommending movies to groups. In *23rd IEEE International Conference on Tools with Artificial Intelligence, ICTAI*, pages 239–244. 2011b.

Quijano-Sánchez, L., Recio-García, J. A. and Díaz-Agudo, B. User satisfaction in long term group recommendations. In *19th International Conference on Case-Based Reasoning, ICCBR*, pages 211–225. 2011c.

Quijano-Sánchez, L., Recio-García, J. A. and Díaz-Agudo, B. Using personality to create alliances in group recommender systems. In *19th International Conference on Case-Based Reasoning, ICCBR*, pages 226–240. 2011d.

Quijano-Sánchez, L., Recio-García, J. A. and Díaz-Agudo, B. A reusable methodology for the instantiation of social recommender systems. In *25th IEEE International Conference on Tools with Artificial Intelligence, ICTAI*, pages 777–782. 2013b.

Quijano-Sánchez, L., Recio-García, J. A. and Díaz-Agudo, B. An architecture for developing group recommender systems enhanced by social elements. *Applied Intelligence*, vol. 40(4), pages 732–748, 2014a.

QUIJANO-SÁNCHEZ, L., RECIO-GARCÍA, J. A. and DÍAZ-AGUDO, B. Development of a group recommender application in a social network. *Special Issue on Knowledge-Bases for Cognitive Infocommmunications, Knowledge-Based Systems*, pages 72–85, 2014b.

QUIJANO-SÁNCHEZ, L., RECIO-GARCÍA, J. A. and DÍAZ-AGUDO, B. A reusable methodology for the instantiation of social recommender systems. *International Journal on Artificial Intelligence Tools*, vol. 23(6), 2014c.

QUIJANO-SÁNCHEZ, L., RECIO-GARCÍA, J. A., DÍAZ-AGUDO, B. and JIMÉNEZ-DÍAZ, G. Happy movie: A group recommender application in facebook. In *FLAIRS Conference*, pages 419–420. 2011e.

QUIJANO-SÁNCHEZ, L., RECIO-GARCÍA, J. A., DÍAZ-AGUDO, B. and JIMÉNEZ-DÍAZ, G. Social factors in group recommender systems. *ACM TIST*, vol. 4(1), page 8, 2013c.

QUORA. Why does the startup idea of social recommendations consistently fail? In *http://www.quora.com/Why-does-the-startup-idea-of-social-recommendationsconsistently-fail*. 2012.

RECIO-GARCÍA, J. A., DÍAZ-AGUDO, B., GONZÁLEZ-SANZ, S. and QUIJANO-SÁNCHEZ, L. Distributed deliberative recommender systems. *T. Computational Collective Intelligence*, vol. 1, pages 121–142, 2010.

RECIO-GARCÍA, J. A., GONZÁLEZ-CALERO, P. A. and DÍAZ-AGUDO, B. Template-based design in COLIBRI studio. *Inf. Syst.*, vol. 40, pages 168–178, 2014.

RECIO-GARCÍA, J. A., JIMENEZ-DIAZ, G., SÁNCHEZ-RUIZ, A. A. and DÍAZ-AGUDO, B. Personality aware recommendations to groups. In *Procs. of the 2009 ACM Conference on Recommender Systems*, pages 325–328. ACM, 2009.

RECIO-GARCÍA, J. A., QUIJANO-SÁNCHEZ, L. and DÍAZ-AGUDO, B. Including social factors in an argumentative model for group decision support systems. *DSS*, vol. 56, pages 48–55, 2013.

RESNICK, P., IACOVOU, N., SUCHAK, M., BERGSTROM, P. and RIEDL, J. Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW*, pages 175–186. 1994.

RICCI, F., ROKACH, L., SHAPIRA, B. and KANTOR, P. B., editors. *Recommender Systems Handbook*. Springer, 2011. ISBN 978-0-387-85819-7.

ROGERS, E. *Diffusion of Innovations*. N.Y., Free Press, 1983.

RUTHERFORD, F. J. and AHLGREN, A. *Science for All Americans*, chapter Chapter 7: Human Society. Oxford University Press, 1991. ISBN 0195067711.

SALAKHUTDINOV, R. and MNIH, A. Probabilistic matrix factorization. In *NIPS*. 2007.

SALAKHUTDINOV, R. and MNIH, A. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *ICML*, pages 880–887. 2008.

SARWAR, B. M., KARYPIS, G., KONSTAN, J. A. and RIEDL, J. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295. 2001.

SCHAFER, J., FRANKOWSKI, D., HERLOCKER, J. and SEN, S. *Collaborative Filtering Recommender Systems*, vol. 4321, chapter 9, pages 291–324. Heidelberg S.B, 2007a.

SCHAFER, J. B., FRANKOWSKI, D., HERLOCKER, J. and SEN, S. Collaborative filtering recommender systems. In *The Adaptive Web, Methods and Strategies of Web Personalization* (ed. by P. Brusilovsky, A. Kobsa and W. Nejdl), Lecture Notes in Computer Science 4321, pages 291–324. Springer, 2007b.

SHI, Y., KARATZOGLOU, A., BALTRUNAS, L., LARSON, M., OLIVER, N. and HANJALIC, A. Climf: learning to maximize reciprocal rank with collaborative less-is-more filtering. In *RecSys*, pages 139–146. 2012.

SHIMAZU, H. Expertclerk: Navigating shoppers buying process with the combination of asking and proposing. In *IJCAI*, pages 1443–1450. 2001.

SHIMAZU, H. ExpertClerk: A conversational case-based reasoning tool for developing salesclerk agents in e-commerce webshops. *Artificial Intelligence Review*, vol. 18(3–4), pages 223–244, 2002.

SINHA, R. R. and SWEARINGEN, K. Comparing recommendations made by online systems and friends. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*. 2001.

SMYTH, B. Case-based recommendation. In *The Adaptive Web*, pages 342–376. 2007.

TAVAKOLIFARD, M., ALMEROTH, K. C. and GULLA, J. A. Does social contact matter?: modelling the hidden web of trust underlying twitter. In *WWW (Companion Volume)*, pages 981–988. 2013.

THOMAS, K. and KILMANN, R. *Thomas-Kilmann Conflict Mode Instrument*. Tuxedo, N.Y., 1974.

TINTAREV, N. Explanations of recommendations. In *RecSys*, pages 203–206. 2007.

TINTAREV, N. *Explaining recommendations*. PhD Thesis, 2009.

TODOROV, A., CHAIKEN, S. and HENDERSON, M. D. *The Persuasion Handbook: Developments in Theory and Practice*, chapter The Heuristic-Systematic Model of Social Information Processing, pages 196–211. SAGE Publications, Inc., 0 edition, 2002.

TOMLINSON, S. Comparing the robustness of expansion techniques and retrieval measures. In *CLEF*, pages 129–136. 2006.

UNGAR, L. H. and FOSTER, D. P. Clustering methods for collaborative filtering. In *AAAI Workshop on Recommendation Systems*. 1998.

VAZ, P. C., DE MATOS, D. M., MARTINS, B. and CALADO, P. Improving an hybrid literary book recommendation system through author ranking. *CoRR*, vol. abs/1203.5324, 2012.

VICTOR, P., CORNELIS, C., COCK, M. D. and TEREDESAI, A. Key figure impact in trust-enhanced recommender systems. *AI Commun.*, vol. 21(2-3), pages 127–143, 2008.

VUCETIC, S. and OBRADOVIC, Z. Collaborative filtering using a regression-based approach. *Knowl. Inf. Syst.*, vol. 7(1), pages 1–22, 2005.

WANG, J., DE VRIES, A. P. and REINDERS, M. J. T. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR*, pages 501–508. 2006.

WANGA, L., DOUCET, L. and NORTHCRAFT, G. Culture, affect, and social influence in decision-making groups. In *National Culture and Groups (Research on Managing Groups and Teams, Volume 9)*, pages 147–172. Emerald Group Publishing Limited, 2006.

WU, A., DIMICCO, J. M. and MILLEN, D. R. Detecting professional versus personal closeness using an enterprise social network site. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 1955–1964. ACM, New York, NY, USA, 2010. ISBN 978-1-60558-929-9.

YAHJA, A. and CARLEY, K. M. Wizer: An automated intelligent tool for model improvement of multi-agent social-network systems. In *FLAIRS Conference*, pages 44–50. 2005.

YANG, X., STECK, H., GUO, Y. and LIU, Y. On top-k recommendation using social networks. In *RecSys*, pages 67–74. 2012.

ZHANG, F., BAI, L. and GAO, F. A user trust-based collaborative filtering recommendation algorithm. In *ICICS*, pages 411–424. 2009.

ZHAO, Z.-D. and SHANG, M.-S. User-based collaborative-filtering recommendation algorithms on hadoop. In *WKDD*, pages 478–481. 2010.

ZIEGLER, C.-N. and GOLBECK, J. Investigating interactions of trust and interest similarity. *Decision Support Systems*, vol. 43(2), pages 460–475, 2007.

ZIMMERMAN, J., KURAPATI, K., BUCZAK, A. L., SCHAFFER, D., GUTTA, S. and MARTINO, J. In *Personalized Digital Television. Targeting Programs to Individual Viewers, volume 6 of Human-Computer Interaction Series, chapter 2*, pages 27–51. Kluwer Academic Publishers, 2004.