Sistemas Informáticos
Curso 04-05

FORT: Una herramienta de regresión borrosa

Raquel Ballester Lorenzo
Jose Ignacio del Campo Montejo
Gonzalo Flórez Puga

Dirigido por:
Prof. F.Javier Crespo Yañez
Dpto. Sistemas Informáticos y Programación

Facultad de Informática
Universidad Complutense de Madrid

Índice

1. Resumen

El uso de las técnicas de regresión sobre las observaciones experimentales ha permitido el estudio de numerosos fenómenos en diversos campos de la ciencia, y muy especialmente en las ciencias sociales. Dichas técnicas requieren de un número suficiente de observaciones “precisas”, exactas y fiables. Sin embargo, no siempre es posible obtener el conjunto de observaciones necesario, o éstas contienen algún tipo de imperfección en los datos, debido a la imprecisión o vaguedad de los mismos.

En cualquier caso, con suficientes datos o no, con imperfecciones o no, los modelos obtenidos deberían proveer de capacidades predictivas y descriptivas [JCr02]. Las actuales herramientas, o las más fácilmente accesibles, tienen limitado el uso de modelos y difícilmente usan las técnicas de la teoría de conjuntos borrosos.

Se propone en este trabajo una herramienta abierta de regresión que admita el uso de cualquier modelo de curva independientemente de su naturaleza. Además, esta herramienta permitirá el uso de diferentes formas de borrosidad y por su diseño permitiría cualquier modelo propuesto por el usuario si éste prevee que éstos tienen características que sean suficientemente predictivas y descriptivas.

Esta primera aproximación de una herramienta abierta de regresión se realiza un estudio sobre diferentes modelos paramétricos simbólicos, usados comúnmente en la práctica en disciplinas tan heterogéneas como pueden ser la Ingeniería del Software, la Economía o en cualquier campo en donde puedan aparecer imprecisiones en la información.

Abstract

The use of regression techniques in experimental observations has led to the study of numerous phenomena in various fields of science, especially in social science. These techniques require a sufficient number of “precise”, exact and reliable observations. However, it is not always possible to obtain all the necessary group of observations or these have some failings, as a result of inexact or vague data.

Nevertheless, having more or less data, with or without failings, the obtained paradigms should provide predictive and descriptive capacities. The current tools or those more accessible have limited paradigm application and hardly use the techniques relating the fuzzy sets theory.

In this first approach to an open regression tool, a study has been carried out of the different parametric, symbolic paradigms, commonly used in the practice of such diverse disciplines as Software Engineering, Economy or any other field where information imprecision can appear.

Palabras clave

Regresión borrosa, Ajuste de curvas, Lógica borrosa, modelos matemáticos simbólicos, imprecisión, herramienta abierta

2. Agradecimientos

En primer lugar, queremos agradecer la ayuda y comprensión de nuestras familias, que han vivido muy de cerca tanto los buenos momentos como los malos, porque gracias a su apoyo incondicional, hemos podido llegar hasta aquí.

Agradecemos a nuestro profesor Javier Crespo, que no sólo ha dirigido el proyecto con eficacia, sino que nos entregado su tiempo, su conocimiento, su paciencia, y sobre todo, agradecemos la confianza que ha depositado en nosotros.

También agradecemos a los profesores Luis Garmendia y Gonzalo Méndez, que han ayudado y colaborado en la elaboración de este trabajo.

Finalmente, agradecemos a todos nuestros compañeros y demás profesores, que nos han acompañado en este largo camino, y muy especialmente a nuestros amigos, que nos han aconsejado y animado.

Muchas gracias a todos.

3. Autorización

Autorizamos a la Universidad Complutense a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Raquel Ballester Lorenzo	Jose I. del Campo Montejo	Gonzalo Flórez Puga
raquelballester@terra.es	campo_ji@yahoo.es	gflorezpuga@yahoo.es

4. Introducción

4.1. Planteamiento del problema

El hombre tiene la capacidad de percibir el mundo que le rodea y expresarlo mediante el lenguaje natural, dicho lenguaje es una forma de comunicación imprecisa y ambigua que se apoya en el conocimiento compartido por aquellos con los que se comunica [Jlaw01].

En el lenguaje natural se describen objetos o situaciones en términos imprecisos: grande, joven, tímido, alto, bonito,... La representación del conocimiento basado en estos términos no puede ser exacta, quizá impreciso, pero no exacta, ya que normalmente representan impresiones subjetivas o percepciones, tal y como analiza Lofti A. Zadeh, profesor de la Universidad de California en Berkeley, en su Teoría computacional de la percepción [Zad01], en la cual trata un proceso de razonamiento automatizado en donde se opera sobre la base de percepciones en lugar de sobre medidas.

Para aclarar conceptos, se muestran algunas definiciones[SME99]:

• **Información incierta**: Aquella que no se conoce o de la que no puede determinarse su veracidad o falsedad.

Por ejemplo, en la sentencia “La mujer de Pedro es María”. La información es completa y precisa, pero incierta, ya que la persona que nos lo ha comentado puede estar equivocada.

Otro ejemplo: “Juan tiene 2 hijos pero no estoy muy seguro.” Es precisa pero incierta.

• **Información imprecisa**: Aquella que se refiere a una variable cuyo valor no se conoce o siendo conocido no puede determinarse con precisión, existe ambigüedad. La información está libre de errores, por lo que no se puede calificar de incierta.

Ejemplo: “¿Tose mucho el paciente?” , o “¿Qué temperatura tiene el paciente?”.

“Juan tiene al meno 2 hijos y estoy seguro de esto”.Esta información es imprecisa pero cierta.

• **Información incompleta**: Aquella en la que se desconoce algún dato significativo.

La imprecisión y la incompletitud son objetivas, aunque dependientes del contexto.

Existen independientemente del observador, son propiedades de los "datos", de la información. Son proposiciones verdaderas o falsas, variables con un valor, pero el interlocutor no lo sabe:

Calificar la información de incierta conlleva subjetividad, es el interlocutor el que no tiene certeza acerca de la información disponible. Esta información sólo induce un conocimiento parcial, tiene asociado un grado de creencia en el interlocutor.

La sentencia "¿Es muy mayor?" induce algo subjetivo, aunque el agente sepa que tiene 60 años.

"¿Juan tose mucho?" . ¿Cuantificación de "tose"?

Entre los modelos numéricos propuestos para enfrentarse a estos diversos problemas que se han mostrado, uno de los más empleados es la teoría de conjuntos borrosos.

La borrosidad es la propiedad asociada al uso de predicados inciertos, como "Juan es alto". Se fundamenta en la pertenencia de un elemento a un conjunto, un grado que no tiene porqué ser necesariamente '0' o '1' como en el caso de la teoría de conjuntos clásica.

Los valores intermedios son admitidos para hacer frente a estos otros casos.

La borrosidad nos aclara acerca de la imprecisión derivada de la pertenencia parcial de un elemento dado y bien definido a un conjunto cuyos extremos no están definidos de una forma clara.

Diversos campos de la Ciencia como la Agricultura, Química, Medicina, Medio Ambiente, Psicología, Biología, Economía... usan modelos lineales para su desarrollo, lo que ha supuesto un gran avance, no solo por los desarrollos matemáticos alcanzados, sino también por su aplicación en situaciones reales. Sin embargo, estas relaciones no son de utilidad a la hora de usar variables lingüísticas, ya que los modelos lineales se basan en datos obtenidos por medio de la planificación y diseño de experimentos.

4.2. Necesidad de una herramienta de regresión borrosa

Los modelos paramétricos matemáticos usados en la actualidad trabajan bajo el dominio de los enteros, obviando las imprecisiones en los datos, por ello, si el objetivo es conseguir resultados más representativos, se hace necesario usar modelos paramétricos matemáticos simbólicos, que sean más predictivos y descriptivos [JCr02].

En los modelos en donde los datos sean insuficientes o imperfectos, originados por la imprecisión o vaguedad, se ha demostrado útil el uso de un análisis borroso [ChA01], [IKW01], [KlF88], [Sug85], [NPr99], [Izy], [Twa99], [TSK82].

El análisis de regresión borroso ha sido estudiado y aplicado en diferentes áreas tal como el modelado de datos económicos o financieros, la ingeniería del software... [CDS86], y se han obtenido resultados comparables y que aportan ajustes, en muchos casos superiores, a los obtenidos con el análisis de regresión clásico. Sin embargo, resulta extraño la dificultad de encontrar herramientas que permitan como entrada diferentes formas de borrosidad, con el fin de obtener una mayor descripción de la información que se pretende representar.

4.3. Objetivos

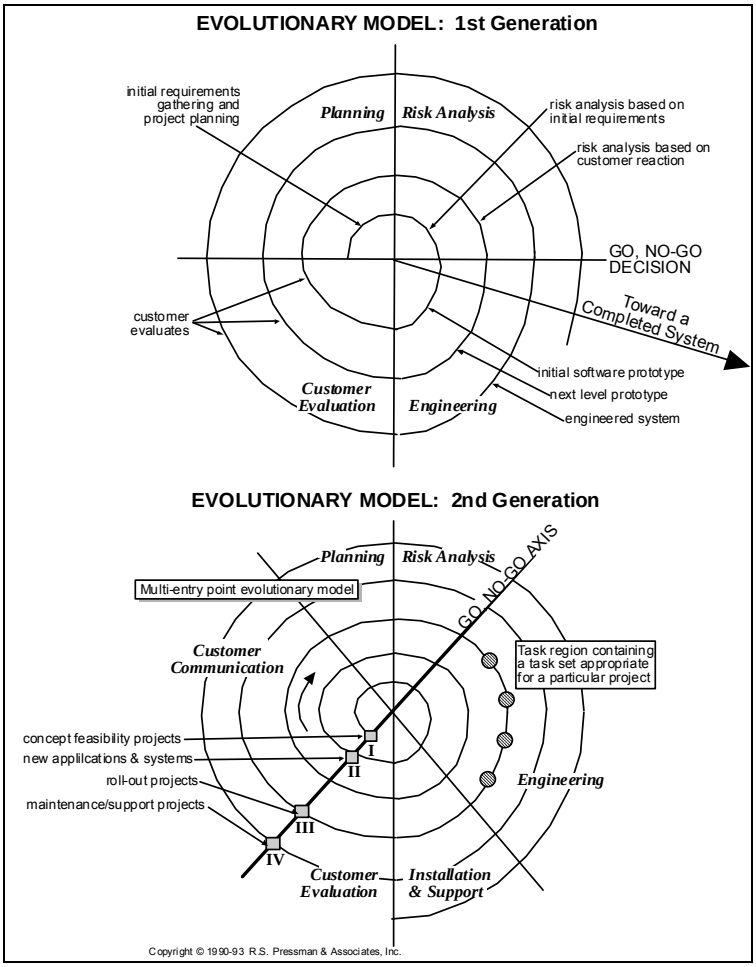
El objetivo del proyecto es desarrollar una herramienta abierta mediante regresión nítida y borrosa, haciendo uso de las técnicas de la Teoría de Conjuntos Borrosos, con el fin de obtener un mejor reflejo de la realidad.

La idea principal del proyecto es incrementar la complejidad de los modelos paramétricos existentes, usando los conjuntos borrosos en la regresión, para conseguir una herramienta capaz de modelar el mundo real de forma más aproximada.

El desarrollo de la herramienta implica ver los sistemas existentes en la actualidad, investigar las aplicaciones de los conjuntos borrosos a la regresión, construir e implementar el modelo de la herramienta, realizar una primera aproximación y obtener las conclusiones del proyecto.

4.4. Metodología

En el desarrollo de la herramienta de regresión borrosa seguiremos el modelo en espiral de segunda generación [PRE93].



Este modelo se centra en tratar las áreas de mayor riesgo en un proyecto (requisitos, arquitectura, etc.). El proceso de diseño del producto en espiral es iterativo e incremental [Boe81], por lo que permite una mayor creatividad y facilita la realización de cambios según se avanza. El proyecto se compone de múltiples iteraciones sobre varias regiones de tareas, cada una de estas regiones está poblada por una serie de tareas que se adaptan a las características del proyecto que va a emprenderse. Con cada iteración alrededor de la espiral (comenzando en el centro y siguiendo hacia el exterior), se construyen sucesivas versiones del software, cada vez más completas.

El proceso de diseño de productos en espiral utilizando el modelo de Boston consta de 6 fases:

- **Comunicación con el cliente**

  Tareas requeridas para establecer comunicación entre el desarrollador y el cliente

- **Planificación**

  Tareas requeridas para definir recursos, tiempo y otras informaciones relacionadas con el proyecto

- **Análisis de riesgos**

  Tareas requeridas para evaluar riesgos técnicos y de gestión

- **Ingeniería**

  Tareas requeridas para construir una o más representaciones de la aplicación

- **Construcción y adaptación**

  Tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario

- **Evaluación por el cliente**

  Tareas requeridas para obtener la reacción del cliente tras su evaluación

Algunas de las ventajas que este modelo aporta al desarrollo de nuestra herramienta son:

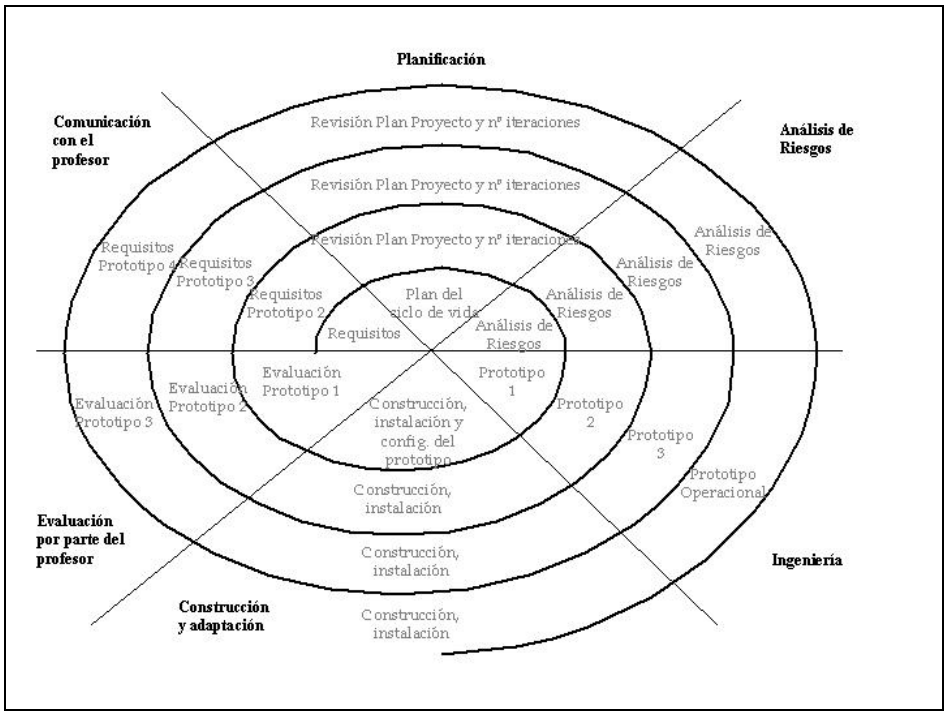
- Gestión explícita de riesgos, algo bastante importante dada la dificultad que conlleva el internarse en un nuevo campo de conocimiento que ninguno de los miembros del proyecto controlaba en profundidad.

- Centra su atención en la reutilización de componentes y eliminación de errores en información descubierta en fases iniciales, otra gran ventaja dado que uno de las principales cualidades de nuestra herramienta es su carácter abierto.

- Enfoque realista, que además permitirá ir desarrollando progresivamente los conocimientos adquiridos.

- Integra desarrollo con mantenimiento, lo que hace que esta metodología sea adecuada para futuros desarrollos de posteriores grupos de Sistemas Informáticos.

A continuación se explican las tareas realizadas en cada una de las iteraciones:



Primera iteración:

- Reunión con el profesor director del proyecto, para definir la herramienta que se va a desarrollar.

- Planificación, para definir el tiempo, esfuerzo, recursos y disponibilidad de cada uno de los miembros del grupo.

- Análisis de riesgos, para evaluar los riesgos técnicos y de gestión. Estudio de la arquitectura a emplear para la posible implementación del sistema.

- Ingeniería, se definen las clases que se van a utilizar y se crean los diagramas de clases para el modelo nítido.

- Construcción y adaptación, se implementan las clases de los diagramas y se construye una herramienta que resuelve el problema de forma analítica, obteniendo así el modelo nítido.

- Evaluación del profesor director del proyecto de las representaciones creadas en las tareas anteriores.

Segunda iteración:

- Reunión con el profesor director del proyecto, para definir las principales funciones de la herramienta para crear un modelo lineal.
- Planificación, para definir el tiempo, esfuerzo, recursos y disponibilidad de cada uno de los miembros del grupo.
- Análisis de riesgos, para evaluar los riesgos técnicos y de gestión.
- Ingeniería, se define un prototipo en el que se crean modelos lineales, teniendo las entradas nítidas.
- Construcción y adaptación, se implementa el prototipo.
- Evaluación del profesor director del proyecto del prototipo creado anteriormente.

Tercera iteración:

- Reunión con el profesor director del proyecto, para definir las funciones de la herramienta cuando se pretende crear modelos no lineales.
- Planificación, para definir el tiempo, esfuerzo, recursos y disponibilidad de cada uno de los miembros del grupo.
- Análisis de riesgos, para evaluar los riesgos técnicos y de gestión.
- Ingeniería, se define un prototipo en el que se permite cualquier modelo de curva.
- Construcción y adaptación, se construye el prototipo de la herramienta.
- Evaluación del profesor director del proyecto del prototipo creado anteriormente.

Cuarta iteración:

- Reunión con el profesor director del proyecto, para definir las funciones que permitan añadir la borrosidad a la herramienta.
- Planificación, para definir el tiempo, esfuerzo, recursos y disponibilidad de cada uno de los miembros del grupo.

- Análisis de riesgos, para evaluar los riesgos técnicos y de gestión.
- Ingeniería, se define el prototipo en el que se permite la borrosidad de sus elementos.
- Construcción y adaptación, se construye el prototipo definido, y se completa la interfaz.
- Evaluación del profesor director del proyecto del prototipo creado anteriormente y obtención de las conclusiones.

4.4.1. Gestión de los riesgos del proyecto

Este apartado se centra en el análisis de los posibles riesgos que pudieron presentarse durante el proceso de desarrollo del proyecto. Los riesgos pudieron incidir en un posible retraso en la entrega e incluso dar lugar a la no finalización del proyecto, por lo que se procede a elaborar una lista detallada de los riesgos que se pudieron presentar con mayor probabilidad y un análisis de las posibles resoluciones a los problemas planteados.

En primer lugar se procede a la **identificación de riesgos**, agrupándolos según criterios en común. En el proyecto se encuentran principalmente cuatro grupos fundamentales de riesgos:

a) Riesgos tecnológicos: aquellos relacionados con el soporte tecnológico y la arquitectura empleados así como el rendimiento de sus componentes.

b) Riesgos de personal: todos aquellos relacionados con los miembros que participan en el desarrollo de la aplicación.

c) Riesgos debidos a la organización: relativos a los cambios de herramientas de desarrollo.

d) Riesgos de estimación: La estimación del tiempo a emplear en el desarrollo de cada funcionalidad de la aplicación no fue el adecuado.

Seguimiento: a continuación se describen todos los riesgos y se plantean las posibles soluciones, ya sean preventivas, para impedir que se den lugar, o resoluciones en el caso de producirse.

a) Riesgos tecnológicos:

Nombre	Descripción	Resolución	Puntuación
Riesgos de la arquitectura seleccionada	La selección de la arquitectura para el proyecto no es la	Se hace un estudio comparativo previo identificando ventajas y desventajas de todas las	Baja

	más adecuada	arquitecturas sobre las que permita implementar el proyecto. En el transcurso de la primera iteración se trata este punto en la fase de riesgos, donde se discutieron diversas posibilidades, como una arquitectura. Net realizando el desarrollo en C#, o un desarrollo en C++. Finalmente se escogió realizar el proyecto utilizando lenguaje Java, al ser bastante conocido por los miembros del grupo al tiempo que posee características suficientemente potentes para las pretensiones del proyecto. Adicionalmente posee un API muy completo y documentado, así como una comunidad de desarrolladores muy activa que puede ser de gran ayuda. Para el desarrollo se decide emplear JBuilder X de Borland.	

b) Riesgos de personal:

Nombre	Descripción	Resolución	Puntuación
Falta de conocimientos y experiencia	Escasa experiencia y conocimiento por parte del personal en algunas de las	Elección de la plataforma de desarrollo más adecuada a los	Baja

	tecnologías empleadas para la realización del proyecto.	conocimientos del grupo. Uso de un repositorio de documentación para compartir el conocimiento que se vaya adquiriendo y consulta de otras aplicaciones similares.	
Enfermedades	Posibilidad de ausencia de algún componente del grupo en un periodo significativo de tiempo, pudiendo ocasionar, si los demás no conocen la parte asignada al ausente, un retraso importante.	Mayor dedicación por parte del resto de miembros en el desarrollo del proyecto para optimizar el tiempo de desarrollo.	Baja
Abandonos	Posibilidad de retiro de algún miembro del equipo de desarrollo, provocando, si los demás no conocían su trabajo, una prolongación importante en el tiempo de desarrollo.	Aprendizaje por parte de los miembros restantes de los aspectos con los que trabajaba el que se retiró. Incorporar a un nuevo miembro que conozca lo necesario para sustituir al que abandonó. Mayor dedicación por parte de todos los que perduran	Baja
Mal ambiente de trabajo.	Mala relación entre miembros por diferencia de caracteres y dificultad de entendimiento.	Potenciar las relaciones personales fuera del ambiente del trabajo. División de tareas indicando de manera clara los aspectos a desarrollar y permitir una alta modularidad, trabajando en paralelo siempre que sea posible.	Baja

Prototipo de grupo de trabajo.	Inestabilidad en el grupo de trabajo	Se trata de un grupo de trabajo de estudiantes sin contraprestación económica, el proyecto se realiza de forma altruista, por lo que no es de esperar problemas en este sentido.	Baja

c) Riesgos debidos a la organización:

Nombre	Descripción	Resolución	Puntuación
Cambio de herramientas	La organización del proyecto decide cambiar la herramienta de desarrollo	Migrar la aplicación a la nueva herramienta, y en caso de desconocerla se tendrá que llevar a cabo un autoaprendizaje.	Baja
No disponibilidad de puestos para el desarrollo de la aplicación	Ocupación excesiva de los laboratorios que impida el acceso, sobre todo en épocas de exámenes	Búsqueda de lugares alternativos.	Media
Falta de herramientas necesarias	Las herramientas elegidas para el desarrollo pueden no estar instaladas en los laboratorios	Petición de su instalación a los técnicos o cambio de herramienta.	Media

d) Riesgos de estimación:

Nombre	Descripción	Resolución	Puntuación
Falta de	El tiempo para el desarrollo del	El objetivo inicial de la realización de la herramienta	Alta

tiempo	proyecto propuesto es insuficiente.	completa queda fuera del alcance del proyecto, ya que en la fase de planificación se estimó que requería un tiempo de realización superior al disponible. Se definirá el proceso y las bases del desarrollo para modelizar la herramienta de forma que su carácter abierto permita llegar a los objetivos iniciales en posteriores versiones de la misma. Se irán haciendo evaluaciones periódicas del rendimiento del equipo de trabajo y, en caso de falta de tiempo, se llevarán a cabo nuevas estimaciones. Se ajustarán los horarios individuales para tratar de coincidir a determinadas horas. Se establecerá un plan de comunicación frecuente a través de teléfono, mail...	
No disponibilidad de recursos	Los recursos empleados en la aplicación no se encuentran disponibles	Se realizará una nueva estimación del tiempo en cuanto a los recursos que sí están disponibles y su reparto entre las tareas, aumentando el esfuerzo.	Media.

A continuación se establece un orden de prioridad en los riesgos del proyecto en base a su probabilidad de ocurrencia.

- Falta de tiempo: probabilidad alta.
- No disponibilidad de recursos: probabilidad media.
- No disponibilidad de puestos para el desarrollo de la aplicación: probabilidad media.
- Falta de herramientas necesarias: probabilidad media.

- Cambio de herramientas: probabilidad baja.
- Mala relación entre miembros: probabilidad baja.
- Riesgos de la arquitectura seleccionada: probabilidad baja.
- Enfermedades: probabilidad baja.
- Abandonos: probabilidad baja.
- Mal funcionamiento de la tecnología: probabilidad baja.
- Falta de conocimientos y experiencia: probabilidad baja.

5. Estado de la Cuestión

5.1. Introducción a la teoría de los conjuntos difusos

La mayoría de los fenómenos que encontramos cada día son imprecisos, es decir, tienen implícito un cierto grado de difusidad en la descripción de su naturaleza. Esta imprecisión puede estar asociada con su forma, posición, momento, color, textura, o incluso en la semántica que describe lo que son. En muchos casos el mismo concepto puede tener diferentes grados de imprecisión en diferentes contextos o tiempo. Un día cálido en invierno no es exactamente lo mismo que un día cálido en primavera. La definición exacta de cuando la temperatura va de templada a caliente es imprecisa -no podemos identificar un punto simple de templado, así que emigramos a un simple grado, la temperatura es ahora considerada caliente. Este tipo de imprecisión o difusidad asociado continuamente a los fenómenos es común en todos los campos de estudio: sociología, física, biología, finanzas, ingeniería, oceanografía, psicología, etc. [Calv03].

Los límites de los conjuntos difusos no están perfectamente definidos, es decir, la transición entre la pertenencia y la no-pertenencia de una variable a un conjunto es gradual.

La teoría de los conjuntos difusos, se caracteriza por las funciones de pertenencia, que da flexibilidad a la modelización utilizando expresiones lingüísticas, tales como mucho, poco, leve, severo, escaso, suficiente, caliente, frío, joven, viejo, etc. Surgió de la necesidad problemas complejos con información imprecisa, para los cuales la matemática y lógica tradicionales no son suficientes. La lógica difusa es un lenguaje que permite trasladar sentencias sofisticadas del lenguaje natural a un formalismo matemático [ACIS3].

La lógica difusa fue inventada en 1960 por Lotfi Zadeh, guiado por el principio de que las matemáticas pueden ser usadas para encadenar el lenguaje con la inteligencia humana. Algunos conceptos pueden ser mejor definidos con palabras, los conjuntos difusos ayudan a construir mejor la realidad [ACIS3].

5.1.1. Definición de conjunto difuso

Sea U un universo de discusión con su elemento genérico denotado por u. Luego un subconjunto difuso A de U está caracterizado por una función de pertenencia:

$$\mu_A: U \longrightarrow [0,1]$$

que asocia a cada elemento u de U un número $\mu_A(u)$ que representa el grado de pertenencia de u en A. A se denota como el conjunto de pares ordenados $\{\mu_A(u), u\}$.

Es decir un conjunto difuso A se considera como un conjunto de pares ordenados, en los que el primer componente es un número en el rango [0,1] que denota el grado de pertenencia de un elemento u de U en A, y el segundo componente especifica precisamente quién es ése elemento de u.

En general los grados de pertenencia son subjetivos en el sentido de que su especificación es una cuestión objetiva. Se debe aclarar que aunque $\mu_A(u)$ puede interpretarse como el grado de verdad de que la expresión "u ∈A" sea cierta, es más natural considerarlo simplemente como un grado de pertenencia.

Puede notarse además que:

a) Mientras más próximo está $\mu_A(u)$ al valor 1, se dice que u pertenece más a A (de modo que 0 y 1 denotan la no pertenencia y la pertenencia completa, respectivamente).

b) Un conjunto en el sentido usual es también difuso pues su función característica

$$\mu_A : u \to \begin{cases} 0 \text{ si } \mu \in A \\ 1 \text{ si } \mu \notin A \end{cases}$$

es también una función μ_A:u⟶ [0,1]; o sea que los conjuntos difusos son una generalización de los conjuntos usuales [ACIS3].

5.1.2. Operaciones de los conjuntos difusos

Las operaciones básicas de los conjuntos difusos son:

- Contención o subconjunto:

  A es un subconjunto de B si y solo si $\mu_A(x) \leq \mu_B(x)$, para todo x

  $A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x)$

- Unión:

  La unión de los conjuntos difusos A y B es el conjunto difuso C, y se escribe como C= AOR B, su función de dependencia está dada por

  $\mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) - \mu_B(x)$

- Intersección:

  La intersección de los conjuntos difusos A y B es el conjunto difuso C, y se escribe como C=A AND B, su función de dependencia está dada por $\mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \cdot \mu_B(x)$

- Complemento (negación):

El complemento del conjunto difuso A, denotado por Ā (¬A, NOT A), se define como

$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$

- Producto Cartesiano:

  Si A y B son conjuntos difusos en X e Y, el producto cartesiano de los conjuntos A y B A x B en el espacio de X x Y tiene la función de pertenencia

  $\mu_{AxB}(x,y) = \min(\mu_A(x), \mu_B(x))$

- Co-producto Cartesiano

  A + B en el espacio X x Y tiene la función de pertenencia

  $\mu_{AxB}(x,y) = \max(\mu_A(x), \mu_B(x))$

5.1.3. Aplicaciones y Soft Computing

Con formato: Numeración y viñetas

La teoría de conjuntos difusos ha sido ampliamente aplicada en campos como: Medicina, Economía, Ecología, Biología,... Se ha empleado en empresas de producción de artículos eléctricos y electrónicos como una herramienta de control, se ha utilizado para el desarrollo de procesadores y computadoras.

Los conjuntos difusos son usados para toma de decisiones y estimaciones de Sistemas de Control como son: aire acondicionado, control de automóviles y controladores en sistemas industriales.

En general, la lógica difusa es aplicada en cualquier campo donde sea muy difícil o casi imposible crear un modelo, en sistemas controlados por expertos humanos, en sistemas donde se tienen entradas y salidas que son continuas y complejas, en sistemas que utilizan observaciones humanas como entradas o reglas básicas, y en cualquier sistema en el cual se trabaje con conceptos imprecisos [ACIS3].

Las técnicas de Soft Computing engloban básicamente, la lógica borrosa, las redes neuronales, la computación evolutiva, los algoritmos genéticos y el razonamiento probabilístico.

El Soft Computing desempeña un papel muy importante en las ciencias y en la ingeniería, aunque su aplicación se extenderá en otros muchos campos, debido a los resultados satisfactorios que se han ido obteniendo con su uso,

Las técnicas de Soft Computing destacan por su tolerancia a la imprecisión, la incertidumbre, grado de credibilidad y la aproximación.

Estas técnicas usan la mente humana como modelo. Las líneas de investigación que lleva a cabo son:

- Una nueva generación de motores de búsqueda en Internet, que usan técnicas de Soft Computing y tratan de mejorar la búsqueda lexicográfica actual, usando una búsqueda conceptual.

- Técnicas avanzadas para descubrir "perfiles de usuario" que permitan un uso de internet más inteligente "a la carta".

- Comercio electrónico basado en técnicas de Soft Computing, por ejemplo, lo que el profesor Mandani denomina Soft Knowledge.

- Semantic Web.

En los últimos años se ha podido comprobar un rápido crecimiento de las aplicaciones de la lógica borrosa y las redes neuronales, en diversos campos: Electrónica de consumo, control de procesos industriales, reconocimiento del habla, visión artificial, tratamiento de la señal, reconocer y clasificar imágenes, manejar vehículos en tráfico denso y un largo etcétera.

Con formato: Numeración y viñetas

5.1.4. Lógica borrosa y sus aplicaciones

La lógica borrosa es básicamente una lógica multievaluada que permite valores intermedios para poder definir evaluaciones convencionales como sí / no, verdadero / falso, negro / blanco, etc. De esta forma se ha realizado un intento de aplicar una forma más humana de pensar en la programación de computadoras. La lógica borrosa se inició en 1965 por Lotfi A. Zadeh, profesor de ciencia de computadoras en la Universidad de California en Berkeley.

Aunque la lógica borrosa se inventó en Estados Unidos el crecimiento rápido de esta tecnología ha comenzado desde Japón y ahora nuevamente ha alcanzado USA y también Europa.

La intención original del profesor Zadeh era crear un formalismo para manipular de forma más eficiente la imprecisión y vaguedad del razonamiento humano expresado lingüísticamente, pero el éxito de la lógica borrosa llegó en el campo del control automático de procesos. Esto se debió principalmente al "boom" de lo borroso en Japón, iniciado en 1987 y que alcanzó su máximo apogeo a principios de los noventa.

Desde entonces, han sido infinidad los productos lanzados al mercado que usan tecnología borrosa, muchos de ellos utilizando la etiqueta "fuzzy" como símbolo de calidad y prestaciones avanzadas.

En 1974 el profesor Mamdani experimentó con éxito un controlador borroso en una máquina de vapor, pero la primera implantación real de un controlador de este tipo fue realizada en 1980 por F. L. Smidth & Co. en una planta cementera en Dinamarca.

En 1983, Fuji aplica lógica borrosa para el control de inyección química para plantas depuradoras de agua, por primera vez en Japón.

En 1987 la empresa OMRON desarrolla los primeros controladores borrosos comerciales con el profesor Yamakawa. A partir de ese momento, el control borroso ha sido aplicado con éxito en muy diversas ramas tecnológicas, por ejemplo la metalurgia, los robots de fabricación, controles de maniobra de aviones, ascensores o trenes (tren-metro de Sendai, Japón, 1987), sensores, imagen y sonido (sistema de estabilización de imagen en cámaras fotográficas y de video Sony, Sanyo, Canon...), electrodomésticos (lavadoras de Panasonic o Bosch, aire acondicionado Mitsubishi, rice-cooker...), automoción (sistemas de ABS de Mazda o Nissan, Cambio automático de Renault, control automático de velocidad, climatizadores...) y una larga lista de aplicaciones comerciales.

Con formato: Numeración y viñetas

5.1.5. Redes neuronales y sus aplicaciones

Las redes neuronales surgen de los estudios sobre neurofisiología debidos a Rosenblatt. En estos trabajos se busca un modelo matemático para la neurona, de manera que sea posible reproducir por técnicas artificiales la capacidad de interpretación de información del cerebro.

Se considera el cerebro como una computadora capaz de procesar información imprecisa a un ritmo increíblemente veloz, y sobre todo, que aprende sin instrucciones explícitas de ninguna clase a crear las representaciones internas que hacen posible tales habilidades.

La estructura de la red se establece por imitación de las estructuras neuronales, tales como el cerebro. Estas redes deben tener capacidad de aprendizaje, es decir, deben ser capaces de modificar sus conexiones con tal de adaptarse de la mejor forma posible al comportamiento requerido.

En una red neuronal se distinguen dos etapas: una primera de aprendizaje, en la cual son presentadas a la red un conjunto de patrones de entrada y de salida. Por medio de algún algoritmo de optimización se modifican sus conexiones con el fin de que ésta imite este comportamiento. Y una segunda etapa, de funcionamiento, en la cual, ante cualquier entrada, la red debe ser capaz de responder con una salida lo más similar posible a las aprendidas [RNA00].

En general, la aplicación más extendida de las redes neuronales es la clasificación, también conocida como reconocimiento de patrones. La red es empleada como un clasificador, de manera que ante una entrada estima la mayor o menor afinidad de ésta a los patrones aprendidos [DNN93].

Actualmente, las redes neuronales se emplean con éxito en numerosas aplicaciones, tales como el reconocimiento de caracteres escritos, reconocimiento del habla y sistemas de identificación.

Las redes neuronales y la lógica borrosa pueden aportar soluciones muy favorables al proceso de traducción Automática, ya que el problema principal que se presenta a la hora de realizar la traducción de un texto es su adecuada comprensión e interpretación.

Con formato: Numeración y viñetas

5.1.6. Algoritmos genéticos y sus aplicaciones

Un algoritmo genético es una técnica de programación que imita a la evolución biológica como estrategia para resolver problemas.

Dado un problema específico a resolver, la entrada del algoritmo genético es un conjunto de soluciones potenciales a ese problema, codificadas de alguna manera, y una métrica llamada función de aptitud que permite evaluar cuantitativamente a cada candidata. Estas candidatas pueden ser soluciones que ya se sabe que funcionan, con el objetivo de que el algoritmo genético las mejore, pero se suelen generar aleatoriamente.

Seguidamente, el algoritmo genético evalúa cada candidata de acuerdo con la función de aptitud. En un acervo de candidatas generadas aleatoriamente, por supuesto, la mayoría no funcionarán en absoluto, y serán eliminadas. Sin embargo, por puro azar, unas pocas pueden ser prometedoras -pueden mostrar actividad, aunque sólo sea actividad débil e imperfecta, hacia la solución del problema.

Estas candidatas prometedoras se conservan y se les permite reproducirse. Se realizan múltiples copias de ellas, pero las copias no son perfectas; se introducen cambios aleatorios durante el proceso de copia.

Esta descendencia digital prosigue con la siguiente generación, formando un nuevo acervo de soluciones candidatas, y son sometidas a una ronda de evaluación de aptitud. Las candidatas que han empeorado o no han mejorado con los cambios en su código son eliminadas de nuevo; pero, de nuevo, por puro azar, las variaciones aleatorias introducidas en la población pueden haber mejorado a algunos individuos, convirtiéndolos en mejores soluciones del problema, más completas o más eficientes.

De nuevo, se seleccionan y copian estos individuos vencedores hacia la siguiente generación con cambios aleatorios, y el proceso se repite. Las expectativas son que la aptitud media de la población se incrementará en cada ronda y, por tanto, repitiendo este proceso cientos o miles de rondas, pueden descubrirse soluciones muy buenas del problema [PSer96].

Los algoritmos genéticos han demostrado ser una estrategia enormemente poderosa y exitosa para resolver problemas, demostrando de manera espectacular el poder de los principios evolutivos.

Se han utilizado algoritmos genéticos en una amplia variedad de campos para desarrollar soluciones a problemas tan difíciles o más difíciles que los abordados por los diseñadores humanos [RSA95].

Las soluciones que consiguen los algoritmos genéticos son a menudo más eficientes, más elegantes o más complejas que nada que un ingeniero humano produciría.

Existen diversos tipos algoritmos evolutivos que se pueden aplicar a numerosos problemas, dependiendo de su naturaleza, ya sea un problema de búsqueda, de optimización, de predicción o de clasificación (Parametrización, Configuración, Maximización, Minimización).

La Computación Evolutiva puede aplicarse en diversos dominios, por ejemplo, en un proceso de generación de conocimiento interpretable por el humano.

Actualmente, la generación automática de conocimiento es un proceso realizable en una computadora con un mínimo de supervisión, retroalimentación y supuestos.

Existen aplicaciones prácticas donde este proceso automático ha sido de gran ayuda, como en el campo del control de plantas depuradoras de agua, en donde se ha logrado revisar las clases de situaciones de una planta propuestas por un experto humano, recordándole situaciones pasadas que se le olvidó mencionar en una entrevista.

5.2. Regresión, métodos y herramientas

5.2.1. Concepto de regresión

El Análisis de Regresión es una técnica estadística que tiene como objetivo establecer modelos matemáticos para representar formalmente las relaciones de dependencia existente entre un conjunto de variables estadísticas.

Tanto en el caso de dos variables (regresión simple) como en el de más de dos variables (regresión múltiple), el análisis de regresión lineal puede utilizarse para explorar y cuantificar la relación entre una variable llamada dependiente o criterio (Y) y una o más variables llamadas independientes o predictoras (X1, X2, ..., Xk), así como para desarrollar una ecuación lineal con fines predictivos. Además, el análisis de regresión lleva asociados una serie de procedimientos de diagnóstico (análisis de los residuos, puntos de influencia) que informan sobre la estabilidad e idoneidad del análisis y que proporcionan pistas sobre cómo perfeccionarlo.

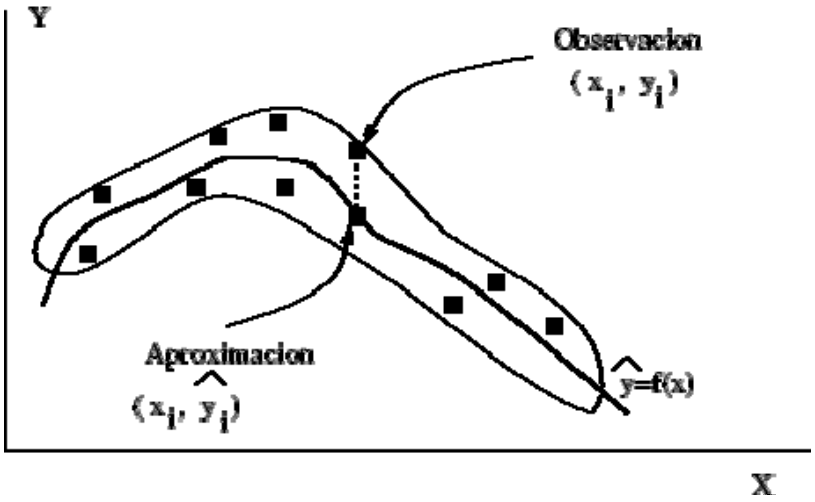
Se adapta a una amplia variedad de situaciones. En la investigación social, el análisis de regresión se utiliza para predecir un amplio rango de fenómenos, desde medidas económicas hasta diferentes aspectos del comportamiento humano. En el contexto de la investigación de mercados puede utilizarse para determinar en cuál de diferentes medios de comunicación puede resultar más eficaz invertir; o para predecir el número de ventas de un determinado producto. En física se utiliza para caracterizar la relación entre variables o para calibrar medidas, etc.

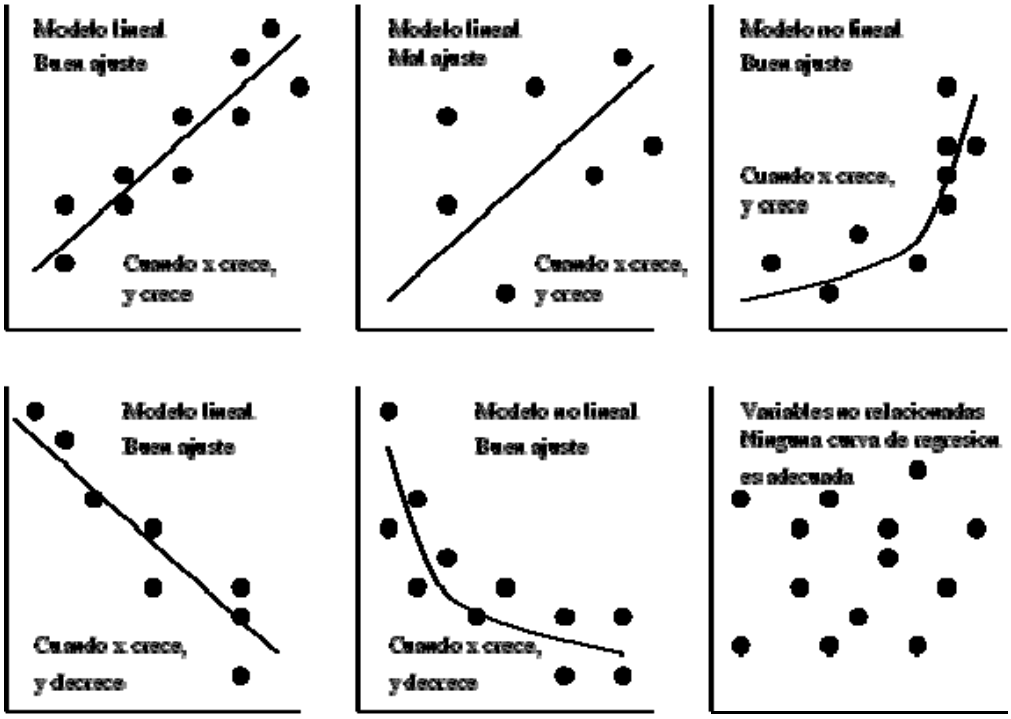
Sean {(xi,yj); nij i=1,...,r; j=1,...,c} los valores y la distribución de frecuencias conjunta de las variables analizadas X e Y. Se denomina diagrama de dispersión o nube de puntos a la representación, en un sistema de ejes cartesianos (X, Y), de los valores observados de las variables, en el que a cada par (xi,yj) se le asocia su frecuencia conjunta de observación nij.

Esta representación es conveniente como paso previo al análisis de regresión en tanto que la forma de la nube de puntos permite obtener una idea inicial del tipo de dependencia existente entre X e Y.

Un diagrama de dispersión ofrece una idea bastante aproximada sobre el tipo de relación existente entre dos variables. Pero, además, un diagrama de dispersión también puede utilizarse como una forma de cuantificar el grado de relación lineal existente entre dos variables: basta con observar el grado en el que la nube de puntos se ajusta a una línea recta.

Mediante las técnicas de regresión de una variable Y sobre una variable X, se busca una función que sea una buena aproximación de una nube de puntos (x_i,y_i), mediante una curva del tipo $\hat{Y} = f(X)$ [BMA04]. Para ello la diferencia entre los valores y_i e $\hat{y}_i$ ha de ser tan pequeña como sea posible.





Diferentes nubes de puntos y modelos de regresión para ellas.

En el cálculo de los coeficientes del modelo de regresión lineal simple intervienen, aparte de la media, la varianza y la covarianza .

.La varianza de Y se expresa mediante : $s_Y^2 = \frac{\sum_{i=1}^{n} (y_i - \overline{y})^2}{n}$.

La covarianza de n valores $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ de (X,Y) indica si la posible relación entre dos variables es directa o inversa.

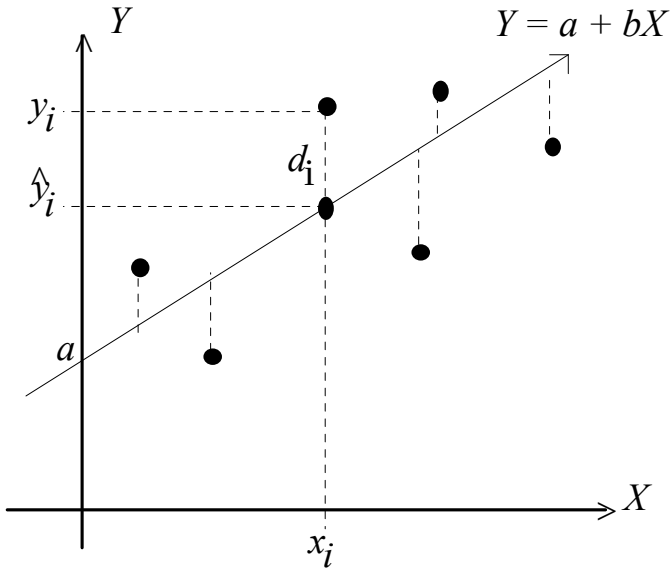
$$S_{xy} = \frac{1}{n}\sum_i (x_i - \bar{x})(y_i - \bar{y})$$

- Directa: $S_{XY} > 0$. La pendiente de la recta también es positiva.
- Inversa: $S_{XY} < 0$. La pendiente de la recta también es negativa.
- Incorreladas: $S_{XY} = 0$. En este caso las rectas de regresión serán paralelas a los ejes de ordenadas.

El signo de la covarianza informa si el aspecto de la nube de puntos es creciente o no, pero no dice nada sobre el grado de relación entre las variables.

5.2.2. Mínimos cuadrados (Regresión lineal Simple)

Se deben determinar los coeficientes a y b de la ecuación de la recta: Y=a+bx, que mejor se ajuste a los n pares (xi,yi) observados.



Esto equivale a que los valores de a y b hagan mínima la ecuación expresada a continuación, para encontrar una función que minimice la distancia entre lo encontrado (y) y lo pronosticado(y'). Las diferencias entre los valores observados y_i y los valores que predice el modelo $f(x_i)$, se denominan residuos.

$$\sum_{i=1}^{n} d_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - (a + bx_i))^2 = \sum_{i=1}^{n} (y_i - a - bx_i)^2$$

Los coeficientes a y b se obtienen de las ecuaciones normales:

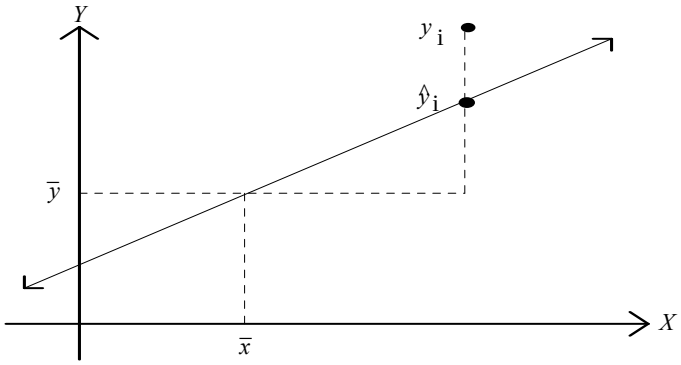
$$\sum_{i=1}^{n} y_i = na + b\sum_{i=1}^{n} x_i \quad \text{y} \quad \sum_{i=1}^{n} x_i y_i = a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2$$

$$b = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2} \qquad b = \frac{s_{XY}}{s_X^2}$$

$a = \bar{y} - b\bar{x}$ $\quad Y = a + bX$ por tanto $Y - \bar{y} = b(X - \bar{x})$

Interpretación: El parámetro a representa la ordenada en el origen, esto es el valor que toma la variable dependiente cuando la variable independiente toma el valor 0 y el parámetro b es la pendiente de la recta de regresión.

Es importante observar la diferencia de los roles que desempeñan x e y. Geométricamente, la recta de regresión lineal de y con respecto a x minimiza la suma de las distancias verticales de los puntos (x_i, y_i) a la recta. La recta de regresión lineal de x con respecto a y minimiza las distancias horizontales. Las dos rectas se cortan en el centro de gravedad, $(\bar{x}, \bar{y})$, de la nube de puntos. La separación entre las dos rectas es mayor cuando la correlación es más débil.



$$y_i - \bar{y} = (y_i - \hat{y}_i) + (\hat{y}_i - \bar{y})$$

5.2.3. Mínimos cuadrados (Regresión lineal múltiple)

En el caso general, el modelo de regresión lineal múltiple con p variables responde a la ecuación:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_p X_{ip} + E_i \qquad \text{con i = 1..n}$$

de modo que los coeficientes β_i se estiman siguiendo el criterio de mínimos cuadrados [RLM00] :

$$\min \sum_{i=1}^{n} (\mathrm{Y}_i - \beta_0 - \beta_1 \mathrm{X}_{i1} - \beta_2 \mathrm{X}_{i2} - \ldots - \beta_p \mathrm{X}_{ip})^2$$

utilizando notación matricial:

$$\mathrm{Y} = \mathrm{X}\beta + \varepsilon$$

donde:

$$\mathrm{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{bmatrix} \quad \mathrm{X} = \begin{bmatrix} 1 & \mathrm{X}_{11} & \mathrm{X}_{12} & \ldots & \mathrm{X}_{1p} \\ 1 & \mathrm{X}_{21} & \mathrm{X}_{22} & \ldots & \mathrm{X}_{2p} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & \mathrm{X}_{(n-1)1} & \mathrm{X}_{(n-1)2} & \ldots & \mathrm{X}_{(n-1)p} \\ 1 & \mathrm{X}_{n1} & \mathrm{X}_{n2} & \ldots & \mathrm{X}_{np} \end{bmatrix}$$

$$\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \ldots \\ \varepsilon_n \end{bmatrix}$$

De donde los estimadores mínimo cuadráticos se obtienen a partir de la ecuación:

$$L = (\mathrm{Y} - \mathrm{X}\beta)^T (\mathrm{Y} - \mathrm{X}\beta)$$

$\hat{\beta}$ es la solución de $\frac{\partial L}{\partial \beta} = 0$

$$\hat{\beta} = (\mathrm{X}^T \mathrm{X})^{-1} \mathrm{X}^T Y$$

5.2.4. Regresión no lineal

Para ciertas familias de funciones, se transforma el problema para conseguir una regresión lineal. En la siguiente tabla se presentan algunos casos frecuentes.

Familia	Funciones	Transformación	Forma afín
exponencial	$y = ae^{bx}$	$y' = \log(y)$	$y' = \log(a) + bx$
potencia	$y = ax^b$	$y' = \log(y)\ x' = \log(x)$	$y' = \log(a) + bx'$
inversa	$y = a + b/x$	$x' = 1/x$	$y = a + bx'$
logística	$y = 1/(1 + e^{-(ax+b)})$	$y' = \log(y/(1-y))$	$y' = ax + b$

Para otras familias, como la función de regresión

$$m(x,\vec{\alpha}) = \alpha_1 \exp(\sqrt{\alpha_2 x} + \alpha_3 x^2)$$

no existe la posibilidad de transformar en lineal.

La forma general de estos modelos es $y_i = m(x_{i,}\vec{\alpha}) + \varepsilon_i$ siendo m una función que depende de un vector de parámetros $\vec{\alpha}$ que es necesario estimar, ε_i son los errores que se supone que verifican las mismas hipótesis que el modelo lineal. [MES03]

La estimación del vector de parámetros $\vec{\alpha}$ se realiza por el método de mínimos cuadrados. Esto es, se calcula el $\vec{\alpha}$ que minimiza la función de la suma de residuos al cuadrado,

$$\psi(\vec{\alpha}) = \sum_{i=1}^{n} (y_i - m(x_i, \vec{\alpha}))^2$$

El algoritmo para minimizar esta función es un procedimiento iterativo que se basa en el método de Gauss-Newton o en algoritmos más complejos como el algoritmo de Levenberg-Marquard. Para aplicar estos procedimientos se parte de unos valores iniciales $\vec{\alpha}_0$ que permiten iniciar el algoritmo iterativo y en cada etapa i se obtiene un nuevo estimador $\vec{\alpha}_i$ hasta obtener la convergencia según un criterio de parada predefinido.

Levenberg-Marquardt trabaja de la siguiente forma [EST04]:

Asume que la función que va a ser modelada es lineal. Basándose en esta suposición, el mínimo puede ser determinado exactamente en un solo paso. El mínimo calculado es testeado, y si el error es menor, el algoritmo cambia los pesos al nuevo punto. Este proceso es repetido iterativamente. Como la suposición inicial es errónea, puede conducir fácilmente al algoritmo a testear un punto que proporcione un error inferior al punto que está en curso. El aspecto más importante del algoritmo es que la determinación del nuevo punto es un compromiso entre el avance en la dirección de la máxima pendiente y el salto anterior. Los pasos satisfactorios son aceptados y conducen a un fortalecimiento de la presunción de linealidad (que es aproximadamente cierta cerca de un mínimo). Los pasos fallidos son rechazados y conducen a una "cuesta abajo", considerando la linealidad como una suposición pobre. De esta forma, el algoritmo cambia su enfoque continuamente y tiene un rápido progreso.

El algoritmo es diseñado específicamente para minimizar funciones que estén en forma de suma de cuadrados. Por tanto existe un compromiso entre el modelo lineal y el método del gradiente. Un paso en el algoritmo es aceptado sólo si mejora el error, y en caso de ser necesario el enfoque de pendiente-gradiente es usado con un incremento suficientemente pequeño para garantizar el movimiento "cuesta abajo".

Levenberg-Marquardt usa la siguiente fórmula:

$$\boxed{\Delta\omega = -(J^T J + \lambda I)^{-1} J^T \varepsilon}$$

donde ε es el vector de errores, y J es la matriz de derivadas parciales(Jacobiano) de estos errores con respecto a los pesos.

El parámetro de control λ regula la influencia relativa de estas dos aproximaciones, influyendo tanto en la dirección como en el tamaño del paso dado.

Cada vez que el algoritmo tiene éxito en la disminución del error, decrementa el parámetro de control por un cierto factor, fortaleciendo la hipótesis de linealidad e intentando saltar directamente al mínimo. Cada

vez que falla en reducir el error, incrementa el parámetro de control por un factor, dando más influencia al método del gradiente, siendo esto positivo si la en la iteración actual se está lejos de la solución [MNT04].

5.2.5. Bondad de ajuste

Correlación

El coeficiente de correlación lineal de Pearson de n observaciones $(x_1,y_1),(x_2,y_2),...(x_n,y_n)$ de (X,Y) mide el grado de asociación lineal entre las variables X e Y. [EE04]

El coeficiente de correlación lineal de Pearson de dos variables indica si los puntos tienen una tendencia a disponerse alineadamente (excluyendo rectas horizontales y verticales).

Está dado por:

$$r = \frac{S_{xy}}{S_x S_y}$$

Propiedades de r:

- Tiene el mismo signo que S_{XY} por tanto de su signo se obtiene si la posible relación es directa o inversa.
- r es útil para determinar si hay relación lineal entre dos variables, pero no servirá para otro tipo de relaciones (cuadrática, logarítmica,...)
- Una correlación alta no indica que una variable dependa de la otra o que sea causa de las variaciones en la otra. La asociación entre ellas no necesariamente es “causal”.
- Una correlación alta indicaría que el modelo lineal es adecuado para hacer predicciones en el intervalo de variación de los datos; fuera de él, el tipo de relación entre las variables puede cambiar o no existir.
- Es adimensional, Sólo tomando valores entre [-1,1]. Las variables son incorreladas si r=0

Coeficiente de determinación

Definimos como medida de bondad de un ajuste de regresión, o coeficiente de determinación a:

$$\sum_{i=1}^{n}(y_i-\bar{y})^2=\sum_{i=1}^{n}(y_i-\hat{y}_i)^2+\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2$$

$SCT = SCE + SCR$. El coef.de determinación es: $r^2=\frac{SCR}{SCT}$

$$1=\frac{SCE}{SCT}+r^2$$

De $1=\frac{SCE}{SCT}+r^2$ se tiene $0\le r^2\le 1$.

Entonces, $-1\le r\le 1$.

- r^2 =1, sólo si, SCE=0, o sólo si, $y_i=\hat{y}_i$

  Entonces, todos los y_i están en la recta de regresión.

  Existe correlación perfecta entre X e Y.

- r^2 = 0, sólo si, SCR=0, o sólo si, $\hat{y}_i=\bar{y}$ .Entonces, no hay correlación ni regresión.

  El coeficiente de determinación r^2, es una medida de la proximidad del ajuste de la recta de regresión. Cuanto mayor sea el valor de r^2, mejor será el ajuste y más útil la recta de regresión como instrumento de predicción.

Testeo de resultados

Tanto el R^2 como el coeficiente de correlación no son las medidas más adecuadas para evaluar la **predicción** de un modelo; en el mejor de los casos se trata de medidas del ajuste de la ecuación a los datos, no de la capacidad predictiva del modelo. En algunos casos la idea que nos transmite el R^2 puede coincidir con la de las variables que a continuación se muestran, pero en otros no.

Desde este punto de vista, la precisión de la herramienta puede ser evaluada usando los criterios propuestos por Samuel Conte, et al. [CDS86] basados en los siguientes estadísticos.

- **Magnitude of relative error (MRE)**

$$MRE=\frac{|Estimated-Observed|}{Observated}$$

A menores valores de MRE mejor es la predicción.

- **Mean Magnitude of relative error (MMRE)**

$$MMRE = \frac{\sum_{i=1}^{n} (MRE_i)}{n}$$

- **Root Mean Square (RMS)**

$$RMS = \sqrt{\frac{\sum_{i=1}^{n} (Estimated_i - Observed_i)^2}{n}}$$

El grado de mal emparejamiento entre la predicción y el valor real se calcula restando los dos valores y elevando el resultado al cuadrado. Este "error cuadrado" se promedia sobre todas las predicciones para estimar la distancia entre los valores reales y las predicciones. La elevación al cuadrado tiene dos ventajas: por un lado, da un mayor peso a los errores graves; por otro lado, asegura que todos los errores son positivos y se suman a la hora de calcular la media. Esta medida da información sobre la potencia del error.

- **Relative Root Mean Square(RRMS)**

$$RRMS = \frac{RMS}{\sum_{i=1}^{n} (Observed_i)/n}$$

- **Prediction Level**

$$\Pr ed(k) = \frac{m}{N}$$

$\Pr ed(k)$ es el *nivel de predicción al k%*, siendo m un subconjunto de N elementos cuyo MRE # k.

Dicho de otra forma, $\Pr ed(k)$ se define como el cociente del número de casos en los que las estimaciones están dentro del límite absoluto *k* de los valores reales entre el número total de casos. Por ejemplo PRED(0.1) = 0,9 quiere decir que 90% de los casos tienen estimaciones dentro del 10% de sus valores reales; PRED(0,25) = 0,9 quiere decir que

el 90% de los casos tiene estimaciones dentro del 25% de sus valores reales. Habitualmente se aceptan modelos que cumplan PRED(0,25) ≥ 0,75.

5.3. Regresión Borrosa

5.3.1. Aspectos teóricos

Características básicas del método de regresión difusa

El análisis de la regresión es una de las herramientas estadísticas más usadas por los científicos e ingenieros. Los métodos de análisis de regresión usan modelos, que basados en un conjunto de datos, obtienen una ecuación de predicción. Los análisis de regresión pueden ser realizados por programas informáticos, pero la mayoría de estos lo hacen sobre valores nítidos.

Los investigadores están considerando el problema de que los datos contengan incertidumbre, imprecisión o vaguedad. También en la utilización de datos con términos lingüísticos o números simbólicos que representen términos cualitativos. La teoría de conjuntos borrosos puede ser utilizada para el análisis de la regresión difusa de diferentes maneras.

En el análisis de regresión clásico o nítido los errores de desajuste, entre el modelo de regresión obtenido y los valores observados, son una variable que tiene distribución normal, varianza constante y media igual a cero. Sin embargo, en el análisis de regresión difusa estos errores son vistos como borrosidad de la estructura del modelo. El primero en tomar esta consideración fue Tanaka en [TUA82]. Desde entonces otros investigadores de esta área han iniciado sus trabajos a partir del estudio citado con construcciones como Celmins, Diamond, Ishibushi, Savic, Pedricz entre otros.

En el estudio de Tanaka propone un modelo lineal borroso cuyos parámetros sean difusos que son obtenidos de entradas nítidas y salidas nítidas o difusas. Esto hace que el problema se pueda reducir a u problema de programación lineal, fácil de implementar y usar.

En este apartado se ha considerado otro método de regresión difusa el cual se basa en entradas y salidas difusas que determinan parámetros nítidos. El método es una modificación del análisis de regresión difusa publicado por Kalenina et al . en [IKW01,IKW03,KW00].

Estos autores formulan el problema de la regresión suponiendo:

- Que existe una dependencia entra las variables independientes y la dependiente y esta se puede formular en forma nítida.
- Que las entradas, obtenidas por observación, se pueden extender como un número borroso teniendo en cuenta el conocimiento del área de trabajo. Esta extensión se puede realizar con número de tipo L simétrico o elíptico.

Eliminado: <#>Soft Computing¶
Las técnicas de Soft Computing engloban básicamente, la lógica borrosa, las redes neuronales, la computación evolutiva, los algoritmos genéticos y el razonamiento probabilístico.¶
¶
El Soft Computing desempeña un papel muy importante en las ciencias y en la ingeniería, aunque su aplicación se extenderá en otros muchos campos, debido a los resultados satisfactorios que se han ido obteniendo con su uso, ¶
¶
Las técnicas de Soft Computing destacan por su tolerancia a la imprecisión, la incertidumbre, grado de credibilidad y la aproximación. ¶
¶
Estas técnicas usan la mente humana como modelo. Las líneas de investigación que lleva a cabo son:¶
¶
<#>Una nueva generación de motores de búsqueda en Internet, que usan técnicas de Soft Computing y tratan de mejorar la búsqueda lexicográfica actual, usando una búsqueda conceptual.¶
¶
<#>Técnicas avanzadas para descubrir “perfiles de usuario” que permitan un uso de internet más inteligente “a la carta”.¶
¶
<#>Comercio electrónico basado en técnicas de Soft Computing, por ejemplo, lo que el profesor Mandani denomina Soft Knowledge.¶
¶
<#>Semantic Web. ¶
¶
En los últimos años se ha podido comprobar un rápido crecimiento de las aplicaciones de la lógica borrosa y las redes neuronales, en diversos campos: electrónica de consumo, control de procesos industriales, reconocimiento del habla, visión artificial, tratamiento de la señal, reconocer y clasificar imágenes, manejar vehículos en trafico denso y un largo etcétera.¶
¶
¶
<#>Lógica borrosa y sus aplicaciones¶
¶
La lógica borrosa es básicamente una lógica multievaluada que permite valores intermedios para poder definir evaluaciones convencionales como sí ... [1]

- Por último, exponen que el modelo gestiona de manera fácil y comprensible este tipo de información.

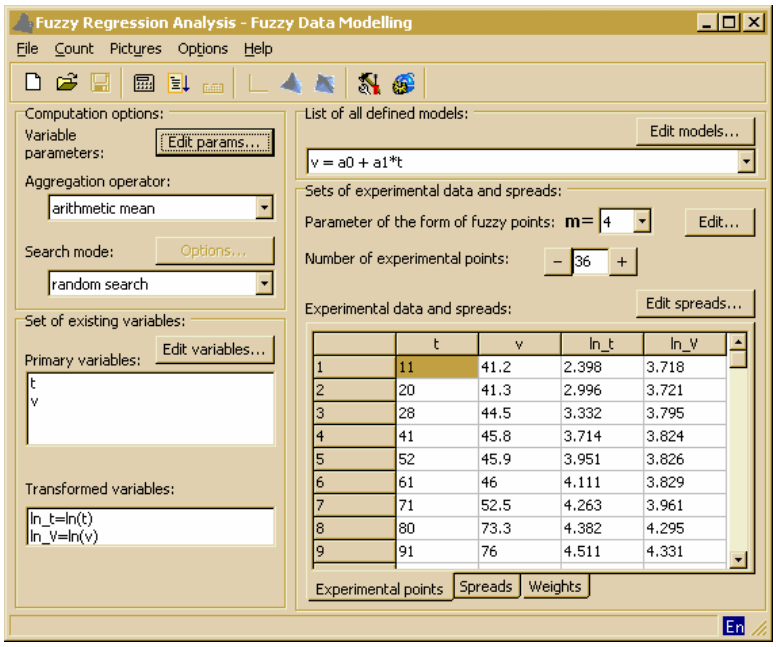
Función de pertenencia de un número difuso de Tipo L simétrico

La versión usada de la herramienta sólo tiene implementada la salida y las entradas de datos modeladas mediante números difusos de tipo L-simétricos (véase la figura y la fórmula)

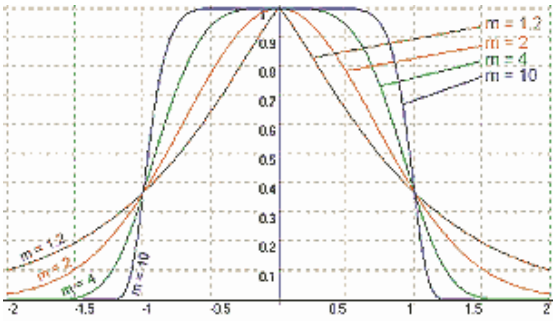
$$\mu(t) = \exp\left\{-\left(\frac{t-b}{s}\right)^m\right\}, s>0, m>1$$

La manipulación de los parámetros , m(forma) y s (amplitud), permite simular diferentes tipos de valores difusos, desde un intervalo a uno nítido.

El valor del parámetro m debe ser mayor que 1 (m<=1). El comportamiento de este parámetro para distintos valores es: si m aprox. igual a 1 se obtiene números borrosos con forma triangular, para m aprox. Igual a 2 se obtienen formas de distribuciones uniformes y para valores de m aprox. Igual a 10 son formas de distribuciones normales (véase la figura)

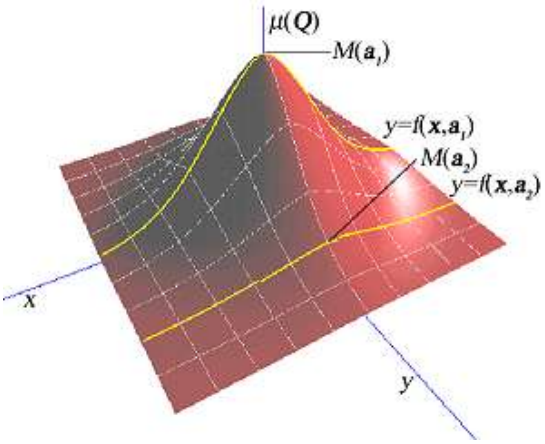


Ventana principal de furea



Gráfica de la función de pertenencia dependiendo del valor del parámetro de forma en un número borroso de tipo L-simétrico.

Por otra parte para el parámetro s: si s = 0 , es decir, sin amplitud, se obtiene la representación formal de un número nítido. Para valores superiores se está considerando el grado de la incertidumbre inherente a la observación.



Medida de similaridad de un punto borroso con una función nítida.

Funciones de pertenencia de un punto difuso

Las funciones de pertenencia de un punto difuso se obtienen al aplicar el producto de la T-Norma a las funciones de pertenencia de sus coordenadas.

$$\mu_Q(x_1,\ldots,x_n,y) = \exp\left(-\frac{|y-\overline{y}|^m}{\alpha}\right)\cdot\prod_{i=1}^{n}\exp\left(-\frac{|x_i-\overline{x_i}|^m}{\beta_i}\right)$$

Medidas de similaridad de un punto difuso con una función nítida

Uno de los conceptos clave de la regresión borrosa es la medida de similaridad de un punto borroso con respecto a una curva paramétrica. Este es igual al máximo de la función de pertenencia del punto de función sobre el conjunto de puntos que pertenecen a la curva.

La representación gráfica de las medidas de similaridad puede observarse en la figura de un punto cualquiera con una función nítida que se define con la siguiente fórmula

$$M_f(a) = \sup_{x\in\mathbb{R}^n}\ \mu_{Q_x}(x, f(x,a))$$

Este criterio de ajuste de cada punto individual puede interpretarse como un grado de compatibilidad entre el punto y el modelo. Para este tipo de números difusos, el uso de la T-Norma y de una función lineal permite obtener una expresión analítica para ese determinado valor

$$M_f(a) = \exp(-d_f(a))$$

donde

$$d_f(a) = \frac{\left|\left(a_0+\sum_{j=1}^{n}a_j\cdot x_j\right)-y_i\right|^m}{\left(\alpha_i^{\frac{m}{m-1}}+\sum_{j=1}^{n}|x_j\cdot\beta_j|^{\frac{m}{m-1}}\right)^{m-1}}$$

Para el caso de funciones no lineales no puede obtenerse una expresión analítica y debe resolverse mediante métodos numéricos que la versión actual de la herramienta no provee.

Operadores de agregación

El segundo concepto clave de la regresión borrosa es un operador de agregación.

Se introduce una familia de operadores de agregación basados en la media ponderada que recogen la información de cómo la curva "pasa a través" de todos los datos borrosos del experimento.

En la herramienta se utilizan tres por su aplicabilidad práctica:

- **Media Geométrica(MG(a))** es el llamado operador fuerte. Éste tiende a dar el grado de conformidad de todos los puntos al mismo tiempo. Su funcionamiento es similar al Método de mínimos cuadráticos (ver fórmula).

- **Media Aritmética(MA(a))** es el llamado operador compensador. Éste tiende a dar la media del grado de similaridad de la mayoría de los puntos del modelo (ver fórmula).

- **Media Cuadrática (MQ(a))** es el llamado operador elitista. Éste tiende a dar la media ajustada del grupo "mejor" (o élite) de puntos (ver fórmula).

Para su aplicación se asume tener N puntos difusos. Esta asunción se debe hacer con el fin de obtener la medida de similaridad agregada para el modelo ajustado a los N puntos difusos, esta finalidad es para la que se introducen los operadores de agregación de las fórmulas antes expuestas.

$$MG(a) = \prod_i M_i^{w_i}(a)$$

$$MA(a) = \sum_i w_i M_i(a)$$

$$MQ(a) = \left(\sum_i^N w_i M_i^2(a) \right)^{\frac{1}{2}} \quad s.t. \sum_i^N w_i = 1$$

La maximización del operador permite obtener la solución óptima de cada problema.

$$a_G^* = \arg \min_{a \in R^n} \sum_i w_i d_i(a)$$

$$a_L^* = \arg \max_{a \in R^n} \sum_i w_i e^{-d_i(a)}$$

Dependiendo del tipo de operador elegido, podemos alcanzar diversos grados de compensación para los puntos de cada experimento.

Consecuentemente, obtenemos el criterio no prefijado, ni sesgado que resulta imparcial con los “puntos atípicos”(outliers) y su alto grado de no linearidad.

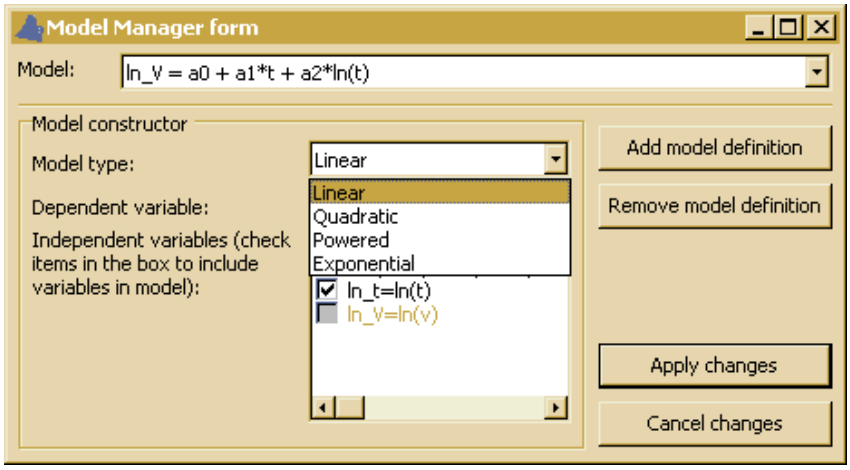
Todo lo expuesto, efectivamente reduce el problema del análisis de regresión a un problema de optimización no lineal sin restricciones.

5.3.2. La herramienta Furea

El método anteriormente mencionado se ha implementado en la herramienta llamada Furea (FUzzy REgresion Analysis) utilizando el lenguaje de programación C++ para Windows.

El proceso de funcionamiento se describe brevemente de la forma siguiente:

El usuario define las variables de entrada a usar, los modelos a construir y la forma de desarrollar los cálculos.



Selección del modelo de regresión de Furea

Definición y borrosificación de variables

En Furea existen 2 tipos de variables de entrada:

Las *primarias* que son aquellas que se obtienen(directamente) como muestra del experimento.

Las variables *transformadas* que son las que el usuario tiene la opción de definir borrosificando las anteriores primarias.

En esta herramienta la función de un vector de números borrosos es reemplazada por la aproximación de un número borroso del tipo L-Simétrico. Para cada punto del experimento la media y la amplitud del intervalo para ese número son obtenidas aplicando la extensión principal

de primer orden de la fórmula da las series de Taylor, evaluada con el valor medio del vector borroso original (véase la fórmula)

$$f(X) = f(a) + \sum_i (X_i - a_i) \frac{\partial f(x)}{\partial x_i}\bigg|_{x=a} = \left(f(a), \sum_i a_i \left| \frac{\partial f(x)}{\partial x_i} \right|_{x=a} \right)_L$$

Siendo $f(x)$ la función de transformación nítida

$X_i = (a_i, \alpha_i)_L$ el valor i-esimo de la variable borrosa

X el vector borroso

Construcción del modelo

Aún habiendo sido diseñado para el análisis de regresión de funciones borrosas lineales, Furea puede extender la capacidad de análisis más allá de los modelos lineales, pero solo a modelos de parámetros lineales.

El usuario puede seleccionar un tipo de modelo predefinido (véase la figura), pero también tiene la opción de construir su propio modelo.

Los tipos de modelos predefinidos son:

Lineal:

$$y = a_0 + \sum_i a_i x_i$$

Cuadrático:

$$y = a_0 + \sum_i \sum_{j \geq i} a_{ij} x_i x_j$$

En potencias:

$$y = a_0 \cdot \prod_i x_i^{a_i}$$

Exponencial:

$$y = a_0 \cdot \exp\left(\sum_i a_i x_i \right)$$

Todas las transformaciones de las variables son hechas de forma automática y transparente al usuario.

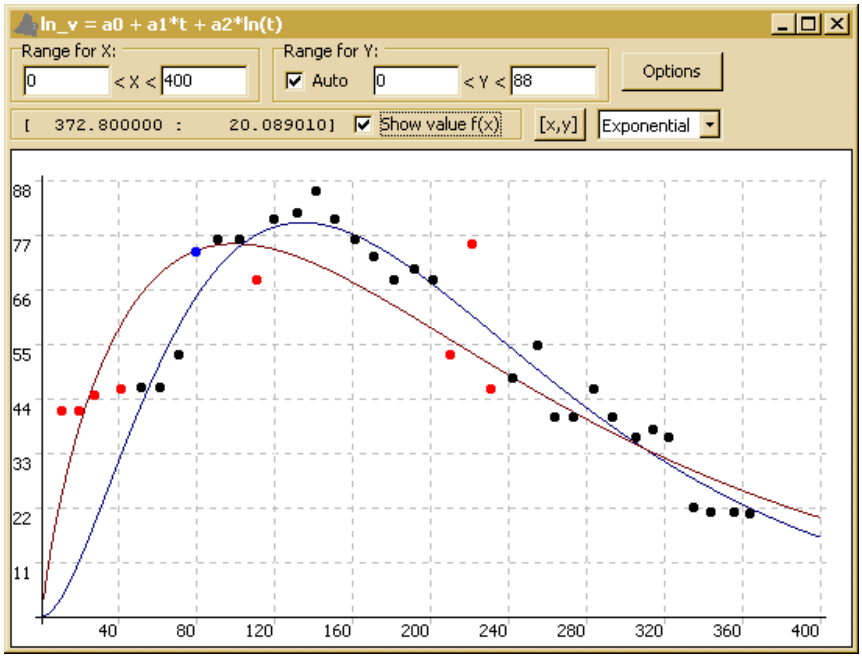


Gráfico de la detección de valores atípicos (“outliers”)

Selección del desarrollo de los cálculos

Todas las decisiones sobre la opción de operadores de agregación a usar, del método de búsqueda de la solución óptima, de la elección de los parámetros de forma y amplitud y del modelo a utilizar en cada cálculo se establecen por selección en la pantalla principal de Furea de manera sencilla.

Detección de datos atípicos o “Outliers”

El programa tiene capacidad para la detección de datos atípicos(“outliers”). Devuelve los resultados con indicación de los puntos (ver figura) con medida de similaridad menor que el umbral definido por el usuario.(El término “medida de similaridad de un punto a una función” y “el grado de pertenencia de una función a un punto borroso” son equivalentes.)

Es decisión del usuario si deja todo como está, o excluye los puntos que pudieran producir imprecisión en los cálculos o si selecciona otros, que le permitan encontrar el modelo no lineal.

Asimismo en el caso de una o dos variables independientes, el usuario puede observar el gráfico con los dos modelos integrados, el llamado LSM clásico y el modelo de la f.regresión, viendo los puntos “outliers” que son dibujados en un color diferente.

Ver la parte inferior derecha de la figura. Este gráfico permite un mejor ajuste de los datos experimentales.

5.4. Herramientas

El objetivo de esta sección es exponer el contexto actual del análisis de regresión mediante el estudio de las principales herramientas que actualmente se usan en el mercado. Se describen a continuación algunas de ellas.

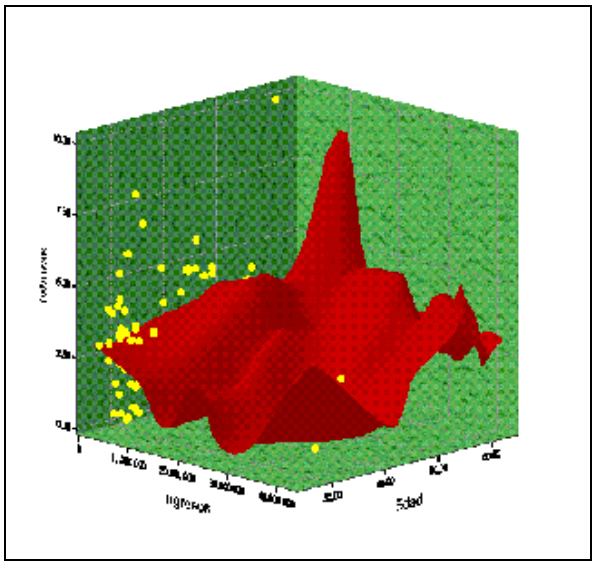
5.4.1. SPSS

SPSS (Statistical Product and Service Solutions) [SPS05] es un paquete de software estadístico y tratamiento de datos con más de 35 años de experiencia en el sector. Engloba una línea de productos que es modular, integrada y con todas las funcionalidades necesarias para llevar a cabo cada paso del proceso analítico - planificación, recogida de datos, acceso y preparación de los datos, análisis, creación de informes y distribución de los mismos.

La interfaz gráfica de usuario lo hace sencillo de utilizar dado que le proporciona toda la gestión de los datos, los estadísticos y los métodos de creación de informes que necesita para realizar todo tipo de análisis.

SPSS proporciona los procedimientos estadísticos más usuales para análisis básico, incluyendo: sumas, frecuencias, tablas de contingencia, estadísticos descriptivos, análisis factorial y regresión.

En el módulo de base se dispone de una serie de modelos de regresión: lineal, logarítmico, inverso, cuadrático, cúbico, compuesto, potencial, S, creciente, exponencial y logístico. Respecto al ajuste de curvas, hay disponibles para especificar 11 tipos de curvas.



El módulo “SPSS Modelos de Regresión” permite ir más allá del análisis de datos básico. Entre las características que añade están:

- Regresión logística multinomial
    - Obtención de los predictores eligiendo uno de los cuatro métodos de regresión posibles: entrada hacia delante, eliminación hacia atrás, por pasos hacia delante, o por pasos hacia atrás
    - Opción de selección de la regla de introducción y eliminación desde el análisis
    - Utilización de los métodos de Puntuación y de Wald que le ayudarán a obtener resultados más rápido si tiene un gran número de predictores
    - Utilización del Criterio e Información de Akaike (AIC) y del Criterio de Información de Bayesiano (BIC), también llamado Criterio Bayesiano de Schwarz (SBC) para acceder al modelo ajustado
- Regresión logística binomial:

  Permite pronosticar variables dicotómicas. Este procedimiento ofrece numerosos métodos por pasos para seleccionar covariables continuas o las categorías que mejor pronostican la variable respuesta.
- Regresión no lineal restringida y no restringida:

  Permite un mayor control sobre el modelo.
- Mínimos cuadrados ponderados:

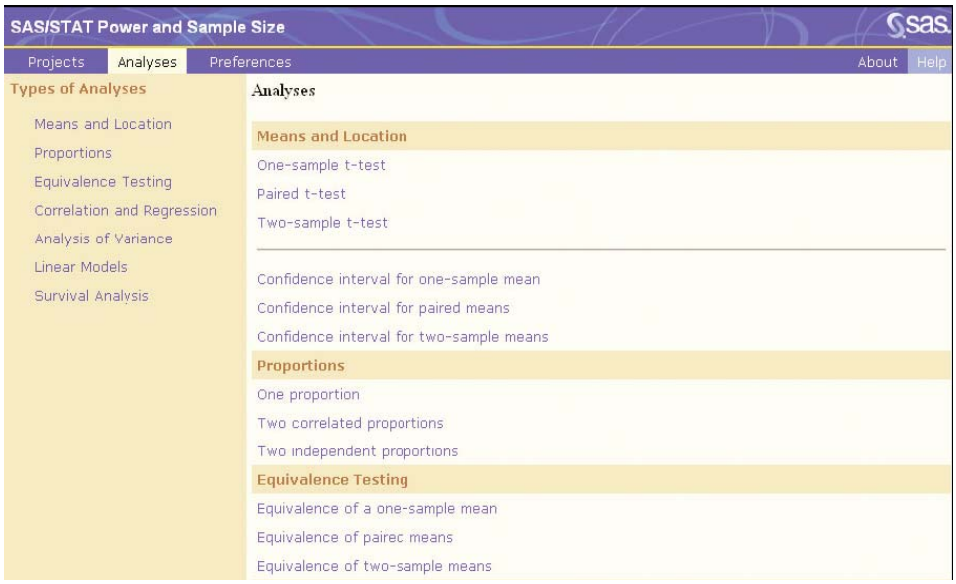
  Sirve para controlar las correlaciones entre las variables predictoras y los términos de error que aparecen a menudo en los datos que dependen del tiempo.
- PROBIT:

  Se utiliza para analizar la potencia de las respuestas a estímulos como, dosis de un medicamento, precios o incentivos. Evalúa el valor del estímulo mediante una transformación logit o probit de la proporción que responde.

5.4.2. SAS

Otra herramienta destacada es **SAS/STAT** [SAS04]. Una de sus características principales es una gran versatilidad, la cual hace posible evaluar datos provenientes de una gran variedad de disciplinas. La tecnología empleada permite aplicar un extenso conjunto de técnicas especializadas para cada tipo de industria.

Proporciona un gran conjunto de herramientas para el análisis estadístico, incluyendo análisis de varianza, regresión, análisis de dato por categorías, análisis multivariante, análisis no paramétrico...



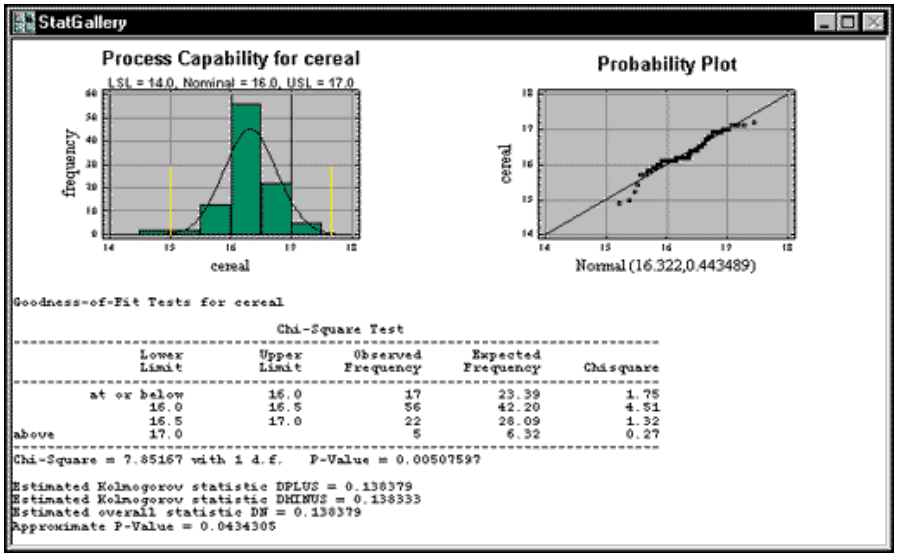
Con respecto a la regresión, el procedimiento general que emplea SAS/STAT se basa en mínimos cuadrados para estimar los parámetros, incluyendo 9 técnicas de selección de modelo diferentes, pudiendo obtener diferentes diagnósticos y medidas. El uso de modelos más especializados se realiza ajustándolos a modelos lineales como los mixtos, no lineales, curvas cuadráticas (quadratic response surface models)...

5.4.3. STATGRAPHICS Plus

Statgraphics [STP05] incluye las utilidades necesarias para el análisis estadístico de datos (análisis estándar para descripción, comparación de datos, análisis multivariante, análisis de series temporales, regresión avanzada, análisis para el control de calidad, y diseño de experimentos), gráficos interactivos, y gráficos e informes para presentaciones.

Statgraphics tiene una estructura modular constituida por 3 módulos diferentes, en los que se puede encontrar más de 150 procedimientos de distribución.

El módulo básico (Standard Edition) aporta todas las herramientas estadísticas básicas, entre ellas la regresión simple, múltiple y polinomial. A partir de éste se pueden seleccionar las funciones estadísticas adicionales necesarias en los otros módulos.



En la versión Professional, el módulo de regresión avanzada permite efectuar una exploración completa de los datos, formular modelos de regresión múltiple complejos, validar los métodos de un laboratorio o simplemente buscar el mejor modelo de regresión. Las características más significativas son:

- **Análisis Rápido de Regresión**

  El programa una vez introducidas las variables de interés, ajusta instantáneamente todas las posibles regresiones, ordenadas según ajusten R2 (Coeficiente de Determinación) o Malows'Cp, y muestra un resumen de algunos estadísticos en una tabla de regresión.

  Una vez obtenido el mejor modelo, se puede obtener la imagen completa de la regresión múltiple. Este módulo está totalmente integrado con el Modulo Básico y con el resto de módulos de Statgraphics, permitiendo el acceso a todos los procedimientos estadísticos.

- **Detección rápida de diferencias entre grupos**

  Con este módulo es posible también detectar diferencias significativas en las relaciones entre dos o más grupos y comparar modelos de regresión simple por medio de una variable categórica.

Statgraphics ajusta de modo automático una línea distinta para cada nivel de la variable categórica y a continuación verifica si las pendientes y los puntos de corte difieren. Así, el usuario evita la necesidad de efectuar la misma regresión sobre diferentes grupos de datos.

Por otra parte, el programa aporta la facilidad de dibujar tantas líneas de regresión como se desee sobre un único gráfico. De este modo, resulta sencillo observar las diferencias.

- **Ajuste no lineal**

El programa permite ajustar cualquier tipo de modelo no lineal con hasta 12 parámetros. El usuario debe introducir tan sólo la función.

Mediante gráficos 3D se pueden explorar los datos de modo visual sobre la pantalla y detectar rápidamente deficiencias en el modelo. Los gráficos se pueden editar con facilidad y personalizar para adaptarlos a las necesidades del usuario.

- **Modelización de datos binarios mediante Regresión Logística**

Statgraphics contiene la posibilidad de construir un modelo de regresión experto. La regresión logística es útil cuando se necesita un modelo que ayude a determinar éxitos o fracasos a partir de varias variables. Esta característica permite examinar los efectos de variables categóricas o continuas sobre datos binarios o de probabilidad. Finalmente, mediante el acceso instantáneo a los gráficos en todos los procedimientos de Statgraphics, se puede fácilmente representar los datos y visualizar la bondad del modelo.

- **Tratamiento de datos confusos**

Los Modelos Lineales Generales (GLM) se utilizan para obtener datos recogidos por medidas repetidas, diseños anidados, cruzados, etc. Estos modelos permiten desarrollar y analizar problemas personalizados de regresión y diseño.

- **Calibración**

Este módulo es especialmente interesante para la industria farmacéutica, química, alimentaria y medioambiental en cuanto al tema de calibración. La calibración permite comparar resultados de laboratorio con un estándar conocido definiendo un modelo que permita, dado Y, predecir X con unos límites de confianza.

5.4.4. 

R [RPS05], [VSR04] es un sistema para análisis estadísticos y gráficos. Tiene una naturaleza doble de programa y lenguaje de programación y es considerado como un dialecto del lenguaje S creado por los Laboratorios AT&T Bell. R se distribuye gratuitamente bajo los

términos de la GNU General Public Licence (con website en http://www.gnu.org).

Consta de un "sistema base" y de paquetes adicionales que extienden la funcionalidad.

Permite realizar regresión múltiple y polinomial, y algunos modelos no lineales, aunque en la mayoría de los casos han de ser aproximados. Las funciones para ello son glm(), optim() y nlm().

Otras facilidades disponibles en R para regresión y análisis de datos:

Modelos mixtos:

Mediante el paquete nlme se proporcionan diversos efectos para modelos lineales y no lineales.

Regresión Local aproximada:

La función standard loess() aproxima una regresión no paramétrica mediante una regresión local con pesos.

Regresión robusta:

Hay diversas funciones, como lqs, rlm en el paquete mass para ajustar modelos de regresión minimizando el efecto de los outliers en los datos.

Modelos aditivos:

Esta técnica permite construir una función de regresión mediante la suma “suavizada” de ciertas funciones .Entre ellas están las funciones avas y ace en el paquete acepack, y las funciones bruto y mars en el paquete mda.

Modelos Arborescentes:

En lugar de buscar un modelo explícito lineal ,estos modelos bifurcan recursivamente los datos en puntos críticos de las variables, buscando particionar los datos en grupos que sean lo más homogéneos posibles entre ellos, y que los grupos sean lo más heterogéneos posibles entre sí.

Los modelos son de nuevo especificados de manera lineal. La función tree() y otras funciones genéricas como plot() y text() son apropiadas para mostrar los resultados de forma gráfica.

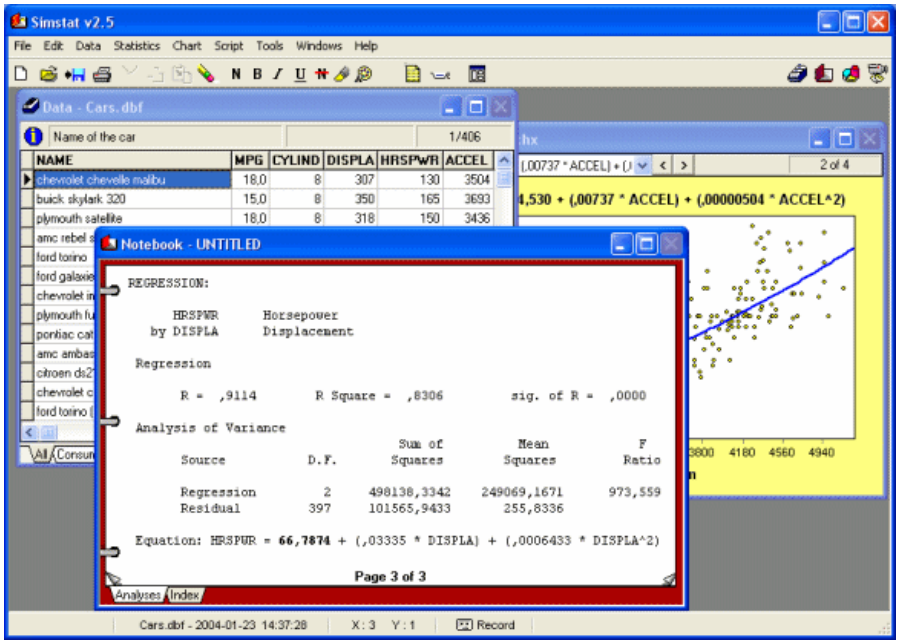
Estos modelos están disponibles en R en los paquetes rpart y tree.

5.4.5. SimStat

Simstat [SSi04] ofrece características innovadoras para manejar los datos de salida, así como un lenguaje de script para automatizar ciertas

tareas y escribir pequeñas aplicaciones. También tiene tutoriales interactivos con recursos multimedia.

El análisis de regresión simple en esta herramienta incluye 7 regresiones no lineales (cuadrática, cúbica, polinomios hasta de 5º grado, logarítmica, exponencial, inversa), ecuación de regresión, análisis de varianza, estadísticos de Durbin-Watson...



El análisis de regresión múltiple incluye 5 métodos diferentes de regresión (hierarchical entry, forward selection, backward elimination, stepwise selection, enter all variables), ANOVA, estadísticos de Durbin-Watson...

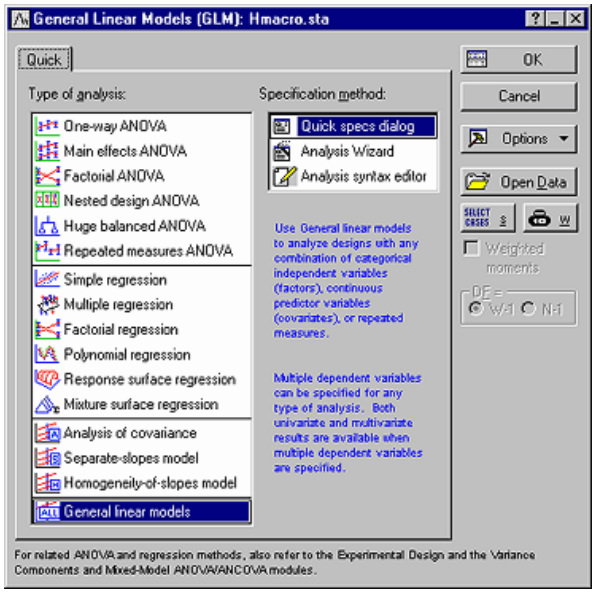
5.4.6. STATISTICA

Statistica [SAM05] ofrece un gran conjunto de herramientas para el manejo, análisis y la visualización de datos, incluyendo procedimientos de data mining. Su tecnología incluye una amplia selección de módulos predictivos, clustering, clasificación y técnicas de exploración en una única plataforma software. Actualmente cuenta con más 20 años de experiencia en el sector.

Esta herramienta está disponible en 4 categorías (Enterprise, Web-based Analytic Applications, Data Mining Solutions, Desktop). La funcionalidad de cada categoría puede ser ampliada con módulos.

Con respecto a la regresión, el módulo base ofrece un conjunto de implementaciones de regresión linear, entre ellas simple, múltiple, paso

a paso, jerárquica, no lineal (polinómica, exponencial, logarítmica...) , regresión contraída, y regresión con estimación por mínimos cuadrados generalizados. Las características de análisis de datos anómalos y de residuos incluyen una amplia selección de gráficos.

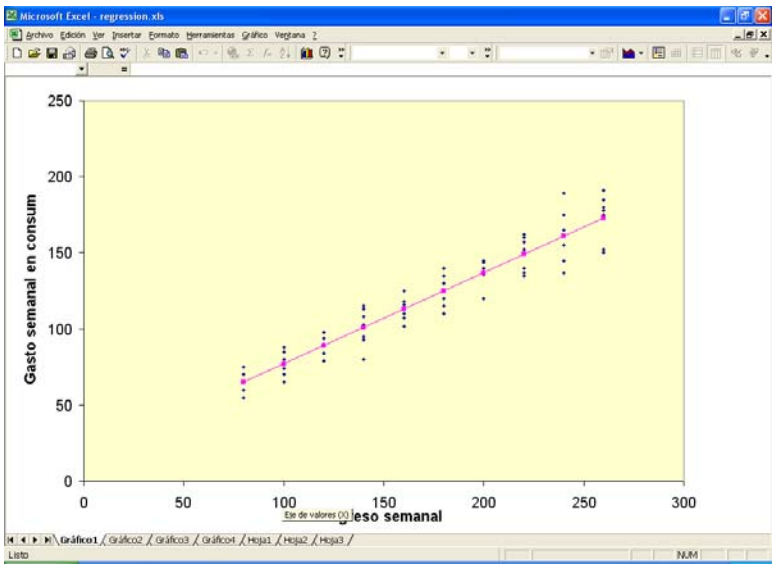


Métodos de regresión más avanzados se proporcionan en el módulo de Regresión General(GRM), como la regresión por subconjuntos, regresión paso a paso multivariante, para múltiples variables dependientes...

Otro módulo de interés es el de Modelos Avanzados, que ofrece una amplia elección de herramientas de modelado y previsión (por ej. modelos lineales, modelos lineales/no lineales generalizados, regresión Quick Logit/Probit, ANOVA/ANCOVA, análisis de supervivencia, series cronológicas y previsión), incluyendo selección automática de modelos y herramientas de visualización interactivas. Con este módulo se puede estimar prácticamente cualquier modelo no lineal definido por el usuario y caracterizar un conjunto de modelos predefinidos.

5.4.7. Microsoft Excel

Excel [IPE03], una hoja de cálculo integrada en el paquete Microsoft Office, incluye algunas utilidades para el análisis estadístico de datos [DTC02].



Dentro del Menú Herramientas , en "Análisis de datos..." están disponibles "Coeficiente de correlación" y "Regresión", los que permiten aplicarse a conjuntos adecuados de datos. Se pueden ajustar modelos de regresión múltiple y hacer un análisis exhaustivo del ajuste del modelo, validez de los parámetros y comportamiento de los residuos. Aunque ciertamente no tiene la facilidad de otros programas para definir los términos del modelo.

5.4.8. MATLAB

MATLAB [MAT04] es un lenguaje de computación técnica de alto nivel y un entorno interactivo para el análisis de datos y el desarrollo de algoritmos y aplicaciones. MATLAB puede emplearse en un amplio rango de aplicaciones, incluyendo procesamiento de señal e imagen, comunicaciones, control digital, modelado financiero, análisis estadístico, biología computacional...

Diversas herramientas y add-on (colecciones de funciones de propósito específico, disponibles por separado) extienden el entorno base para resolver problemas específicos de ciertas áreas. Entre ellos se encuentra el "Statistics Toolbox", el cual da soporte a una gran variedad de tareas estadísticas comunes, desde la generación de números aleatorios al ajuste de curvas [STU05]. Este add-on distingue dos categorías de herramientas:

- Herramientas de probabilidad y estadística.

  Se compone de funciones que pueden ser llamadas desde la linea de comandos. Pueden ser editadas a gusto del usuario.

- Herramientas relacionadas con el aspecto gráfico e interfaces(GUI).

En la primera categoría, y dentro del área que nos ocupa, se encuentran:

- **Modelos lineales**: ANOVA, análisis de covarianza (ANOCOVA), regresión lineal múltiple, regresión por pasos, superficies de predicción, análisis multivariante de la varianza (MANOVA). También da soporte a versiones no paramétricas de ANOVA.

- **Modelos no lineales**: Para lo modelos no lineales, este add-on provee de funciones para la estimación de parámetros, predicción interactiva y visualización de ajustes no lineales multidimensionales, intervalos de confianza para predicciones de valores y parámetros. También contiene funciones para la clasificación y el establecimiento de árboles de regresión para aproximar cualquier tipo de relación que pudiera derivarse de ella.

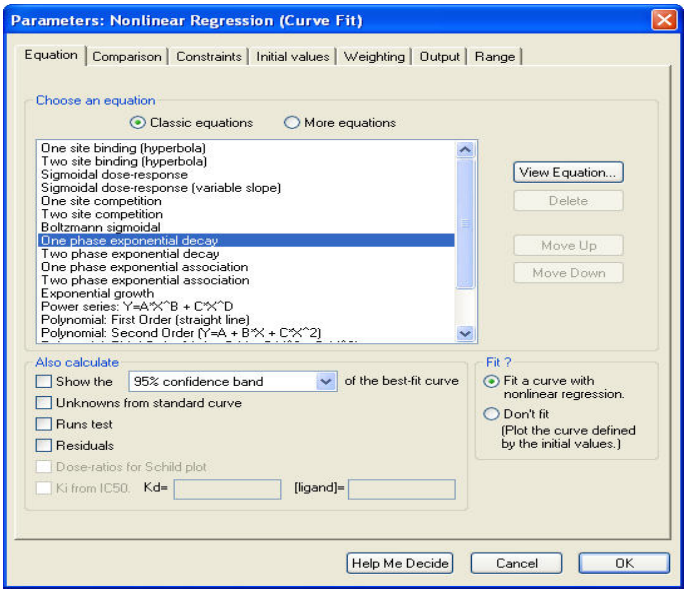
Otras áreas de la estadística a las que da soporte este add-on son: distribuciones probabilísticas, estadística descriptiva, test de hipótesis, estadística multivariante, control de procesos (SPC), Modelos de Markov...

5.4.9. 

GraphPad Prism [GSP05] es una potente combinación de bioestadística básica, ajuste de curvas y graficación científica en una sola aplicación. Ofrece ayuda para organizar, analizar y mostrar experimentos; escogiendo el apropiado test estadístico e interpretando los resultados, es adecuado para satisfacer las necesidades del campo de las ciencias de la salud.

Diseñado para investigadores, Prism no exige profundos conocimientos estadísticos, sino que guía al usuario a través del proceso de análisis, informando tanto como necesite, organizando el trabajo realizado de manera única. La filosofía de Prism es que el usuario centre su esfuerzo en los datos, no en el uso del programa.

Para realizar el ajuste de curvas, una vez introducidos los datos, se puede o bien escoger un modelo entre los 15 que ofrece el programa (los más frecuentes en el campo de la biología) o bien introducir una ecuación. Ésta puede ser no lineal, pero está sujeta a una serie de restricciones impuestas por el programa, ajustándose a una serie de reglas. También pueden escogerse diferentes pesos para los puntos.



Una vez realizado el análisis, Prism memoriza las selecciones escogidas, de manera que para repetir el experimento basta con reemplazar los datos, y la herramienta se encarga de repetir el análisis y redibujar el gráfico.

5.4.10. FINDGRAPH

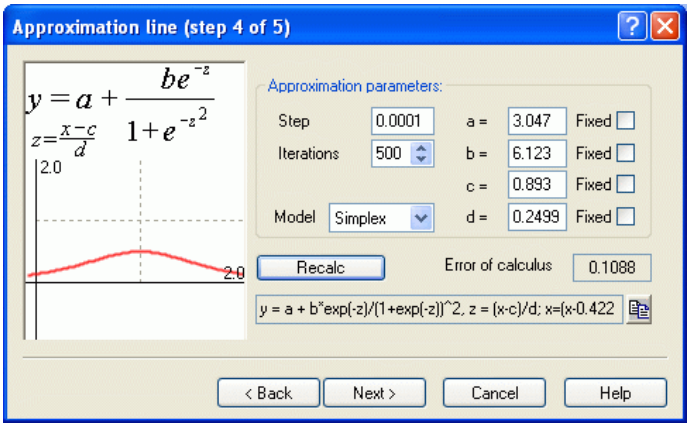
FindGraph [FGL05] es una herramienta científica y de ingeniería que simplifica la tarea de ajuste de curvas y análisis de regresión. Una de sus principales ventajas es su facilidad de uso.

Provee fácilmente de un camino para determinar los parámetros que ofrezcan un mejor ajuste para el modelo de regresión lineal.

En Findgraph, el modelo de regresión lineal es una combinación lineal de funciones $f_{jk}(X)$ polinomiales, racionales, logarítmicas, exponenciales o de Fourier.

Los parámetros Aj son estimados por el método de mínimos cuadrados para minimizar la diferencia entre el modelo y los datos. El asistente de Aproximación puede ser usado para encontrar la mejor ecuación y obtener un informe de resultados rápidamente.

FindGraph usa algunos algoritmos para ajuste de curvas no lineales por mínimos cuadrados. El método simplex y del gradiente son usados por su rapidez. El usuario puede introducir la fórmula deseada (función analítica) y variar el número de parámetros.



El usuario puede incluir, si lo desea, sus propias ecuaciones y modelos de ajuste de curvas.

5.4.11. FuReA

FuReA [FRA03] es una herramienta para el análisis de regresión borroso, capaz de trabajar con datos imprecisos o inciertos.

La lógica borrosa es una técnica ampliamente usada con un gran número de aplicaciones. La teoría de conjuntos borrosos, en la cual se basa Furea, fue presentada por el Prof. Lotfi A.Zadeh en 1965.

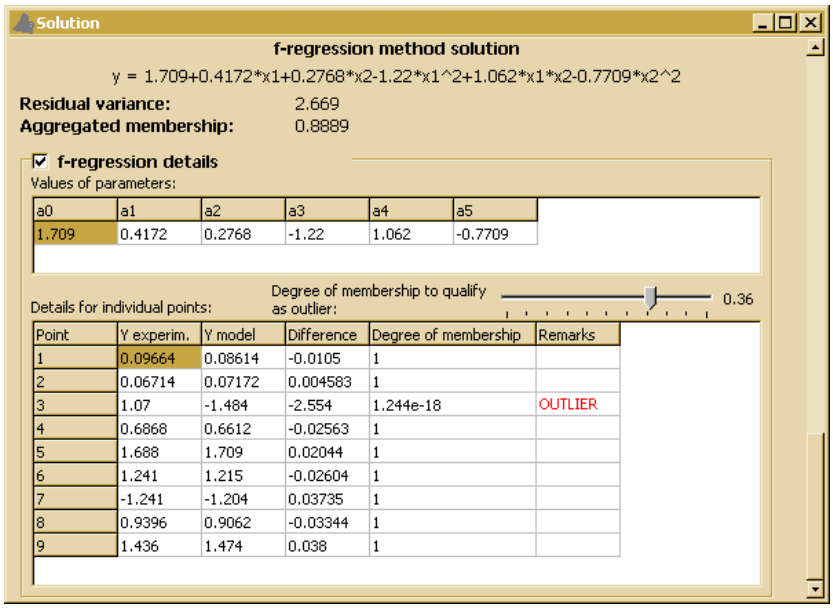
La idea original de análisis de regresión borrosa fue propuesta por Tanaka et al. en 1982 [TSK82] .

El análisis de regresión borrosa asume que alguno de los componentes del sistema es descrito mediante conjuntos borrosos, más concretamente mediante números borrosos.

Un conjunto borroso es aquel en el cual cada elemento posee un cierto grado de pertenencia al conjunto. En el análisis de regresión, los outliers siempre suponen una dificultad por el hecho de que modelan errores. Pero bajo ciertas circunstancias, también pueden ofrecernos valiosa información. Este es el motivo por el que la tarea de identificación y análisis de outliers presenta un doble interés desde el punto de vista técnico.

Se ha demostrado que el método de f-regresión empleado en Furea ofrece una mejor capacidad de detección de outliers y puede ser aplicado satisfactoriamente al análisis de regresión borrosa.

Furea permite introducir datos experimentales y añadir variables obtenidas indirectamente mediante transformaciones. También es posible construir arbitrariamente modelos lineales o no lineales usando variable definidas, y realizar cálculos que den valores más precisos de los parámetros del modelo y encontrar outliers.



Todo esto permite observar como el modelo calculado se ajusta a los datos experimentales.

5.5. Limitaciones

Según lo anteriormente expuesto, estas herramientas presentan una serie de limitaciones para el tratamiento de la imprecisión, la vaguedad y la incertidumbre de la información:

- Están sometidas a procesos transparentes, que no pueden ser controlados por el usuario.

- La propia potencia o rapidez de ejecución restringe el uso de otros procedimientos de la literatura del área, empleando la herramienta en cuestión métodos específicos implementados por los creadores de la misma.

- El desarrollo de herramientas de fácil acceso se ha enfocado en arquitecturas monoprocesador, lo que impide el empleo de éstas en proyectos de gran envergadura que exigen una gran cantidad de cómputo y el uso de paralelismo.

- Las herramientas expuestas en esta sección o bien no tratan la borrosidad, o bien no permiten una suficiente diversidad de métodos y modelos de regresión.

Por estos motivos se considera la necesidad de desarrollar una herramienta práctica que considere todas estas limitaciones.

6. Estudio y resolución del problema

6.1. Modelo Propuesto

6.1.1. Introducción

Dadas un serie de observaciones como las representadas en la tabla 1 y dado el modelo matemático paramétrico de una curva tal que:

$$y = f(\vec{x}, \vec{t}) \qquad (1)$$

siendo t un vector de parámetros:

$$\vec{t} = (t_1, ..., t_k)$$

donde x es un vector de variables independientes e y es la variable dependiente, se denomina regresión a la búsqueda del vector de parámetros t que ofrezca el mejor ajuste para todas las observaciones p_i tal que

$$p_i = (y_i, \vec{x}_i)$$
$$i=1..m$$

siendo

$$\vec{x}_i = (x_{i1}, ..., x_{in})$$
$$i=1...m$$

Observación	Salida (y)	Entradas ($\vec{x}$)
p_1	y_1	$x_{11},\ldots, x_{1n}$
p_2	y_2	$x_{21},\ldots, x_{2n}$
. . .	. . .	. . .
p_m	y_m	$x_{m1},\ldots, x_{mn}$

Tabla 1. Datos de entrada y salida de una serie de observaciones

el enfoque fundamental para realizar la regresión es encontrar la **relación** entre los datos de entrada y salida, decidiendo cuál es el mejor ajuste de los parámetros de la curva dada por la ecuación (1), que es el modelo que define dicha relación.

Existen dos aproximaciones al proceso de regresión los datos de las observaciones que se posean:

- Si se existen datos suficientes referentes a las observaciones se puede obtener su distribución, su media, su varianza, etc. En este caso se usan modelos no simbólicos.

- Si los datos son insuficientes o las fuentes de la información son imperfectas, aún se puede hallar una relación entre las entradas y las salidas. En este caso se usan modelos simbólicos.

Tanto la bondad del modelo de regresión o de la curva que representa la relación como la del ajuste suelen estimarse por la minimización de la **distancia** entre el valor esperado y el observado, es decir, entre los valores de las observaciones y la curva obtenida aplicando el vector de parámetros al modelo de curva.

6.1.2. Método básico de la resolución de la regresión lineal (Método de mínimos cuadrados o LSM)

El proceso de regresión usando el método de mínimos cuadrados consiste en obtener un vector de parámetros

$$\vec{t} = (t_1, ..., t_k)$$

que minimicen r en la expresión

$$r = \sum_{i=1}^{m} \left((y_i - f(\vec{x}_i, \vec{t}))^2 \right)$$

siendo p_i la i-ésima observación (tabla 1).

Es decir, obtener el mínimo en t para la expresión r:

$$\min_{\vec{t}}(r) = \min_{\vec{t}} \left\{ \sum_{i=1}^{m} \left((y_i - f(\vec{x}_i, \vec{t}))^2 \right) \right\}$$

6.1.3. La distancia en el modelo de regresión

La distancia se define como una función binaria

$$\delta(a,b): X^2 \longrightarrow R$$

que verifica las siguientes condiciones [Cas02]:

- No negatividad. $\delta(a,b) \geq 0 \ \forall a,b \in X$
- Conmutatividad. $\delta(a,b) = \delta(b,a) \ \forall a,b \in X$
- Desigualdad triangular. $\delta(a,b) \leq \delta(a,c) + \delta(c,b) \ \forall a,b,c \in X$

En este proyecto se define el método de obtención de la distancia entre cada punto observado y el modelo de curva como:

$$\delta(p_i, f(\vec{x}, \vec{t})) \qquad (2)$$

siendo p_i el punto:

$$p_i = (y_i, \vec{x}_i)$$

y el modelo de curva

$$f(\vec{x}, \vec{t})$$

En el modelo que se propone, esta distancia puede ser cualquier función como las expuestas en el punto 6.1.1.

Sirva como ejemplo la utilizada en la regresión clásica usando el método de mínimos cuadrados (LSM):

$$\delta_{LSM}(p_i, f(\vec{x}, \vec{t})) = |y_i - f(\vec{x}_i, \vec{t})|$$

donde y_i es el valor observado (la variable dependiente del punto p_i) y $f(\vec{x}_i, \vec{t})$ es el valor esperado o pronosticado.

Diferentes métodos para el cálculo de la distancia nítida

Dentro de esta definición de distancia encajan distintos métodos para obtenerla [PID05], [Bla05]:

- Distancia discreta:

$$\delta(\vec{a}, \vec{b}) = \begin{cases} 0 \text{ si } \vec{a} = \vec{b} \\ 1 \text{ en otro caso} \end{cases}$$

- Distancia euclídea:

$$\delta(\vec{a},\vec{b}) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

- Distancia L_m:

$$\delta(\vec{a},\vec{b}) = \sqrt[m]{\sum_{i=1}^{n}(a_i - b_i)^m}$$

- Distancia de Mahalanobis:

$$\delta(\vec{a},\vec{b})^2 = \frac{\sum_{i=1}^{n}(a_i - b_i)^2}{\sigma_{ab}}$$

  Hay que reseñar sobre esta distancia que no cumple la propiedad de desigualdad triangular por lo que no es una distancia en un espacio métrico.

- Distancia Manhattan (absoluta, de bloque o *city-block*):

$$\delta(\vec{a},\vec{b}) = \sum_{i=1}^{n}|a_i - b_i|$$

- Distancia de Chebishev:

$$\delta(\vec{a},\vec{b}) = \max_{i=1}^{n}\{|a_i - b_i|\}$$

- Distancia de Fréchet:

$$\delta(\vec{a},\vec{b}) = \sum_{i=1}^{n}\frac{|a-b|}{\sigma_i}$$

6.1.4. El problema de la selección de una distancia representativa.

Como se ha expuesto anteriormente, se necesita hallar un vector de parámetros

$$\vec{t} = (t_1,...,t_k)$$

tal que minimice la distancia del conjunto de observaciones al modelo de curva.

Dado un determinado $\vec{t}$, se puede obtener todo un vector de distancias

$$\vec{d} = (d_1, \ldots, d_m)$$

en el que cada d_i es la distancia existente entre la observación p_i y el modelo de curva:

$$d_i = \delta(p_i, f(\vec{x}, \vec{t}))$$

Distintos modelos de expresión de la distancia

Seleccionado un determinado modelo paramétrico de curva $f_k(\vec{x}, \vec{t})$ se puede expresar la distancia con distintos modos de expresión de la misma.

En el caso del LSM se utiliza la suma de los cuadrados de cada d_i:

$$r = \sum_{i=1}^{n} d_i^2 \qquad (3)$$

El modelo propuesto en la herramienta permite el uso de funciones con operadores matemáticos de agregación para la expresión de la distancia.

Los operadores de agregación son objetos matemáticos cuya función es reducir un conjunto de números a un único número representativo (o significativo) [BCF05] [Det01].

Se consigue así reducir a un solo valor representativo el conjunto de distancias obtenidas al aplicar la función de distancia (véase la ecuación (2)) a cada una de las muestras recogidas en la tabla 1, como se formula en la ecuación (3) para el LSM.

Operadores propuestos para la expresión de la distancia

Existen diversos y variados agregadores, que podrán ser utilizados en la aplicación como modelos de expresión de la distancia.

A continuación se recoge una breve selección de los mismos:

- Agregador media:

$$\underset{i=1}{\overset{m}{\Omega}}_{Media}(d_i) = \frac{\sum_{i=1}^{m} d_i}{m}$$

- Agregador suma de cuadrados:

  Este agregador es el utilizado en el LSM

$$\underset{i=1}{\overset{m}{\Omega}}_{LSM}(d_i) = \sum_{i=1}^{m} d_i^2$$

- Agregador suma:

$$\underset{i=1}{\overset{m}{\Omega}}_{Suma}(d_i) = \sum_{i=1}^{m} d_i$$

- Agregador mínimo:

$$\underset{i=1}{\overset{m}{\Omega}}_{Min}(d_i) = \min_{i=1}^{m}(d_i)$$

- Agregador máximo:

$$\underset{i=1}{\overset{m}{\Omega}}_{Max}(d_i) = \max_{i=1}^{m}(d_i)$$

- Agregador OWA [Det01]:

$$\underset{i=1}{\overset{m}{\Omega}}_{Max}(d_i) = \sum_{i=1}^{m} w_i \cdot d_{\sigma(i)}$$

  donde la función σ es una permutación que ordena los elementos d de tal forma que $d_{\sigma(1)} \leq d_{\sigma(2)} \leq ... \leq d_{\sigma(m)}$.

Más que un operador de agregación, el operador OWA da nombre a toda una familia de operadores matemáticos parametrizados de agregación que incluyen a algunos de los anteriormente mencionados.

Por ejemplo, en el caso del mínimo, se trataría de un operador OWA cuyos pesos serían $w_1 = 1$ y $w_i = 0$ con $i \neq 1$

En el modelo propuesto, esta función de agregación sobre las distancias se representará con el símbolo Ω_d:

$$\min_{\vec{t}}(r) = \min_{\vec{t}}\left(\underset{i=1}{\overset{m}{\Omega}}_{d}\left(\delta(p_i, f(\vec{x},\vec{t}))\right)\right)$$

6.1.5. Ponderación de los puntos

En el modelo de la aplicación se permite al usuario dar distintas valoraciones o ponderaciones a los puntos de forma que cada uno tenga diferente influencia en el proceso de regresión [BCF05].

Este objetivo se consigue utilizando un operador de ponderación, notado en adelante como Γ, que se aplicará a cada una de las distancias d_i antes de aplicar el operador de agregación de las distancias:

$$r = \underset{i=1}{\overset{m}{\Omega}}\left(\Gamma\left(d_i\right)\right) \qquad (4)$$

El uso de la ponderación en el modelo de regresión nítido

A partir del modelo obtenido en la ecuación (4) se obtiene el modelo para realizar la regresión:

$$\min_{\vec{t}}\left(r\right) = \min_{\vec{t}}\left(\underset{i=1}{\overset{m}{\Omega}}{}_{d}\left(\gamma_i\right)\right)$$

siendo cada γ_i el resultado de aplicar el operador de ponderación Γ a cada una de las distancias d_i:

$$\gamma_i = \Gamma\left(d_i\right)$$

donde d_i es la distancia existente entre el punto p_i y la curva definida por el modelo $f\left(\vec{x},\vec{t}\right)$:

$$d_i = \delta\left(p_i, f\left(\vec{x},\vec{t}\right)\right)$$

Es decir, el mínimo en t de la expresión r:

$$\min_{\vec{t}}(r) = \min_{\vec{t}}\left\{\Omega_{d\,i=1}^{m}\left(\Gamma\left(\delta\left(p_i, f\left(\vec{x},\vec{t}\right)\right)\right)\right)\right\} \quad (5)$$

o lo que es lo mismo, hallar un vector de parámetros $\vec{t}$ tal que minimice la agregación de las distancias ponderadas de cada uno de los puntos p_i a la curva $f\left(\vec{x},\vec{t}\right)$.

6.1.6. Introducción de la imprecisión en la información con el uso de la Teoría de Conjuntos Borrosos y marcos matemáticos relacionados

Como se indicó en el punto 4.1, la modelización del mundo real es inherentemente imprecisa, incierta y vaga.

La modelización de la información aportada por el ser humano precisa de modelos que estimen estas diferentes formas de imprecisión. En esta herramienta se propone para su modelización el uso de la Teoría de Conjuntos Borrosos y marcos matemáticos relacionados.

En el modelo de regresión existen diversos valores en los que puede haber imprecisión. Estos son las entradas[1], las salidas[2], los parámetros de la curva, la distancia entre los puntos y la curva y la ponderación.

Para modelizar la vaguedad se propone la borrosificación de estos valores.

El significado semántico de la borrosificación de estos parámetros se deja abierto al sentido que quiera darles el usuario de la aplicación.

Números borrosos

Para dar borrosidad a un valor nítido, se le asignará una función de pertenencia μ:

$$\mu_{x_\beta}(x) \in [0,1]$$

[1] Se entienden como entradas los valores observados de las variables independientes, es decir, cada uno de los x_{ij} que aparecen en la tabla 1.

[2] Se entienden como salidas los valores observados para la variable dependiente, es decir, las y_i que aparecen en la tabla 1.

El único requisito que debe cumplir el grado de pertenencia μ es que su rango debe ser el intervalo [0,1]

En adelante, la borrosidad se notará con un subíndice β.

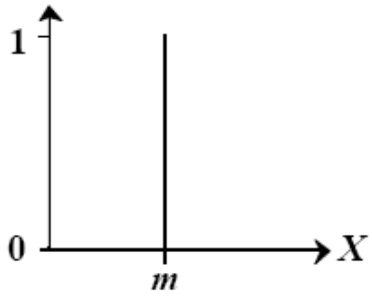
Distintas funciones para expresar el grado de pertenencia

Algunas de las funciones de pertenencia encontradas más frecuentemente en la literatura son [Gal05], [Car01]:

- Singleton:

  Está definida por un valor m, para el cual el valor de la función es uno, siendo 0 para los demás:

$$\mu(x)=\begin{cases}1 & \text{si } x=m\\ 0 & \text{si } x\neq m\end{cases}$$



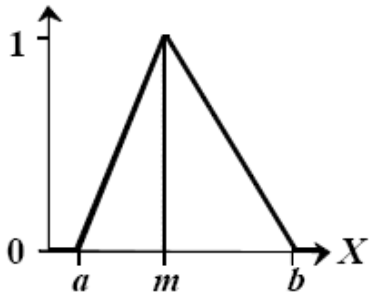
  Un número nítido puede definirse como un número borroso con una función de pertenencia singleton asociada, cuyo m es el valor del número nítido [ACIS03]:

$$a=7 \qquad \mu_a(x)=\begin{cases}1 & \text{si } x=7\\ 0 & \text{si } x\neq 7\end{cases} \qquad (6)$$

- Triangular:

  Definida por sus límites superior e inferior a y b y un valor modal m tal que a<m<b

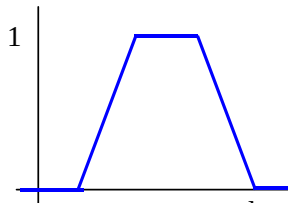
$$\mu(x)=\begin{cases}0 & \text{si } x \le a \text{ o } x \ge b \\ \dfrac{x-a}{m-a} & \text{si } x \in (a,m] \\ \dfrac{b-x}{b-m} & \text{si } x \in (m,b)\end{cases}$$



- Trapezoidal:

  Definida por cuatro parámetros a, b, c y d tales que a < b < c < d

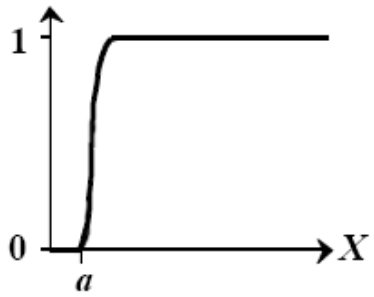
$$\mu(x)=\begin{cases}0 & \text{para } x \le a \\ \dfrac{x-a}{b-a} & \text{para } a < x \le b \\ 1 & \text{para } b < x \le c \\ \dfrac{d-x}{b-c} & \textit{para } c < x \le d \\ 0 & \text{para } x > d\end{cases}$$



- Gamma:

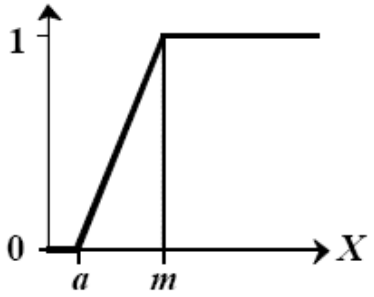
  Esta función queda definida por su límite inferior a y un valor k > 0.
  A mayor k más rápido es el crecimiento de la función.

$$\mu(x)=\begin{cases}0 & \text{si } x \le a \\ 1-e^{-k(x-a)^2} & \text{si } x > a\end{cases}$$



Se puede aproximar linealmente mediante la siguiente función:

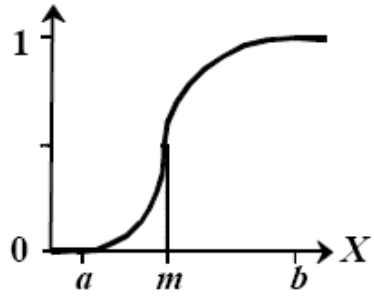
$$\mu(x)=\begin{cases}0 & \text{si } x \le a \\ \dfrac{x-a}{m-a} & \text{si } a < x < m \\ 1 & \text{si } x \ge m\end{cases}$$



- Función S:

  Definida por sus límites superior e inferior a y b y un punto de inflexión m tal que a<m<b

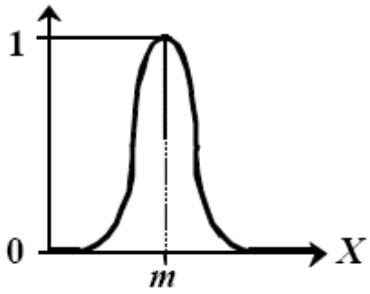
$$\mu(x)=\begin{cases}0 & \text{si } x \le a \text{ o } x \ge b \\ 2\cdot\left(\dfrac{x-a}{b-a}\right)^2 & \text{si } x \in (a,m] \\ 1-2\cdot\left(\dfrac{b-x}{b-m}\right)^2 & \text{si } x \in (m,b)\end{cases}$$



- Gaussiana:

  Definida por su valor medio m y un valor k > 0. Tiene forma de campana, siendo más estrecha cuanto mayor es el valor k.

$$\mu(x) = e^{-k(x-m)^2}$$



Puntos borrosos

Un punto se considerará borroso si alguna de sus componentes es borrosa. En ese caso, el punto se notará como $p_{\beta i}$:

$$p_{\beta i} = \left(y_{\beta i}, \vec{x}_{\beta i}\right)$$

siendo

$$\vec{x}_{\beta i} = \left(x_{\beta i 1}, \ldots, x_{\beta i n}\right)$$

y las funciones de pertenencia asociadas

$$\mu_{x_{\beta ij}}(x)$$
$$j=1\ldots n$$

y

$$\mu_{y_{\beta i}}(x)$$

En el caso de que la componente sea nítida, la función de pertenencia asociada será una singleton con un valor para su parámetro m igual al valor de la componente (véase (6)).

A efectos de conseguir mayor claridad en las fórmulas, dado un punto $p_{\beta i}$, la componente $y_{\beta i}$ se notará como $p_{\beta i0}$, y cada $x_{\beta ij}$ como $p_{\beta ij}$, por lo que un punto puede expresarse como:

$$p_{\beta i} = (y_{\beta i}, x_{\beta i1}, \ldots, x_{\beta in}) = (p_{\beta i0}, p_{\beta i1}, \ldots, p_{\beta in}) \quad (7)$$

La función de pertenencia de un punto borroso $P_{\beta i}$ se puede considerar como una agregación de las funciones de pertenencia de cada uno de los números borrosos que lo componen. Así pues, cada punto borroso tendrá asociado un agregador (Ω_{pi}) para obtener el valor de su función de pertenencia. Esta se calculará de la siguiente manera:

$$\mu_{P_{\beta i}}(p) = \Omega_{Pi}^{n}{}_{j=0} \left(\mu_{p_{\beta ij}}(p_j) \right)$$

siendo p un punto nítido de la forma:

$$p = (y, x_1, \ldots, x_n) = (p_0, p_1, \ldots, p_n)$$

y $p_{\beta i}$ un punto borroso como se expuso en (7).

6.1.7. Introducción de imprecisión en las entradas y las salidas

El modelo de esta aplicación abierta permite asignar a cada una de las entradas o a la salida una función de pertenencia para borrosificar su valor.

Esto puede modelar, por ejemplo, cierta imprecisión o vaguedad en la medida o la medición de los datos de entrada o salida.

En ese caso, cada una de las observaciones p_i es en realidad un punto borroso $p_{\beta i}$ de la forma referida en la ecuación (7).

Como se puede observar en la siguiente ecuación, el modelo es análogo al propuesto en el apartado 6.5 para la regresión nítida (véase (5)).

$$\min_{\vec{t}}(r) = \min_{\vec{t}}\left\{\Omega_d^m\left(\Gamma\left(\delta\left(p_{\beta i}, f\left(\vec{x},\vec{t}\right)\right)\right)\right)\right\}$$

La diferencia estriba en que, en este caso, la distancia a hallar es con respecto a un punto borroso $p_{\beta i}$, y no nítido.

Cálculo de la distancia de un punto borroso a una curva

Para obtener la distancia de un punto borroso $p_{\beta i}$ a una curva

$$\delta\left(p_{\beta i}, f\left(\vec{x},\vec{t}\right)\right)$$

se calcula la distancia existente entre la curva y cada uno de los puntos que forman el soporte[3] de $p_{\beta i}$.

Para calcular esta distancia se puede usar cualquier método existente de cálculo de la distancia, entre ellos, los propuestos en el apartado 5.1.3.

Al conjunto de puntos del soporte le llamaremos s_i

$$s_i = sop\left(p_{\beta i}\right)$$

[3] El soporte de un subconjunto borroso lo forman aquellos elementos que tienen un grado de pertenencia al subconjunto mayor que 0 [And00][Gal05]:

$$sop\left(A_\beta\right) = \left\{x \middle| \mu_{A_\beta}(x) > 0\right\}$$

y $\tilde{d}_i$ al conjunto de distancias entre los elementos del soporte s_i y la curva:

$$\tilde{d}_i = \left\{ d \,\middle|\, d = \delta_n\left(m, f\left(\vec{x},\vec{t}\right)\right) \text{con } m \in s_i \right\}$$

donde δ_n es una función de cálculo de la distancia que puede ser cualquiera de las expuestas en el apartado 6.1.

A continuación, cada uno de los valores del conjunto $\tilde{d}_i$ se pondera con el grado de pertenencia al punto $p_{\beta i}$ del elemento correspondiente en s_i. De esta forma se puede dar más peso en el cálculo de la distancia a los elementos con mayor grado de pertenencia a $p_{\beta i}$. Para realizar esta operación se puede utilizar cualquier operador de ponderación existente. En la aplicación implementada, por ejemplo, se ha utilizado el producto. De esta forma se obtiene un conjunto de distancias ponderadas, $\tilde{d}_{i\,pond}$

$$\tilde{d}_{i\,pond} = \left\{ d \,\middle|\, d = \mu_{p_{\beta i}}(m) \otimes \delta_n\left(m, f\left(\vec{x},\vec{t}\right)\right) \text{con } m \in s_i \right\}$$

Por último, se necesita obtener, a partir del conjunto de distancias ponderadas $\tilde{d}_{i\,pond}$, un valor que sea representativo de todas ellas como valor de distancia entre el punto borroso y la curva. Para ello se aplica un operador de agregación (que se notará como $\Omega_{\delta b}$) sobre los elementos del conjunto, obteniéndose así la distancia definitiva:

$$\delta\left(p_{\beta i}, f\left(\vec{x},\vec{t}\right)\right) = \underset{d \in \tilde{d}_{pond\,i}}{\Omega_{\delta_b}}\left(d\right)$$

Otra forma de expresar esta fórmula es

$$\delta_b\left(p_{\beta i}, f\left(\vec{x},\vec{t}\right)\right) = \underset{m_j \in sop\left(p_{\beta i}\right)}{\Omega_{\delta_b}}\left(\mu_{p_{\beta i}}\left(m_j\right) \otimes \delta_n\left(m_j, f\left(\vec{x},\vec{t}\right)\right)\right)$$

donde

- $p_{\beta i}$ es un punto borroso con función de pertenencia $\mu_{p_{\beta i}}(p)$
- Ω_{δ_b} es un agregador de la función de cálculo de la distancia de un punto borroso a una curva. No debe confundirse este agregador Ω_{δ_b} con Ω_d, la función de agregación utilizada en el modelo de regresión para obtener la agregación de las distancias.
- m_j es cada uno de los puntos pertenencientes al soporte de $p_{\beta i}$.
- la función $\delta_n(m_j, f(\bar{x}, \bar{t}))$ es una función de cálculo de la distancia de un punto nítido a una curva.

6.1.8. Introducción de imprecisión en los parámetros del modelo de curva

Otra fuente de incertidumbre pueden ser los parámetros del modelo de curva a ajustar. En el modelo clásico de regresión, las diferencias entre los valores observados y los estimados se achacan a errores en las observaciones. Por el contrario, en el caso que nos ocupa, se asumirá que estas diferencias vienen motivadas por inexactitudes en la estructura del sistema. Estas desviaciones se considerarán como borrosidad en los parámetros del sistema [TSK82].

Así pues, el modelo de curva será de la forma:

$$f(\vec{x}, \vec{t}_\beta)$$

siendo $\vec{t}_\beta$ el vector de parámetros borrosos de la curva:

$$\vec{t}_\beta = (t_{\beta 1}, ..., t_{\beta k})$$

con funciones de pertenencia asociadas $(\mu_{t_{\beta 1}}, ..., \mu_{t_{\beta k}})$

Por ejemplo, en el caso del modelo de curva $f(\vec{x}, \vec{t}_\beta) = t_{\beta 1} \cdot e^{t_{\beta 2} \cdot sen(x)}$ con $t_{\beta 1}$ y $t_{\beta 2}$ números borrosos, centrados en 1 y con una amplitud de 0.9, se obtiene el siguiente rango de curvas:

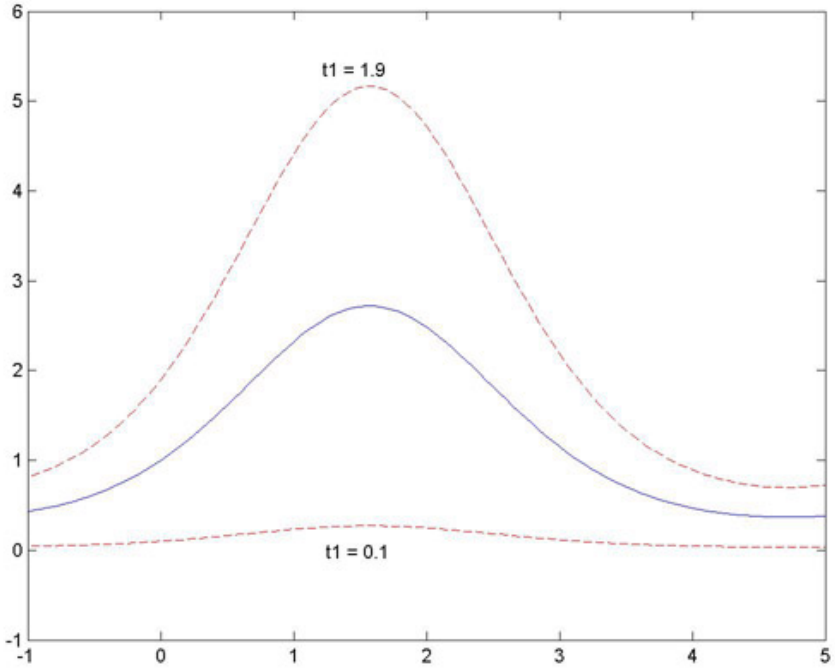


Figura 1. Modelo de curva con parámetros borrosos variando el primero de ellos

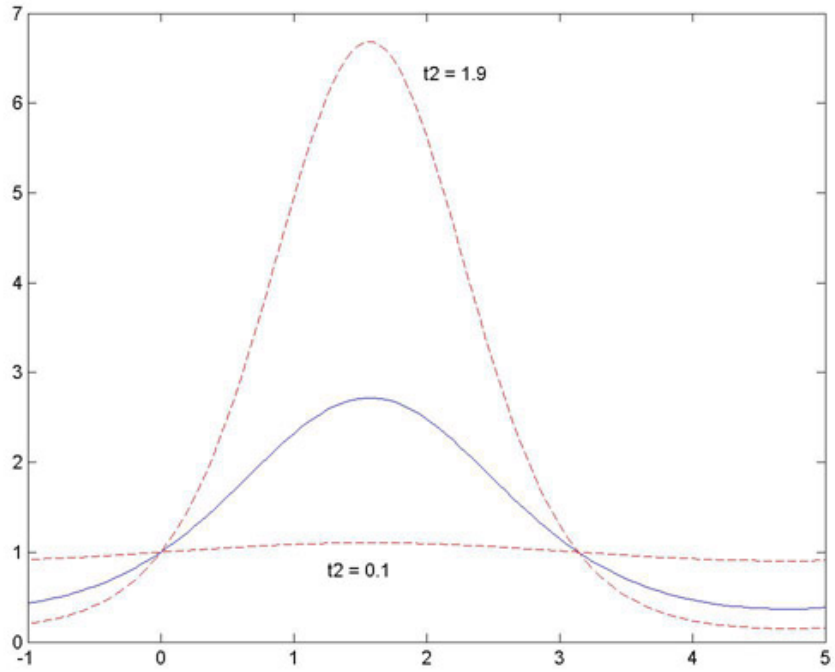


Figura 2. Modelo de curva con parámetros borrosos variando el segundo de ellos

cada una de ellas con un valor de pertenencia al modelo $f(\vec{x},\vec{t}_\beta)$ diferente.

De nuevo esto influye en el cálculo de la distancia punto-curva. La distancia entre un punto nítido y una curva con parámetros borrosos se puede calcular de la siguiente forma:

$$\delta(p, f(\vec{x},\vec{t}_\beta)) = \Omega_{\substack{\delta_b \\ j}} \left(\mu_{t_\beta}(\vec{t}_j) \otimes \delta_n(p, f(\vec{x},\vec{t}_j))\right) \qquad (8)$$
$$\vec{t}_j \in sop(\vec{t}_\beta)$$

donde

- Ω_{δ_b} es un agregador de la función del cálculo de la distancia de un punto a una curva con parámetros borrosos

- $\mu_{t_\beta}(\vec{t}_j)$ es el grado de pertenencia del vector de parámetros nítidos $\vec{t}_j$ al vector de parámetros borrosos $\vec{t}_\beta$.

- la función $\delta_n(p, f(\vec{x}, \vec{t}_j))$ es una función de cálculo de la distancia de un punto nítido a una curva

Si se aplica la ecuación (8) al modelo general de regresión dado por la ecuación (5) se obtiene el modelo de regresión para un modelo de curva con parámetros borrosos:

$$\min_{\vec{t}_\beta}(r) = \min_{\vec{t}_\beta}\left\{\Omega_{i=1}^{m}{}_d\left(\Gamma\left(\delta\left(p_i, f\left(\vec{x}, \vec{t}_\beta\right)\right)\right)\right)\right\}$$

6.1.9. Introducción de imprecisión en entradas, salidas y parámetros del modelo de curva

Para el caso en que tanto los puntos como los parámetros de la curva sean borrosos el cálculo de la distancia se realizará de la siguiente manera:

$$\delta\left(p_\beta, f\left(\vec{x}, \vec{t}_\beta\right)\right) = \Omega_{\substack{\delta_b \\ j,k}}\left(\mu_{\tilde{t}_\beta}\left(\vec{t}_k\right) \otimes \mu_{p_\beta}\left(m_j\right) \otimes \delta_n\left(m_j, \vec{t}_k\right)\right)$$
$$m_j \in sop\left(p_\beta\right)$$
$$t_k \in sop\left(\vec{t}_\beta\right)$$

donde

- Ω_{δ_b} es un agregador de la función del cálculo de la distancia de un punto borroso a una curva con parámetros borrosos

- $\mu_{t_\beta}(\vec{t}_k)$ es el grado de pertenencia del vector de parámetros nítidos $\vec{t}_k$ al vector de parámetros borrosos $\vec{t}_\beta$.

- $\mu_{p_\beta}(m_j)$ es la función de pertenencia del punto nítido m_j a p_β

- la función $\delta_n(m_j, f(\vec{x}, \vec{t}_k))$ es una función de cálculo de la distancia de un punto nítido a una curva

Aplicando esta ecuación al modelo obtenido anteriormente (véase ecuación (5)) se obtiene lo siguiente:

$$\min_{\vec{t}_\beta}(r) = \min_{\vec{t}_\beta}\left\{\bigoplus_{i=1}^{m}{}_d\left(\Gamma\left(\delta\left(p_{\beta i}, f\left(\vec{x},\vec{t}_\beta\right)\right)\right)\right)\right\}$$

Donde $p_{\beta\, i}$ representa cada uno de los puntos borrosos observados y $\vec{t}_\beta$ es el vector de parámetros borrosos del modelo de curva.

6.1.10. Introducción de imprecisión en el cálculo de las distancias

Otra fuente de imperfección en los datos, además de las ya vistas, se puede encontrar en la medida de la distancia.

A pesar de partir de unos puntos nítidos y unos parámetros para la curva también nítidos es posible que la información sobre la distancia que los separa sea sujeto de imprecisiones. Un ejemplo muy claro de esta situación es el caso en el que el cálculo de la distancia se realiza mediante métodos aproximados o de aproximaciones sucesivas. El grado de borrosidad en este caso vendría dado por la fiabilidad que se espera en el resultado del cálculo de la distancia, o por la cantidad de aproximaciones realizadas.

La borrosidad en las distancias se interpreta como una función

$$\delta_\beta\left(p_i, f\left(\vec{x},\vec{t}\right)\right)$$

que dará como resultado un número borroso

$$d_{\beta i} = \delta_\beta\left(p_i, f\left(\vec{x},\vec{t}\right)\right)$$

con una función de pertenencia asociada

$$\mu_{d_{\beta i}}(d)$$

En este caso, se procede a desborrosificar dicho número borroso que expresa la distancia antes de la aplicación del ponderador en el modelo.

Existen gran cantidad de técnicas de desborrosificación, siendo las más comunes el máximo, el centroide o integral de Sugeno, y la media ponderada [TJR95].

En el modelo presentado para esta aplicación no se propone ningún método concreto, sino que se deja libertad de elección al usuario de la aplicación para que escoja la técnica más conveniente.

Se propone como método experimental el siguiente método de desborrosificación:

Dado $d_{\beta i}$, un número borroso que expresa una distancia borrosa entre el punto p_i y el modelo de curva, se propone la desborrosificación de $d_{\beta i}$ mediante la agregación de cada uno de los elementos que forman su soporte, ponderados por su grado de pertenencia al conjunto borroso $d_{\beta i}$:

$$d_i = \Omega\left(\mu_{d_{\beta i}}(d_{ij}) \otimes d_{ij}\right)$$

con

$$d_{ij} \in sop\left(d_{\beta i}\right)$$

Este método deja abierta la utilización de diferentes operadores de agregación para obtener los resultados que mejor se adapten a las necesidades del usuario.

Aplicando esta fórmula al modelo general se obtiene que el valor a minimizar es:

$$\min_{\bar{t}}(r) = \min_{\bar{t}}\left\{\Omega_{i=1}^{m}{}_{d}\left(\Gamma(d_i)\right)\right\}$$

siendo d_i el valor obtenido al desborrosificar la distancia $d_{\beta i}$

6.2. Primera aproximación

En esta sección se presenta el diseño de una “herramienta abierta” de regresión borrosa.

Esta herramienta es una extensión de la regresión paramétrica con números nítidos, que se generaliza haciendo uso de las técnicas de la Teoría de Conjuntos Borrosos y marcos matemáticos relacionados

El objetivo principal planteado, que sea totalmente abierta, se concreta en que no presente limitaciones de ningún tipo en cuanto a incluir borrosidad, como ha quedado explicado en el capítulo 5.1.

Este uso de las técnicas de la teoría de conjuntos borrosos debe poder concretarse tanto en los datos como en los procesos del procedimiento, no debe limitar los modelos a estudiar ni por su forma ni por el método que se desee emplear para mejorar su ajuste y debe permitir obtener las metas elegidas de predicción y descripción sin inconvenientes e inciertos pasos que oscurezcan el desarrollo u obtención de los parámetros de mejor ajuste del modelo o modelos propuestos.

A continuación se hace un estudio detallado de la aplicación desarrollada mediante la descripción de los paquetes que la componen.

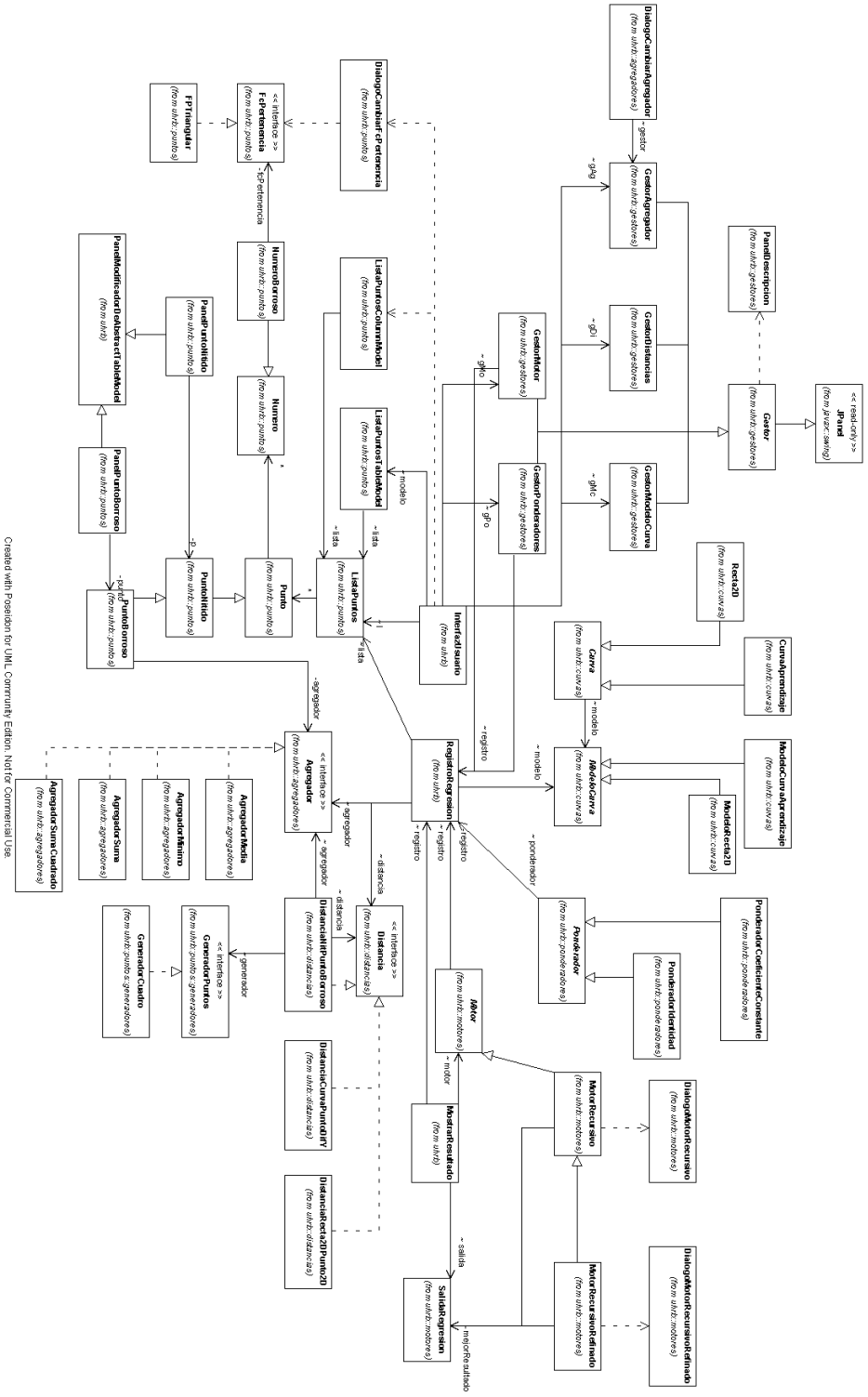
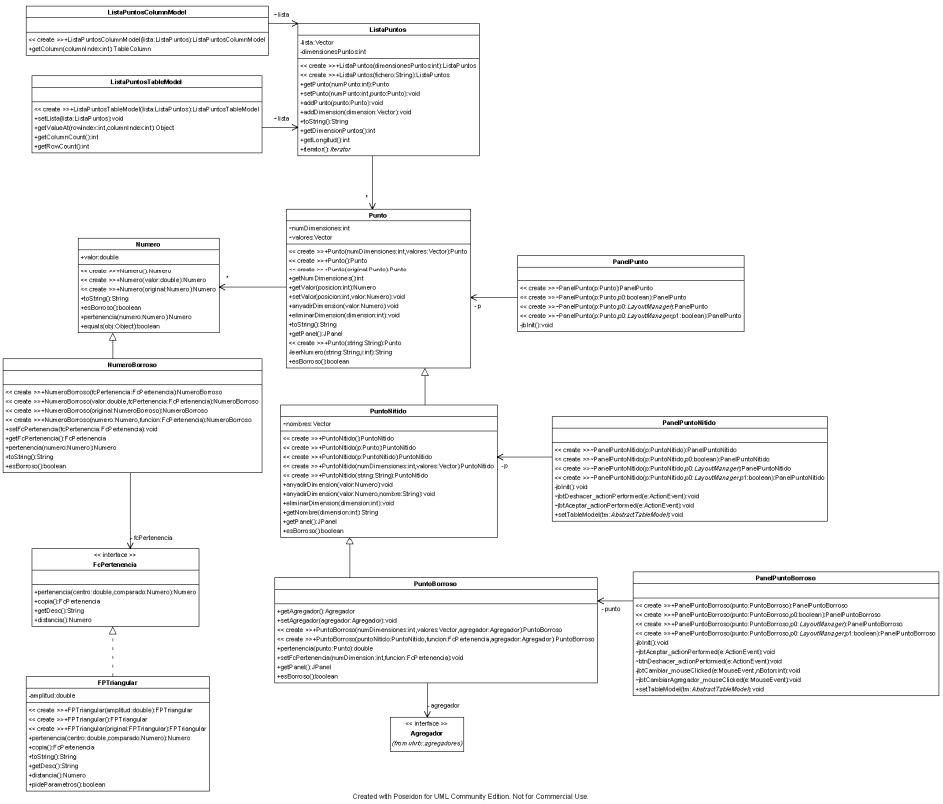


Figura 3. Modelo de clases completo

6.2.1. Puntos.



Este paquete contendrá las implementaciones de las clases que representan los puntos, nítidos y borrosos, tanto las que actualmente están implementadas como las que incorporen futuros grupos de trabajos de Sistemas Informáticos de esta facultad que continúen el trabajo de investigación, o colaboradores externos. También se incluyen en él otras clases utilizadas para su mantenimiento o representación gráfica.

Todos los puntos heredan de la clase abstracta **Punto**, en la que se define el comportamiento que debe poseer un punto en esta aplicación.

Esta clase permite implementar diferentes tipos de números borrosos, según su función de pertenencia, ya sea triangular, L-type, parabólica, etc. También permite representar puntos nítidos.

Se han realizado dos implementaciones de los puntos, una para puntos nítidos y otra para puntos borrosos. Para ello se usan las clases **Numero** y **NumeroBorroso** respectivamente.

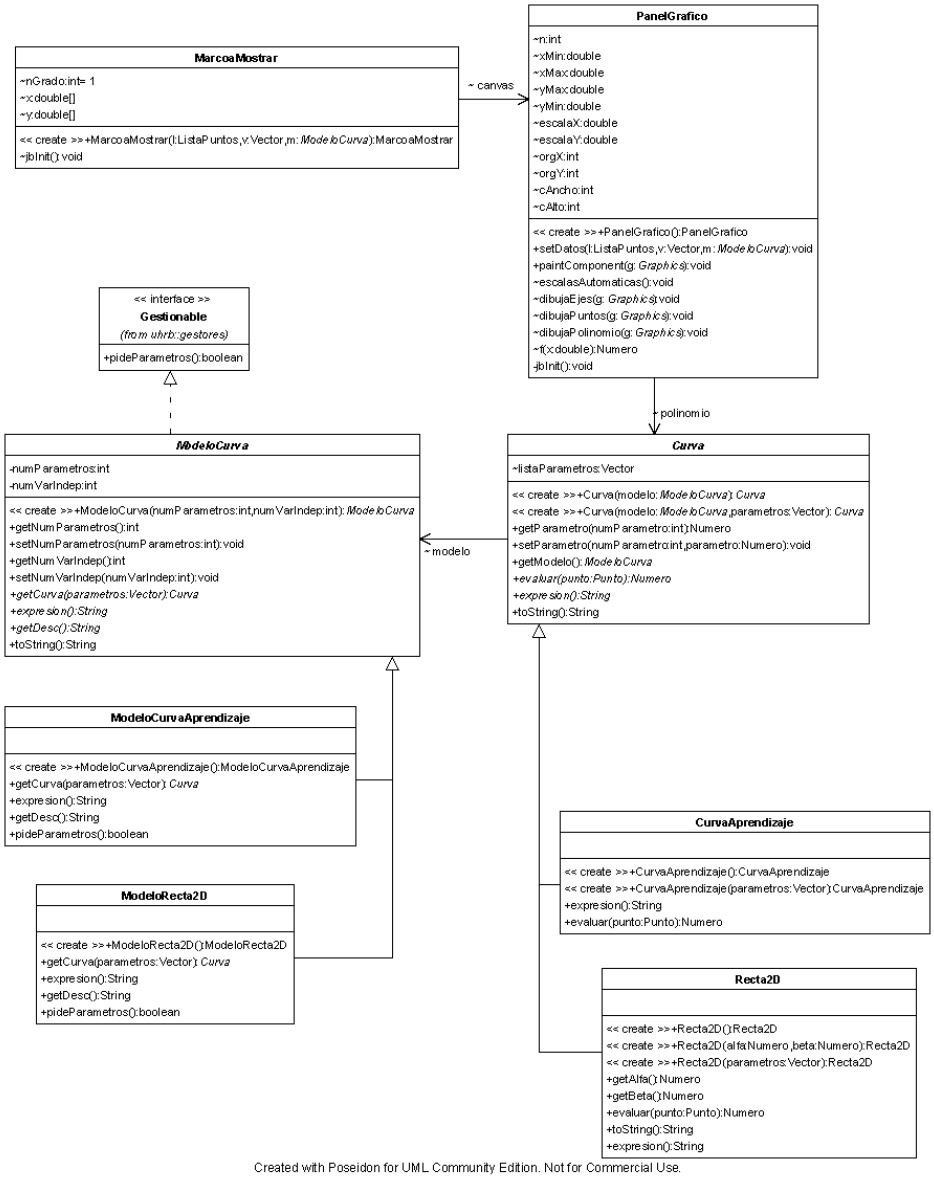
La clase **PuntoNitido** representa los puntos nítidos n-dimensionales. Esta clase está formada por una lista de valores de la clase Numero. En ella existen operaciones que permiten cargar puntos desde un fichero, cambiar el número de dimensiones del punto, obtener y modificar los valores de cada una de sus componentes y asignar un nombre descriptivo a cada una de ellas.

La clase **PuntoBorroso** es una subclase de PuntoNitido y en ella se extiende esta clase para poder representar puntos borrosos. Cada instancia de esta clase está formada por una lista de instancias de la clase NumeroBorroso y tiene un agregador asociado. Este agregador se emplea para componer el resultado de la función de pertenencia del punto borroso como agregación de las funciones de pertenencia de cada una de sus componentes.

Todos los puntos que representan las observaciones realizadas por el usuario se incluyen en una lista de puntos, implementada en la clase **ListaPuntos**, y sobre ellos se realiza la regresión. El uso de la clase abstracta Punto permite a la herramienta realizar el proceso de regresión sobre una lista de puntos heterogéneos, es decir, que en la lista se pueden mezclar puntos nítidos y borrosos.

En este paquete, además de Puntos y ListaPuntos, se han incluido las clases que representan las funciones de pertenencia de los números borrosos; todas ellas deben implementar la interfaz **FcPertenencia**, que define los métodos básicos que deben poseer todas las funciones de pertenencia.

6.2.2. Curvas



Dentro del paquete curvas se incluyen las clases en que se implementan los modelos de curva que servirán para construir las instancias de las curvas de la aplicación.

Un modelo de curva representa un modelo paramétrico matemático simbólico de un tipo de curva. Este modelo, junto con un vector de parámetros de la clase Número, es una instancia de curva, y representa una curva determinada.

Por ejemplo, el modelo para una recta será:

$$y = t_1 \cdot x_0 + t_0$$

Si a este modelo se le asigna un vector de parámetros, por ejemplo (1, 2), se obtiene la siguiente instancia de curva:

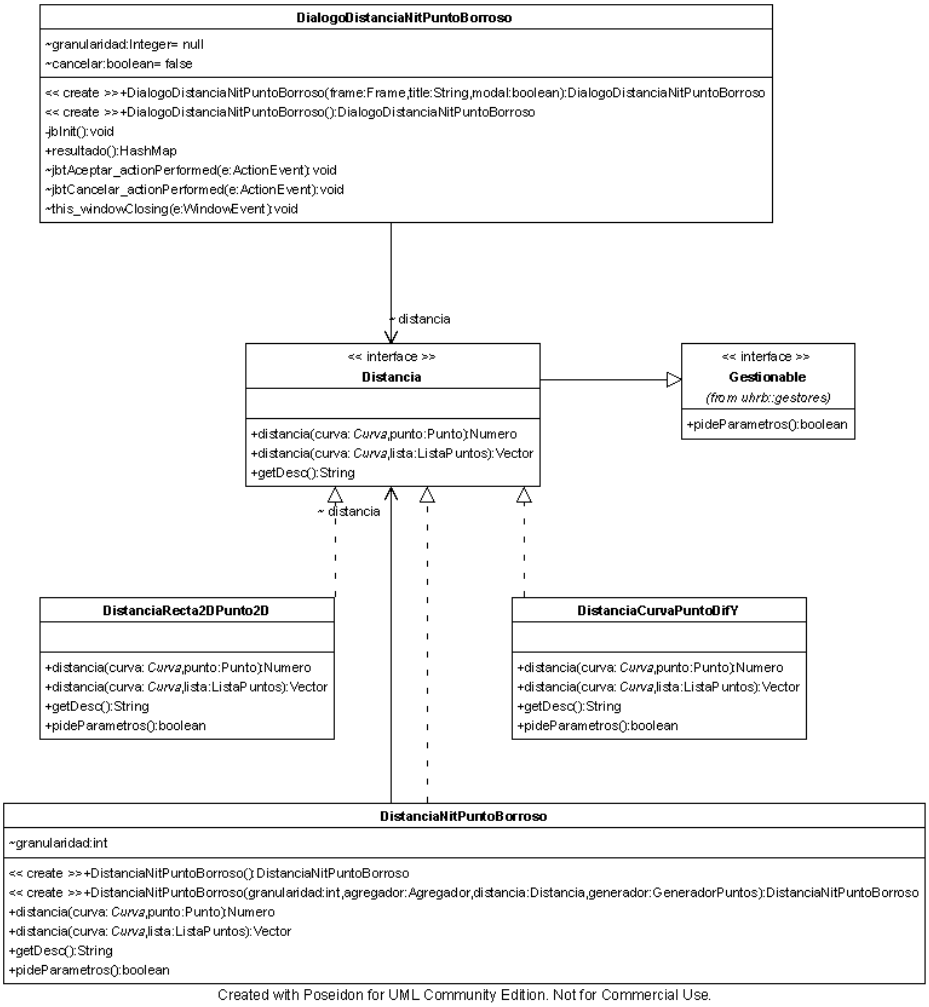
$$y = 2 \cdot x_0 + 1$$

Los parámetros asignados a un modelo de curva para obtener una instancia de curva pueden ser tanto nítidos como borrosos.

En este paquete se encuentra la clase abstracta **ModeloCurva**, que determinará el comportamiento de todos los modelos de curvas simbólicas que se pueden usar en el proceso de regresión, con el fin de conseguir la modelización más descriptiva y predictiva. Todos los modelos de curva que se implementen serán subclases de esta clase.

Las instancias de curva están representadas en el modelo de clases por la clase abstracta **Curva**.

6.2.3. Distancias



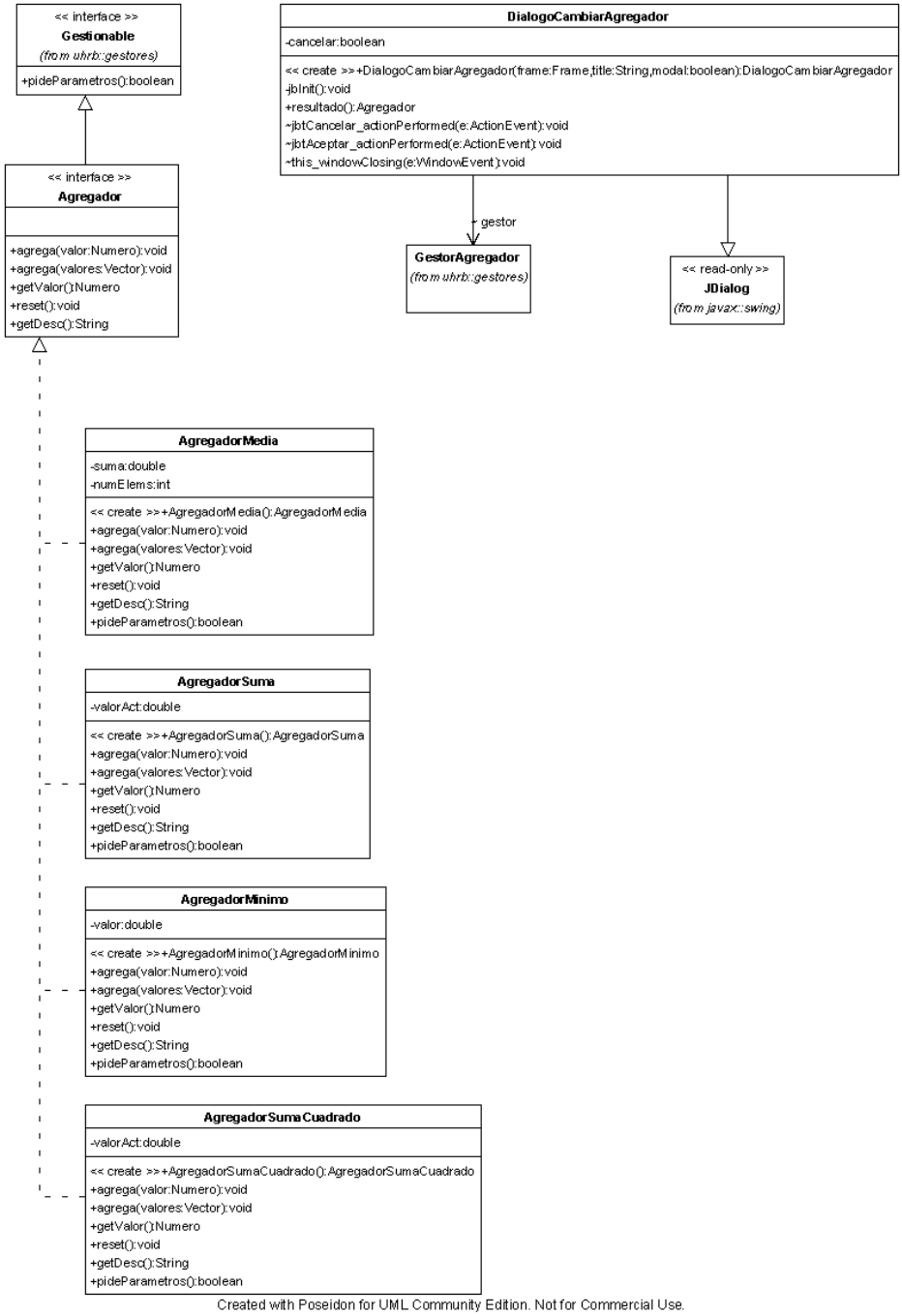
El paquete distancias se compone de distintos métodos de cálculo de la distancia existente entre una curva y un punto. Esta puede ser borrosa o nítida.

En este paquete tienen cabida todos los métodos de cálculo de la distancia, desde los más concretos, como la distancia existente entre un punto y una recta en dos dimensiones usando la fórmula analítica, hasta métodos más genéricos de cálculo de distancias de un punto n-dimensional a una curva arbitraria.

Dentro de este paquete se encuentra la interfaz **Distancia**. La implementación de esta interfaz permite utilizar diferentes tipos de métodos de cálculo de la distancia de un punto a una curva. Todos los métodos de cálculo de la distancia tienen que implementar la interfaz Distancia, contenido en este paquete. Así, se pueden implementar métodos analíticos, métodos por aproximaciones, métodos para casos concretos (distancia de un punto a una recta en dos dimensiones) o para

casos genéricos (distancia de un punto a una curva cualquiera en n dimensiones).

6.2.4. Agregadores



En este paquete se incluye la jerarquía de clases que contiene los diversos operadores de agregación que se irán implementando de entre los existentes en la literatura al efecto, así como otras clases estrechamente relacionadas con su uso, construcción o mantenimiento.

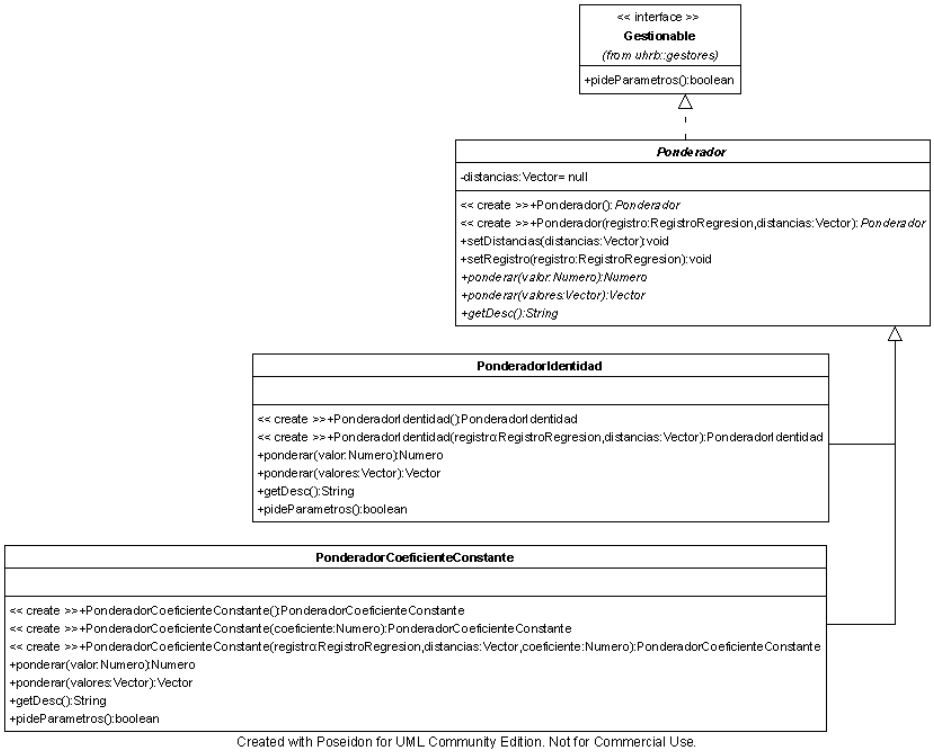
Los operadores de agregación son objetos matemáticos cuya función es reducir un conjunto de números a un único número representativo (o significativo) [4]. Ejemplos de agregadores son la media, el máximo, la suma de cuadrados, el operador OWA o la integral de Choquet.

Dentro de la herramienta se emplean para diversas operaciones, tales como calcular el resultado de la función de pertenencia de un punto borroso con respecto a sus componentes o calcular la distancia nítida de un punto borroso a una curva.

La clase principal de este paquete es la interfaz **Agregador**, que se encarga de dar un esqueleto al que se deberán ajustar todos los agregadores, con métodos para agregar valores y obtener su resultado.

La implementación de esta interfaz permite usar una jerarquía de herramientas de agregación que contengan cualquiera de los agregadores formalizados en la literatura al efecto.

6.2.5. Ponderadores

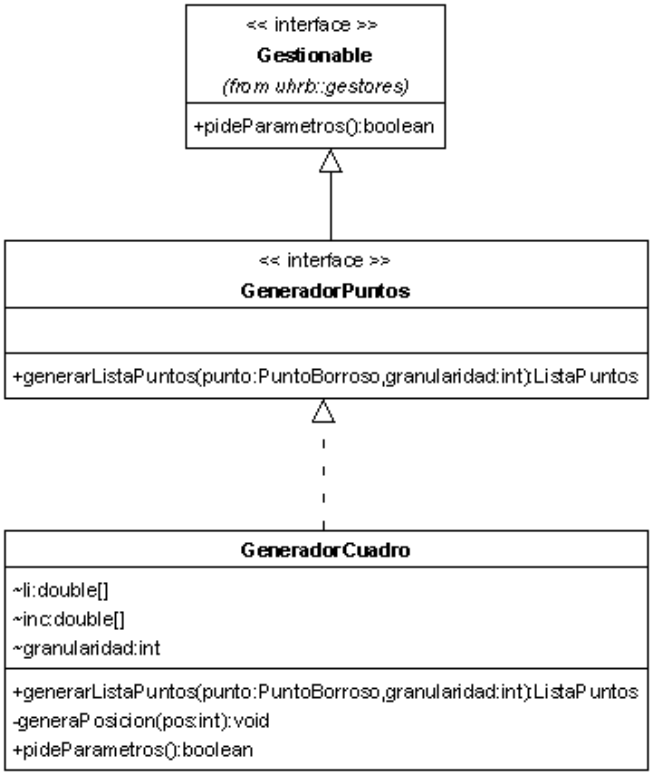


Este paquete está compuesto por clases que se utilizan para valorar los puntos nítidos y borrosos de forma que tengan pesos distintos en la operación de regresión. Este valor de ponderación podrá ser nítido o borroso.

Todos los ponderadores heredan de la clase **Ponderador**, que define los métodos principales que deben tener sus hijos.

Un ejemplo de uso de los ponderadores puede ser dar un valor de compensación a los puntos que representan observaciones atípicas o outlayers en la regresión.

6.2.6. Generadores

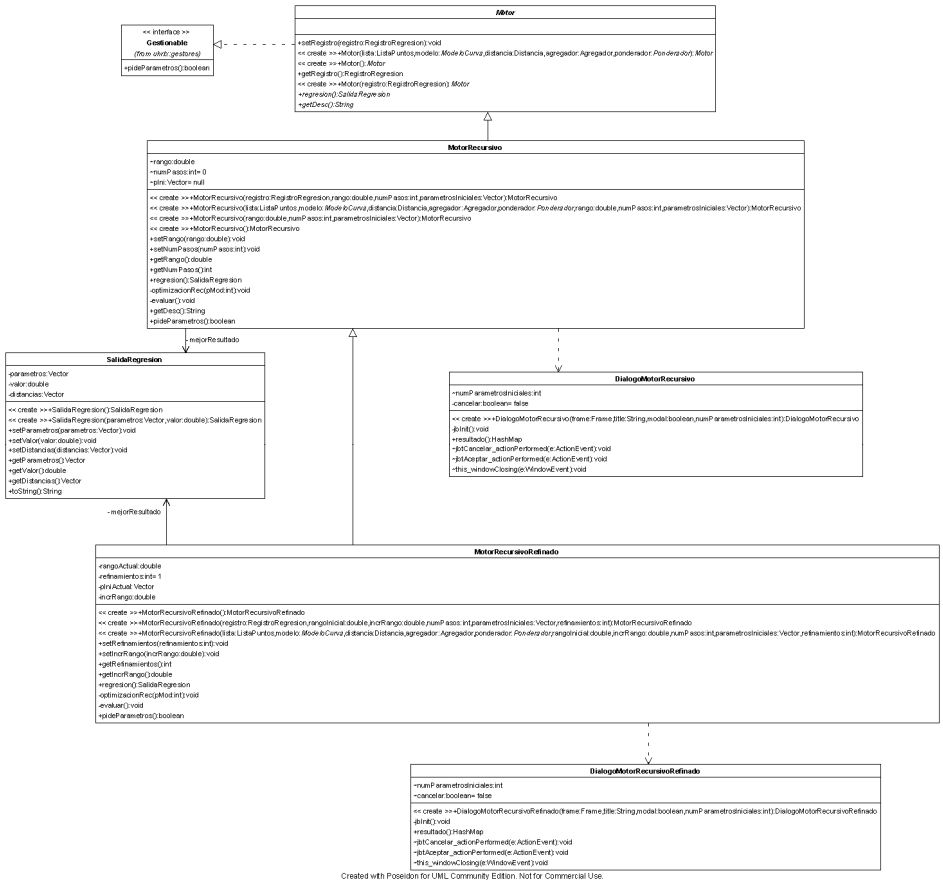


Created with Poseidon for UML Community Edition. Not for Commercial Use.

En este paquete se incluyen clases que se utilizan para generar puntos. Tendrán cabida en él aquellas clases que generen listas de puntos de acuerdo a unos criterios suministrados como parámetros de entrada.

Todos los generadores de puntos implementan la interfaz **GeneradorPuntos**.

6.2.7. Motores



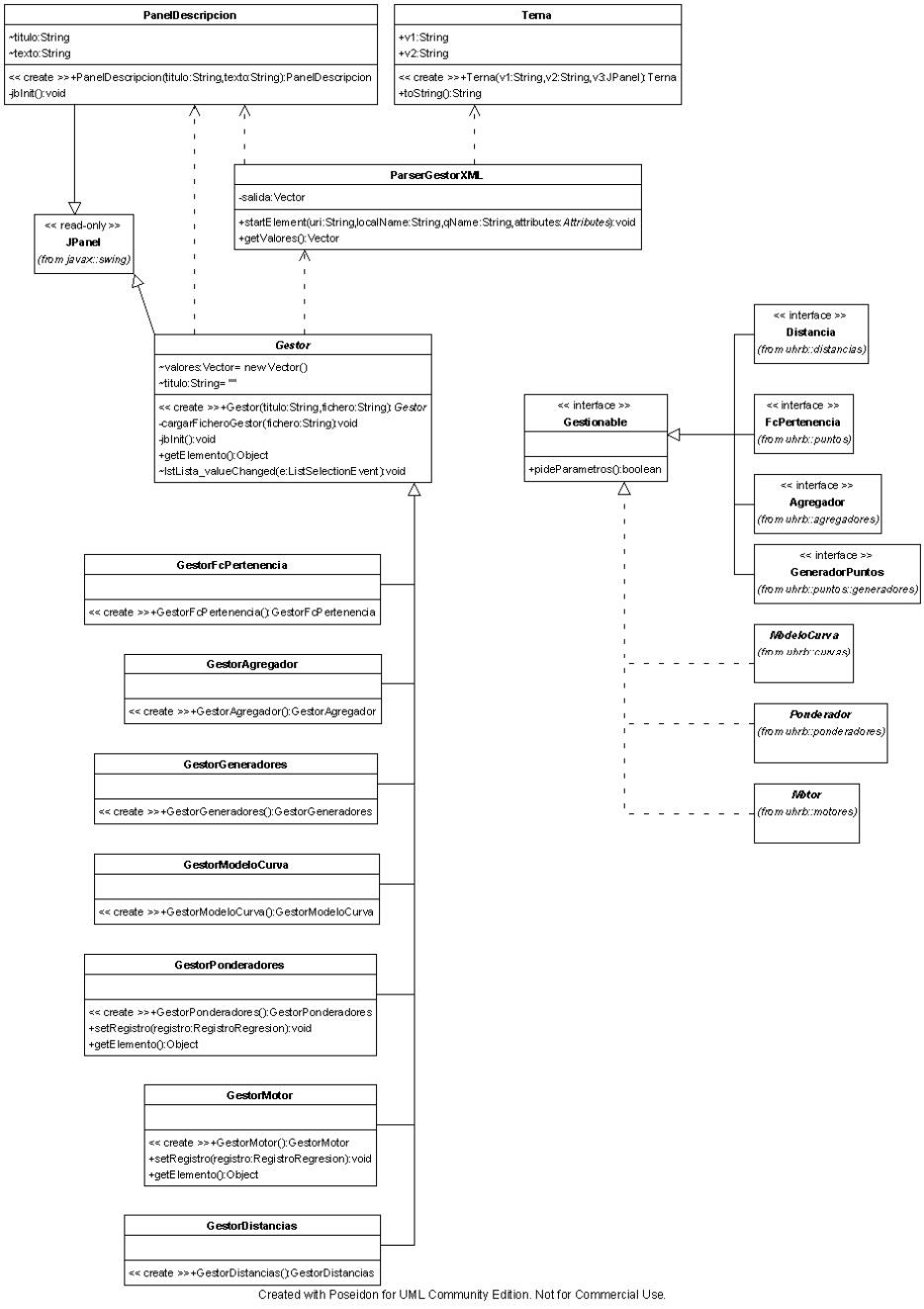
En este paquete estarán incluidos todos los motores de regresión implementados. Un motor es un objeto que, utilizando los restantes elementos propios del proceso de regresión (puntos, modelos de curva, métodos de cálculo de distancia, agregadores y ponderadores), lo dirige para obtener los parámetros que den el mejor ajuste de los puntos al modelo de curva.

La clase principal de este paquete es la clase abstracta **Motor**, de la que heredan todas las implementaciones de los distintos motores de regresión.

Las distintas implementaciones de esta clase abstracta son el núcleo de la herramienta, ya que serán las que guíen el proceso de regresión.

Se puede integrar en la aplicación toda una jerarquía de motores de regresión diferentes, cada uno de los cuales implementará un método de regresión distinto. Así, si se quiere realizar una regresión lineal se podría usar el método analítico clásico implementando un motor al efecto. De la misma manera se podrían añadir otros métodos más complejos de regresión no lineal, como Gauss-Newton, Marquardt o linear descent.

6.2.8. Gestores



Para conseguir una mejora en la explotación de la aplicación, aparte de la modularidad en el diseño, se ha realizado un conjunto de clases llamadas gestores, todas ellas incluidas en este paquete.

Un gestor es un componente gráfico que permite añadir, modificar y mostrar de forma sencilla las distintas implementaciones de cada una de las clases abstractas e interfaces que componen la aplicación. También poseen toda la información necesaria para crear, las instancias de las clases que representan los diferentes elementos que participan en el proceso de regresión.

Todos los gestores heredan de la clase abstracta **Gestor**, que a su vez es descendiente de la clase **JPanel**, por lo que todas ellas pueden representarse en una interfaz gráfica de usuario.

La mantenibilidad que permiten los gestores es uno de los puntos fuertes y reseñables de esta herramienta. Gracias a ella se ve poco afectada por los cambios que puedan producirse al añadir nuevos elementos a la aplicación.

El uso de los gestores y la forma en que están construidos permite generar motores que realicen cálculos en paralelo con distintos modelos y técnicas de regresión a la vez, usando técnicas de programación distribuida.

Por ejemplo, proporcionará a la herramienta la capacidad de realizar varias regresiones con los mismos datos de entrada usando distintos modelos de curva para obtener el modelo con el mejor ajuste para los datos.

6.3. Procedimiento de aplicación

6.3.1. Introducción

En el presente capítulo se detalla cómo se realiza el proceso de regresión en el prototipo implementado.

En los siguientes apartados se describen los procedimientos necesarios para hacer funcionar este prototipo y se da una visión general de las partes de las que se compone el mismo.

6.3.2. Formulario de entrada de datos

Mediante este formulario se introducen en la aplicación todos los datos necesarios para realizar la regresión. Estos datos son:

- Lista de puntos sobre los que se realiza la regresión.
- Modelo de curva de la que se desea obtener el ajuste de los puntos.
- Método de cálculo de las distancias de los puntos a la curva.
- Ponderador de las distancias.
- Agregador de distancias.

- Motor de regresión.
- Borrosidad para los puntos que se considere oportuno

Como se puede ver en la figura 4, este formulario está dividido en dos partes.

En la parte superior se encuentra un panel que contiene la lista de puntos y las propiedades de los mismos (valor de cada componente y propiedades de borrosidad). En la parte inferior hay una serie de pestañas en las que se pueden seleccionar los restantes elementos de la regresión, es decir, el modelo de curva, el método de cálculo de las distancias, el ponderador de distancias, el agregador de distancias y el motor de regresión.

Cada una de las pestañas contiene un Gestor del tipo de elemento mediante el cual se realiza la selección y la creación del elemento. Como se explicó en el capítulo 5.2.8.

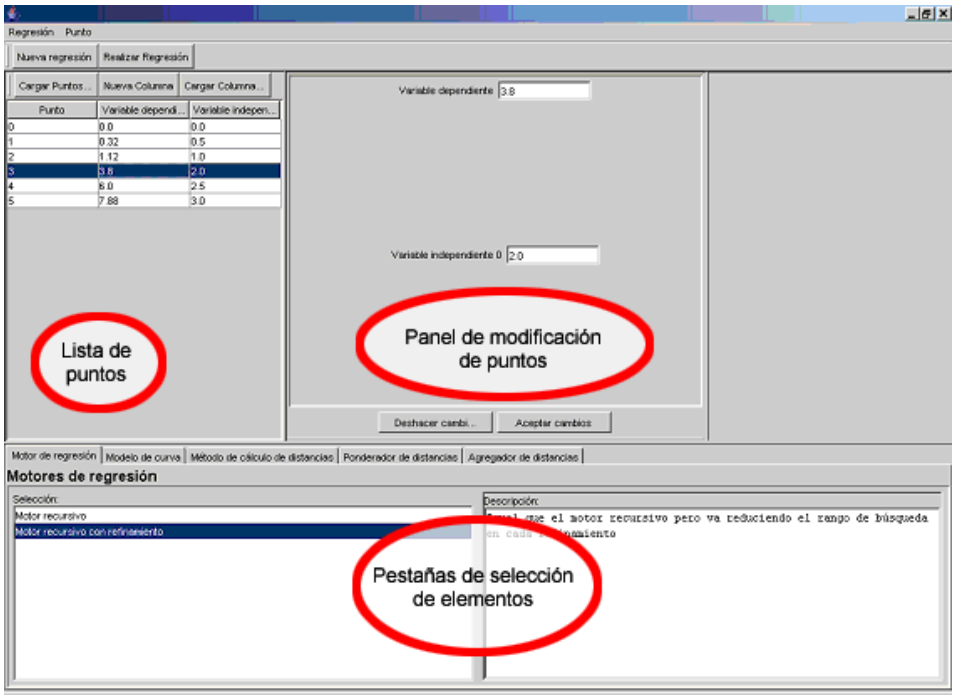


Figura 4. Formulario de entrada de datos

Los pasos para realizar la regresión son los siguientes.

1. Cargar una lista de puntos.

El paso inicial debe ser cargar desde un fichero la lista de puntos para realizar la regresión.

Para cargar los puntos se puede usar el botón 'Cargar puntos' que se encuentra situado sobre la *lista de puntos*. También existe la posibilidad de acceder a esta opción a través del menú *Punto*, en la opción *Cargar puntos*. En ambos casos, esto hará que se muestre una ventana en la que se puede seleccionar el fichero que contiene los puntos.

La gramática que define el formato del fichero que contiene los puntos es la siguiente:

f → **entero** '<eol>' lista | lista

lista → punto '<eol>' lista | punto

Es decir, que un fichero que define una lista de puntos está compuesto por una primera línea opcional en la que aparece un entero. Este entero indica el **tamaño de los puntos** (el número de componentes que tiene cada uno de los puntos que aparecen en el fichero). A continuación debe aparecer una línea por cada uno de los puntos de la lista. El formato de cada punto es:

punto → espacio '[' **real** vars_indep ']'

espacio → blanco espacio | ε

blanco → ' ' | '<tab>'

vars_indep → ',' **real** | ε

donde ε es la cadena vacía y '<eol>' es el carácter de fin de línea. Según esto, cada punto puede estar precedido por un número indeterminado de espacios o tabuladores. A continuación de ellos debe encontrarse una apertura de corchete ('[') y después un número que se interpretará como la **variable dependiente** del punto. Después de la variable dependiente se pueden encontrar cero o más valores numéricos, separados entre sí por comas (sin espacios) que serán interpretados como las **variables independientes**. Para indicar el final del punto se usa el símbolo de cierre de corchete (']').

En posteriores versiones se implementará una gramática más flexible para el fichero de puntos y otros métodos alternativos

para la introducción de los mismos, como puede ser la introducción manual.

Una vez cargados los puntos, estos aparecerán en la **lista de puntos**, que se encuentra en la parte superior izquierda del formulario de entrada de datos.

2. Modificar los puntos cargados.

   Una vez se han cargado los puntos puede modificarse el valor de cualquiera de sus componentes. Para ello ha de seleccionarse el punto en la *lista de puntos*. Las componentes del punto seleccionado aparecerán en el *panel de modificación de puntos*, pudiendo ser cambiado su valor por el usuario.

   Una vez realizadas las modificaciones debe pulsarse el botón 'Aceptar cambios' si se desea que estas tengan efecto, o 'Deshacer cambios' si se desea que se anulen.

   En este panel también se pueden modificar las propiedades de borrosidad de cada una de las componentes de los puntos, como se indica en el paso siguiente.

3. Asignar borrosidad a los puntos.

   Cuando se ha cargado una lista de puntos se puede asignar borrosidad a los mismos.

   Para ello se procede de la siguiente manera: primero se selecciona el punto a borrosificar en la *lista de puntos*. A continuación se selecciona el menú *Punto* y la opción *Asignar borrosidad*.

   Para asignar la borrosidad hay que seleccionar dos elementos:

   3.1. Un agregador para calcular la función de pertenencia del punto.

   3.2. Una función de pertenencia que, inicialmente, se asignará a todas y cada una de las componentes del punto.

   Dependiendo de la función de pertenencia seleccionada es posible que la aplicación solicite la introducción de nuevos parámetros, si estos son necesarios para su construcción. Estos parámetros se explicarán más adelante en la sección correspondiente. Lo mismo puede suceder en el caso del agregador.

Una vez que se ha introducido el agregador y la función de pertenencia para el punto, estos aparecerán en el *panel de modificación de puntos*. Ahora, en este panel, se puede modificar el agregador de la función de pertenencia de los puntos (presionando el botón 'Cambiar' que se encuentra al lado del nombre del agregador) y la función de pertenencia de cualquiera de las componentes. Nótese que de esta manera las funciones de pertenencia pueden ser diferentes para cada una de las componentes del punto.

Esta previsto que en posteriores versiones de la aplicación se implementen métodos para cargar la borrosidad asociada a los puntos desde un fichero, de la misma forma que se cargan estos puntos.

4. Seleccionar un modelo de curva, un método de cálculo de la distancia, un ponderador de distancias, un agregador de distancias y un motor de regresión.

   Para seleccionar un elemento para realizar la regresión hay que presionar en la pestaña correspondiente y seleccionarlo en la lista que aparece debajo de esta. En el panel de la derecha se mostrará una descripción del elemento seleccionado. El elemento resaltado es el que se utilizará para realizar la regresión.

5. Realizar la regresión.

   Una vez realizados todos los pasos anteriores, para realizar la regresión se debe pulsar el botón 'Realizar regresión' o seleccionar en el menú *Regresión* la opción *Realizar regresión*.

   En este momento, la aplicación intentará crear todos los elementos participantes en la regresión, pidiendo al usuario los parámetros necesarios como se detalla en las secciones subsiguientes.

6.3.3. Elementos propios del proceso de regresión implementados hasta el momento

A continuación se enumeran los elementos que se han implementado y, para cada uno de ellos, los parámetros que debe introducir el usuario.

1. Modelos de curva

   1.1. Recta en dos dimensiones

   Es una recta en dos dimensiones de la forma

$$y = t_1 \cdot x_0 + t_0$$

1.2. Curva de aprendizaje

Es una curva de la forma

$$y = t_1 \cdot x_0^{t_0}$$

2. Métodos de cálculo de la distancia

2.1. Recta 2D – Punto 2D

Obtiene la distancia euclídea entre cada punto y la curva. Es el resultado de aplicar la siguiente fórmula:

$$d = \frac{\sqrt{\left(y_i \cdot t_1 - x_i \cdot t_1^2 - t_1 \cdot t_0\right)^2 + \left(x_i \cdot t_1 + t_0 - y_i\right)^2}}{1 + t_1^2}$$

donde la recta tiene la forma

$$y = t_1 \cdot x_0 + t_0$$

y para el punto, y_i es la variable dependiente y x_i la variable independiente.

Este método sólo puede aplicarse a puntos en dos dimensiones y rectas en dos dimensiones.

2.2. Distancia en Y

Se obtiene la distancia de la misma manera que en el método de mínimos cuadrados, aplicando la siguiente fórmula:

$$d = \left| y_i - f\left(\vec{x}_i, \vec{t}\right) \right|$$

2.3. Distancia nítida de puntos borrosos

Este método permite calcular la distancia entre puntos borrosos y cualquier modelo de curva. Este cálculo se basa en la **agregación** de las distancias de un cierto número de puntos nítidos, generados utilizando un **generador de puntos**, ponderadas por su grado de pertenencia al punto borroso.

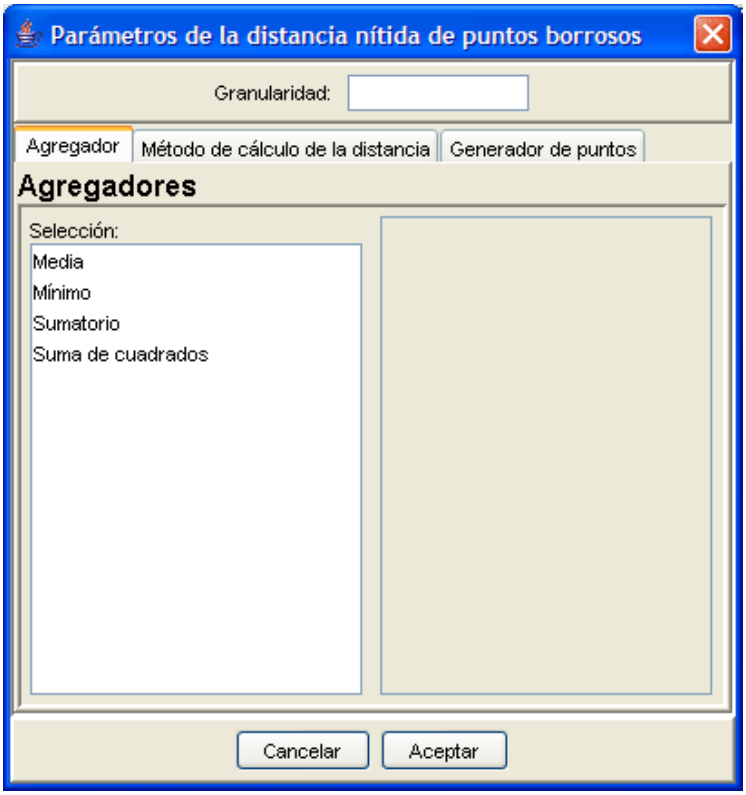


Figura 5. Parámetros de la distancia nítida de puntos borrosos

Al realizar una regresión utilizando este método de cálculo de la distancia se muestra al usuario el formulario de la figura para la entrada de los parámetros. Los parámetros solicitados son:

2.3.1. Agregador

Un agregador de entre los que se describirán en el apartado 4.

2.3.2. Método de cálculo de la distancia

Un método de cálculo de la distancia. Puede utilizarse cualquiera de los mencionados anteriormente.

2.3.3. Granularidad

Es un valor proporcional al número de puntos que generará el generador de puntos.

2.3.4. Generador de puntos

El generador de puntos se encargará de obtener una lista de puntos en torno a cada uno de los puntos borrosos de los que se calcula la distancia.

En la versión actual, el único generador de puntos implementado es el llamado *Generador por cuadratura* que describiremos a continuación.

2.3.4.1. Generador por cuadratura

Para cada componente borrosa del punto se obtiene un conjunto de valores nítidos comprendidos dentro del rango del valor de la componente ± la amplitud de la función de pertenencia.

La lista de puntos generada se formará con las permutaciones formadas cogiendo un elemento de cada uno de los conjuntos y colocándolo en la posición de la componente que lo originó.

Por ejemplo, para el punto (3, 5), con una amplitud de 2 para la función de pertenencia de la primera componente y de 1 para la segunda, y con una granularidad de 5, los puntos generados son:

(1, 4)	(2, 4)	(3, 4)	(4, 4)	(5, 4)
(1, 4.5)	(2, 4.5)	(3, 4.5)	(4, 4.5)	(5, 4.5)
(1, 5)	(2, 5)	**(3, 5)**	(4, 5)	(5, 5)
(1, 5.5)	(2, 5.5)	(3, 5.5)	(4, 5.5)	(5, 5.5)
(1, 6)	(2, 6)	(3, 6)	(4, 6)	(5, 6)

Gráficamente:

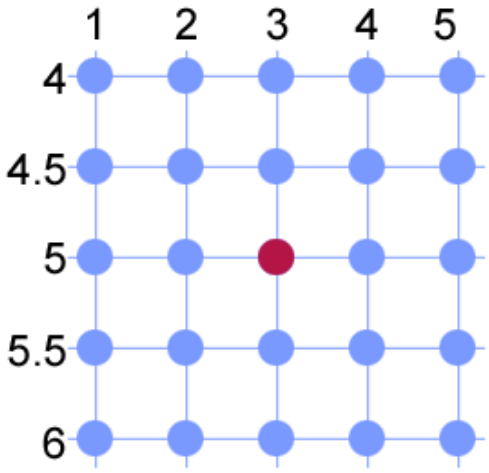


Figura 6. Puntos generados por un generador por cuadratura

Para obtener la distancia nítida de un punto borroso (p_β) a una curva se procede de la siguiente manera:

- Se obtiene una lista de puntos del *Generador de puntos*
- Para cada uno de los puntos generados se obtiene
  - la distancia a la curva
  - el grado de pertenencia a p_β
- Se multiplican estos valores y se agregan usando el agregador indicado por el usuario. El valor obtenido de esta agregación será el utilizado como medida de la distancia.

3. Ponderadores

3.1. Identidad

La ponderación realizada es una multiplicación por 1. Este ponderador es útil si no se desea aplicar ninguna ponderación.

3.2. Producto por un coeficiente constante

Multiplica la distancia de cada punto a la curva por un coeficiente introducido por el usuario.

4. Agregadores

4.1. Media

Calcula la media de todos los valores que recibe:

$$a = \frac{\sum_{i=1}^{n} d_i}{n}$$

4.2. Mínimo

Devuelve el mínimo de todos los valores recibidos:

$$a = \min_{i=1}^{n}(d_i)$$

4.3. Sumatorio

Obtiene la suma de los valores recibidos:

$$a = \sum_{i=1}^{n} d_i$$

4.4. Suma de cuadrados

Obtiene la suma de los valores recibidos, elevándolos previamente al cuadrado:

$$a = \sum_{i=1}^{n} \left(d_i^2\right)$$

5. Motores de regresión

5.1. Motor recursivo

Este motor hace variar cada parámetro de la curva dentro de un intervalo determinado, que introduce el usuario, y aplica cada una de las combinaciones de parámetros posibles al

modelo de curva, para obtener, como se ha visto en capítulos anteriores, la agregación de las distancias ponderadas. El valor que devuelve como solución será el que minimice la agregación de las distancias.

En la figura 7 se muestra la pantalla de entrada de parámetros para este motor:

Figura 7. Parámetros del motor recursivo

El significado de los parámetros es el siguiente:

- Rango: máxima distancia que se alejará cada parámetro de su valor inicial.
- Número de pasos: número de valores intermedios dentro del rango que se generarán para los parámetros.
- Parámetro i: valor inicial para cada uno de los parámetros.

Así pues, cada parámetro tomará tantos valores como indique el número de pasos, y estos estarán comprendidos entre su valor inicial – rango y su valor inicial + rango.

Los valores de los parámetros del motor recursivo, principalmente los valores iniciales de los parámetros, serán determinantes para obtener un resultado mejor o peor.

5.1. Motor recursivo con refinamiento

El funcionamiento de este motor es similar al del anterior, aunque puede alcanzar un mejor ajuste partiendo de los mismos parámetros. Como contrapartida necesitará, por regla general, mayor tiempo de ejecución.

El motor realiza inicialmente la búsqueda de los parámetros óptimos para el modelo de curva de la misma manera que en el caso anterior. Una vez obtenidos estos parámetros óptimos, aplica de nuevo el mismo algoritmo de búsqueda pero partiendo en este caso de los parámetros óptimos obtenidos, y

reduciendo el rango de búsqueda. Al reducir el rango de búsqueda pero no el número de pasos se está buscando en valores más próximos entre sí. Esto se repite iteративamente un cierto número de veces indicado por el usuario.

Figura 8. Parámetros del motor recursivo con refinamiento

Los parámetros necesarios para hacer funcionar este motor son:

- Rango, número de pasos y parámetro i: estos parámetros tienen el mismo significado que en el caso anterior.
- Coeficiente del rango: coeficiente por el que se multiplicará el rango después de cada iteración. Para que el rango disminuya en cada iteración, este coeficiente debe ser menor que 1.
- Número de iteraciones: número de veces que se aplicará el algoritmo.

6. Funciones de pertenencia

6.1. Triangular

Según lo expuesto en el apartado 5.1.6 esta función de pertenencia está definida por la siguiente expresión:

$$\mu(x)=\begin{cases}0 & \text{si } x \le a \text{ o } x \ge b \\ \dfrac{x-a}{m-a} & \text{si } x \in (a,m] \\ \dfrac{b-x}{b-m} & \text{si } x \in (m,b)\end{cases}$$

siendo a y b los límites superior e inferior y m un valor intermedio en el que la función alcanza su máximo valor.

Para la implementación se ha considerado que los valores a y b están a la misma distancia de m, por lo que la ecuación resultante se simplificará, resultando la siguiente:

$$\mu(x) = 1 - \frac{|x - m|}{a}$$

donde a no es el límite sino la distancia existente entre el centro y los límites, como se puede ver en la figura siguiente.

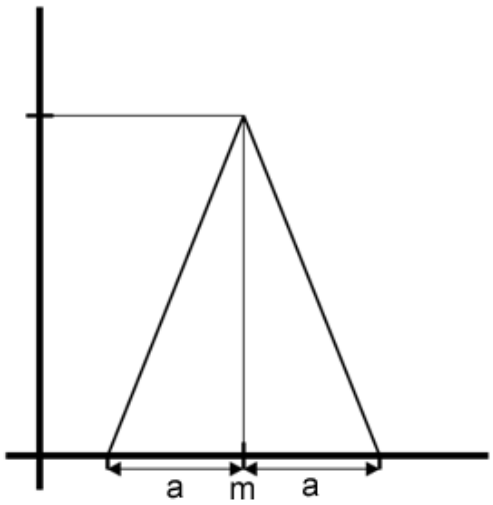


Figura 9. Función de pertenencia triangular

6.3.4. Formulario de resultados

Como se muestra en la figura 10, en este formulario se pueden ver los resultados del proceso de regresión realizado, así como cada uno de los elementos participantes en este proceso.

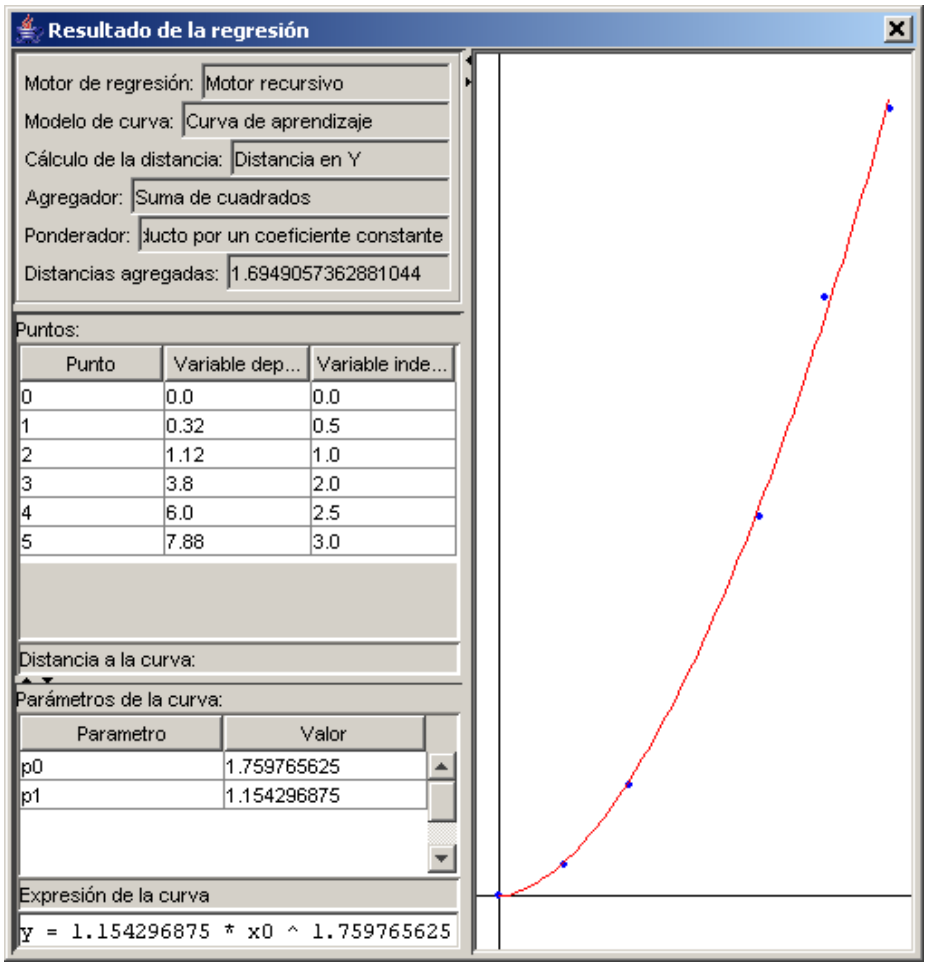


Figura 10. Formulario de resultados

Este formulario se divide en dos partes:

En la parte derecha se muestra un panel que contiene una gráfica de la curva obtenida durante el proceso de regresión y los puntos introducidos por el usuario. La implementación del presente prototipo permite representar cualquier modelo de curva en dos dimensiones y puntos nítidos. En el caso de los puntos borrosos se representarán sin tener en cuenta su borrosidad.

En la parte izquierda se recogen todos los datos numéricos del proceso de regresión, tanto los introducidos por el usuario como los calculados por la aplicación. Estos datos son:

- Sección 1. Parámetros de entrada:

  En la parte superior se muestran los parámetros introducidos a la aplicación para el cálculo de la regresión.

  También se muestra el valor obtenido de agregar todas las distancias ponderadas de los puntos a la curva con los parámetros obtenidos como solución.

- Sección 2. Puntos:

  En esta sección se recogen en una lista todos los puntos introducidos por el usuario. Al seleccionar alguno de los puntos en la lista, se muestra en la parte inferior la distancia de este punto a la curva obtenida en la regresión (es decir, a la curva representada en el panel de la derecha)

- Sección 3. Parámetros de la curva:

  En el panel inferior se muestran en una tabla los mejores valores que ha obtenido el motor de regresión para cada uno de los parámetros del modelo de curva. Debajo se muestra la expresión de la curva sustituyendo cada uno de los parámetros por su valor correspondiente.

6.3.5. Procedimiento para añadir nuevos elementos propios del proceso de regresión

Una de los principales objetivos que se ha tenido en mente en todo momento durante el diseño y desarrollo de esta aplicación ha sido el permitir añadir elementos del proceso de regresión (motores de regresión, métodos de cálculo de la distancia, agregadores...) una vez que la aplicación estuviera terminada, sin que esto supusiera un gran cambio en el diseño o en el código fuente de la misma.

Se puede afirmar que el éxito en este apartado ha sido completo ya que para añadir nuevos elementos a la aplicación no es necesario modificar en absoluto su código fuente, y mucho menos su diseño. Tan solo hay que realizar cambios en un fichero de configuración, como se indica a continuación.

Pasos a seguir:

1. Colocar el fichero compilado (.class) del elemento que se va a añadir en una carpeta que se encuentre en el **classpath**.

   Es conveniente utilizar las carpetas creadas a tal efecto para los elementos del prototipo, que se encuentran en el directorio de instalación de la herramienta, dentro de la carpeta uhrb. En ese caso, es importante reseñar que las clases añadidas deben estar contenidas en el paquete uhrb.<nombre de la carpeta del tipo de elemento>.

   Por ejemplo:

   Supongamos que se desea añadir el modelo de curva *Curva de aprendizaje* a la aplicación. El primer paso es situar el fichero ModeloCurvaAprendizaje.class, es decir, el fichero compilado, en una carpeta del classpath, en este caso, en la carpeta uhrb/curvas. Como se puede observar en el código fuente, el paquete correspondiente a esta clase, dada su ubicación, es uhrb.curvas.

2. Modificar el fichero XML del gestor correspondiente al nuevo elemento añadido.

   Es necesario añadir una nueva línea a este fichero para que le gestor sea consciente de que existe un nuevo elemento y sepa dónde se encuentra el fichero compilado para así poder crear las clases.

   Los ficheros XML del prototipo se encuentran en la carpeta classes/xml.

   La DTD (Document Type Definition) que describe el formato de estos ficheros es la siguiente:

```
<?xml version="1.0" encoding="UTF16" ?>

<!ELEMENT elemento EMPTY >

<!ATTLIST elemento clase NMTOKEN #REQUIRED >

<!ATTLIST elemento descCorta CDATA #REQUIRED >

<!ATTLIST elemento descLarga CDATA #REQUIRED >

<!ELEMENT elementos ( elemento+ ) >
```

   Es decir, que el formato del fichero será:

   En primer lugar una línea opcional en la que se indican la versión de XML utilizada y otras características que debe tener en cuenta el parser. Una de las características más importantes es la codificación del fichero XML (el atributo 'encoding').

   Es vital asegurarse de que el editor que se utiliza para modificar el fichero XML utiliza la codificación indicada en el mismo. De no ser así, esto podría provocar fallos graves en la aplicación, impidiendo la ejecución de la misma.

   A continuación se abre un elemento del tipo '**elementos**'. Este elemento puede contener en su interior un número arbitrario de elementos del tipo '**elemento**', uno por cada una de las componentes del gestor. Cada '**elemento**' está formado necesariamente por tres atributos:

   - clase: es el nombre de la clase totalmente calificado para poder encontrarlo en el classpath.

- descCorta: un texto descriptivo corto. Este texto se mostrará en la lista del gestor como nombre del elemento.

- descLarga: un texto descriptivo más largo en el que se explique la función del elemento. Este texto se mostrará en un panel de descripción junto al elemento cuando este se seleccione.

Por último, se cierra el elemento '**elementos**'.

Siguiendo con el ejemplo del paso anterior, el fichero a modificar sería classes/xml/modelos.xml.

Antes de añadir el nuevo modelo de curva su apariencia es la siguiente:

```
<?xml version='1.0' encoding='UTF-16'?>

<elementos>

  <elemento

    clase="uhrb.curvas.ModeloRecta2D"

    descCorta="Recta en 2 dimensiones"

    descLarga="Recta en 2 dimensiones. y = ax + b"/>

</elementos>
```

Como se ha dicho en el paso anterior, la clase ModeloCurvaAprendizaje se encuentra en el paquete uhrb.curvas, por lo que el nombre de la clase completamente calificado será uhrb.curvas.ModeloCurvaAprendizaje. Para los restantes parámetros se introduce una cadena de texto suficientemente descriptiva, resultando el fichero XML final como sigue:

```
<?xml version='1.0' encoding='UTF-16'?>

<elementos>

  <elemento

    clase="uhrb.curvas.ModeloRecta2D"

    descCorta="Recta en 2 dimensiones"

    descLarga="Recta en 2 dimensiones. y = ax + b"/>

  <elemento

    clase="uhrb.curvas.ModeloCurvaAprendizaje"

    descCorta="Curva de aprendizaje"

    descLarga="Curva de aprendizaje. y = p1 * x^p0"/>

</elementos>
```

6.3.6. Notas sobre la creación de nuevos elementos

Para construir nuevos elementos que se puedan incorporar a la aplicación hay que tener en cuenta una serie de restricciones que deben cumplir y que se enumeran a continuación:

1. Para que el nuevo elemento pueda ser utilizado por otras clases durante el proceso de regresión, todo nuevo elemento de un tipo debe **implementar la interfaz o clase abstracta** existente para ese tipo. Por ejemplo, si se desea implementar un nuevo método de cálculo de la distancia este debe implementar la interfaz Distancia. Esto asegura que todos los elementos de un determinado tipo (por ejemplo, todos los métodos de cálculo de la distancia) poseen ciertos métodos necesarios para que sean utilizados por las demás clases.

2. Para que el nuevo elemento pueda ser mostrado y manipulado por su Gestor, debe **implementar la interfaz Gestionable**. Este apartado está incluido intrínsecamente en el anterior, ya que todas las interfaces y clases abstractas de estos elementos implementan o extienden la interfaz Gestionable.

   También es necesario, por cómo se han construido los gestores, que exista en la clase un **constructor sin parámetros**.

Lo reseñado en este apartado describe el procedimiento de manejo del prototipo de aplicación implementado, con la intención de que los

futuros cambios que lleguen a hacerse continúen con la misma línea de trabajo.

6.4. Ensayos preliminares

Se han realizado diversos ensayos para comprobar el buen comportamiento de la aplicación en la práctica.

Se reproducen a continuación los resultados obtenidos en uno de ellos.

Datos de partida

Los datos de partida se han generado a partir de una ecuación ($y = 5 \cdot x^3$), introduciendo ruido en los valores para romper su uniformidad. El ruido se introdujo en las *x*, siguiendo una distribución normal de media 0 y desviación estándar 1.

En la siguiente tabla se muestran los valores generados:

```
0.  [10968.146565,12.290067]
1.  [12651.729034,14.190259]
2.  [0.017642,0.101942]
3.  [11196.715460,14.246579]
4.  [33781.549956,19.563511]
5.  [9226.082893,10.715332]
6.  [19196.649069,12.629441]
7.  [0.001254,0.603648]
8.  [20247.250095,14.930159]
9.  [10575.305973,13.744380]
10. [227.414733,5.151876]
11. [5934.902628,9.608926]
12. [418.663929,5.382772]
13. [6584.553549,11.119540]
14. [7.900691,0.577820]
15. [8114.779801,13.325824]
16. [2881.802944,7.805422]
17. [259.247074,4.956705]
18. [10.441939,2.862090]
19. [16.728799,-0.593295]
20. [1192.072016,9.150294]
21. [33658.376981,20.237823]
22. [37731.484912,20.664601]
23. [6842.424773,10.335140]
24. [38638.815269,19.512860]
25. [13229.681985,12.459353]
26. [564.519511,3.565562]
27. [21242.974196,15.301325]
28. [32644.814692,19.279323]
29. [85.475735,4.418699]
30. [12959.904598,15.084505]
31. [1050.517655,5.453646]
32. [10844.697302,10.766849]
33. [3989.665674,9.512176]
34. [31431.718387,17.720382]
35. [564.947431,3.055019]
36. [11508.023227,13.651161]
37. [1467.573833,7.226954]
```

38. [4334.512658,10.391667]
39. [4120.486790,9.109202]
40. [14071.582301,13.701109]
41. [552.610100,4.593181]
42. [14756.574212,14.169729]
43. [25906.250257,17.521511]
44. [2776.851568,9.904088]
45. [3066.014266,8.615272]
46. [34761.312031,19.736395]
47. [27390.921548,19.708433]
48. [13637.122210,13.632528]
49. [1139.388824,6.838136]
50. [22815.814689,16.880491]
51. [36575.223867,18.562983]
52. [1081.109722,3.468196]
53. [39767.919727,17.582303]
54. [3424.777088,8.468693]
55. [0.009658,-0.486399]
56. [994.872178,5.429831]
57. [13247.973936,12.421959]
58. [4788.531357,10.084108]
59. [23.268607,1.877527]
60. [300.201249,3.196025]
61. [37371.159338,20.309657]
62. [1964.311843,5.680839]
63. [108.430371,1.731835]
64. [0.128506,0.043449]
65. [10517.307124,11.514383]
66. [16059.728218,15.987695]
67. [0.766279,2.029771]
68. [42.639118,2.278990]
69. [28516.490182,16.462245]
70. [17829.099326,15.936321]
71. [7912.537011,9.096677]
72. [12881.007692,13.173660]
73. [36184.226266,22.545195]
74. [6667.509483,11.446086]
75. [18778.162095,14.394036]
76. [9094.896004,13.093773]
77. [38646.759241,19.488481]
78. [4.496578,2.000581]
79. [38272.182558,19.342904]
80. [343.276213,5.436711]
81. [30389.673223,19.258415]
82. [11792.184819,13.524795]
83. [3952.683141,8.947164]
84. [4.516459,1.222514]
85. [3904.602070,9.017958]
86. [20476.204393,15.919951]
87. [969.850206,6.488514]
88. [13434.029701,13.105507]
89. [697.206977,4.384295]
90. [14512.702580,14.256996]
91. [14952.929222,13.680583]
92. [15771.159994,13.174632]
93. [9641.038403,13.316950]
94. [38786.020495,19.529913]
95. [141.522469,1.480861]
96. [336.214812,3.671815]
97. [21999.621895,16.242477]
98. [7.981550,-1.165540]
99. [6246.453734,9.412596]

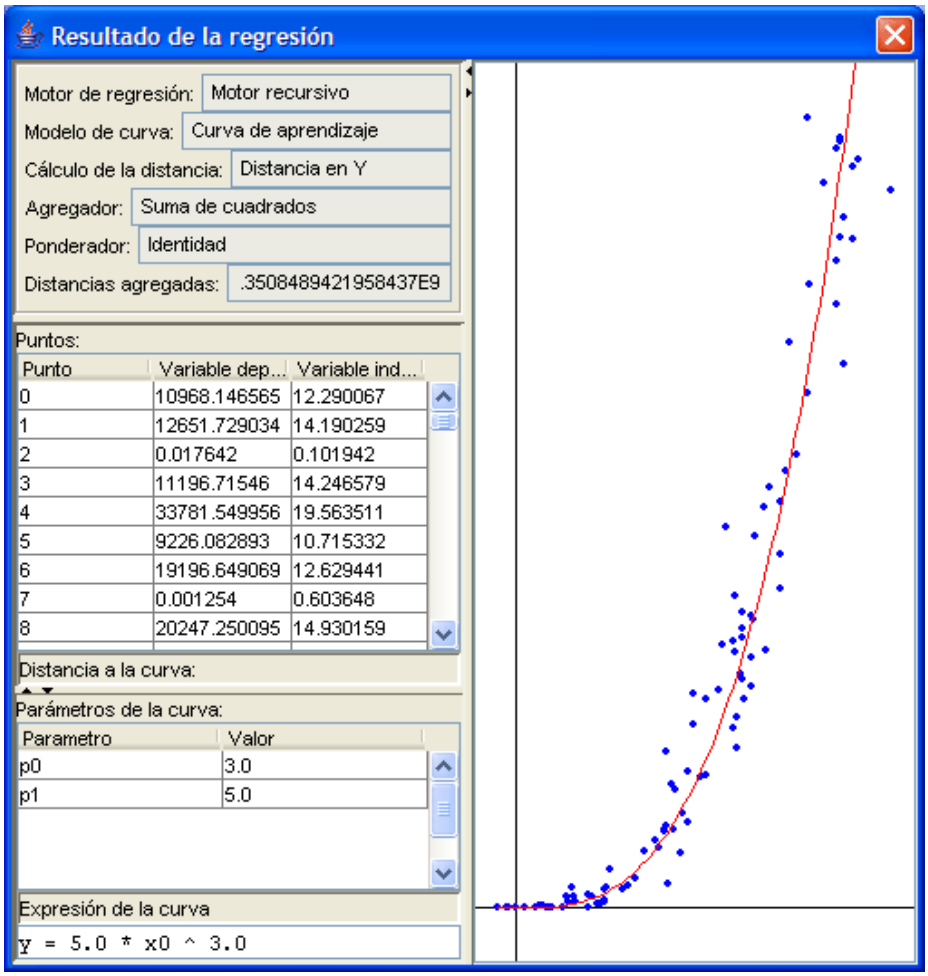
Parámetros y resultados

A continuación se muestran los resultados de los ensayos con los datos de partida anteriormente expuestos y variando los parámetros de la regresión.

Parámetros:

- Motor de regresión: motor recursivo
    - Rango: 10
    - Número de pasos: 10
    - Parámetro 0: 0
    - Parámetro 1: 0
- Modelo de curva: curva de aprendizaje
- Método de cálculo de la distancia: distancia en Y
- Ponderador de distancias: identidad
- Agregador de distancias: suma de cuadrados

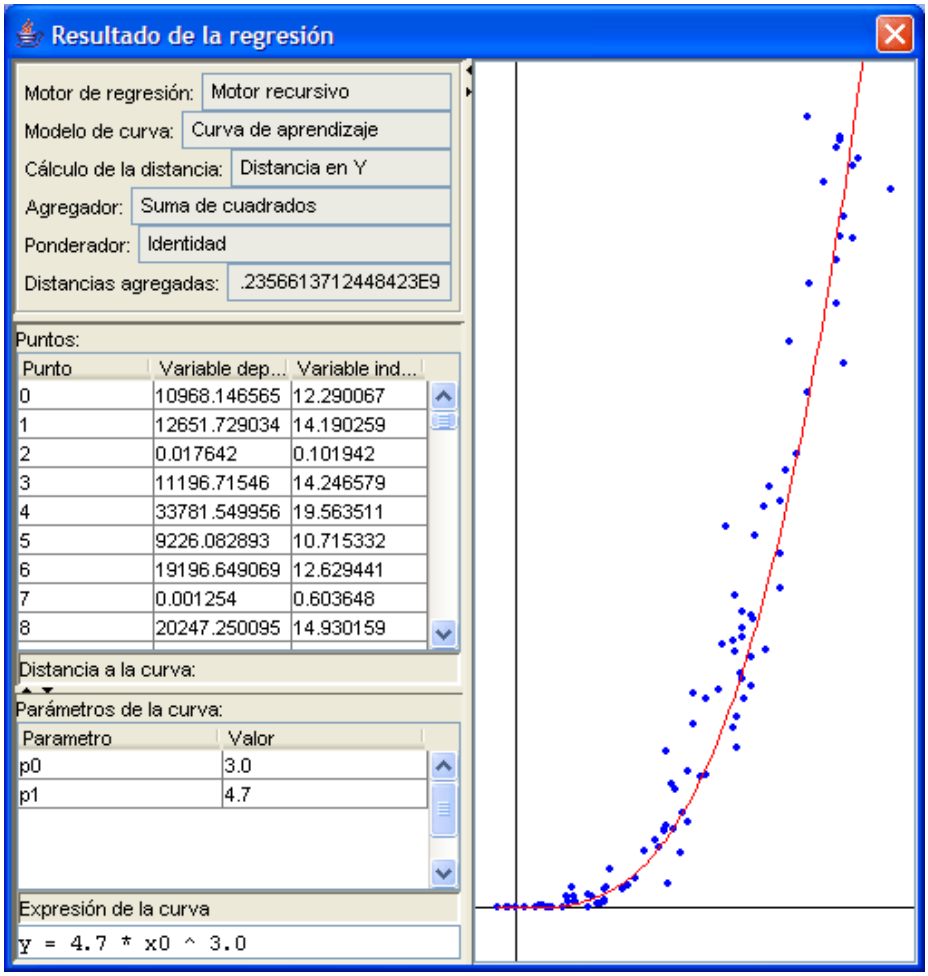
Resultados



Parámetros:

- Motor de regresión: motor recursivo
    - Rango: 2
    - Número de pasos: 100
    - Parámetro 0: 3
    - Parámetro 1: 5
- Modelo de curva: curva de aprendizaje
- Método de cálculo de la distancia: distancia en Y
- Ponderador de distancias: identidad
- Agregador de distancias: suma de cuadrados

Resultados



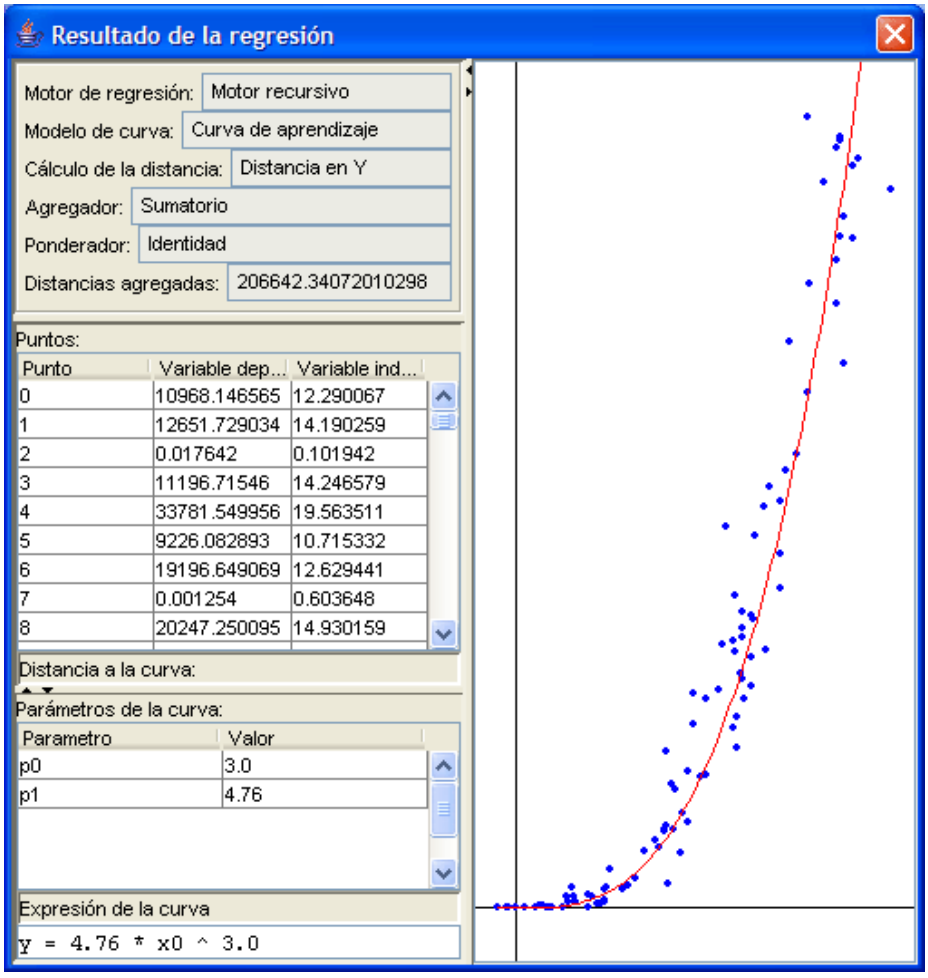
En este caso se observa que hay un mayor ajuste de los parámetros (concretamente del parámetro p1). Esto se debe a que se ha aumentado el número de pasos que realiza el motor y se ha reducido el rango, con lo que se disminuye la separación entre los valores de los parámetros que se comprobarán.

Parámetros:

- Motor de regresión: motor recursivo
    - Rango: 2
    - Número de pasos: 100
    - Parámetro 0: 3
    - Parámetro 1: 5

- Modelo de curva: curva de aprendizaje
- Método de cálculo de la distancia: distancia en Y
- Ponderador de distancias: identidad
- Agregador de distancias: sumatorio

Resultados



En este caso se ha cambiado el agregador de distancias con respecto al caso anterior. No obstante, el resultado es muy parecido, dado que ambos agregadores realizan operaciones parecidas.

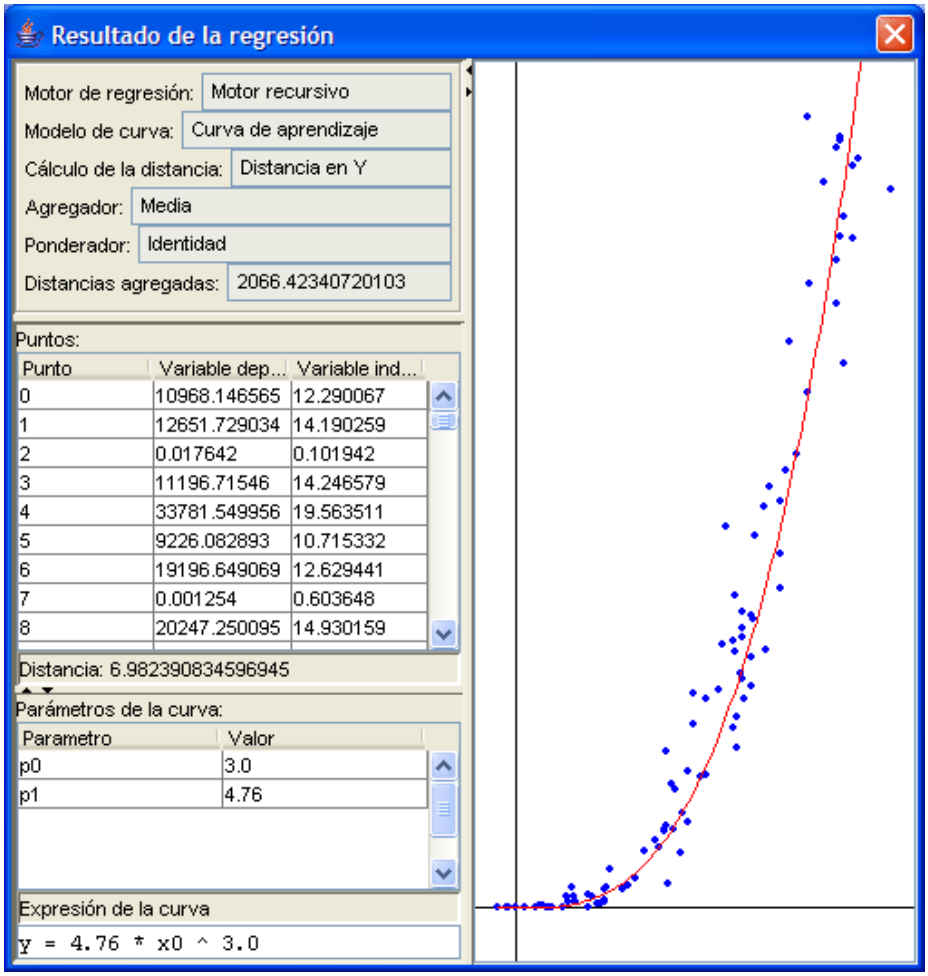
Parámetros:

- Motor de regresión: motor recursivo

- Rango: 2
- Número de pasos: 100
- Parámetro 0: 3
- Parámetro 1: 5

- Modelo de curva: curva de aprendizaje
- Método de cálculo de la distancia: distancia en Y
- Ponderador de distancias: identidad
- Agregador de distancias: media

Resultados

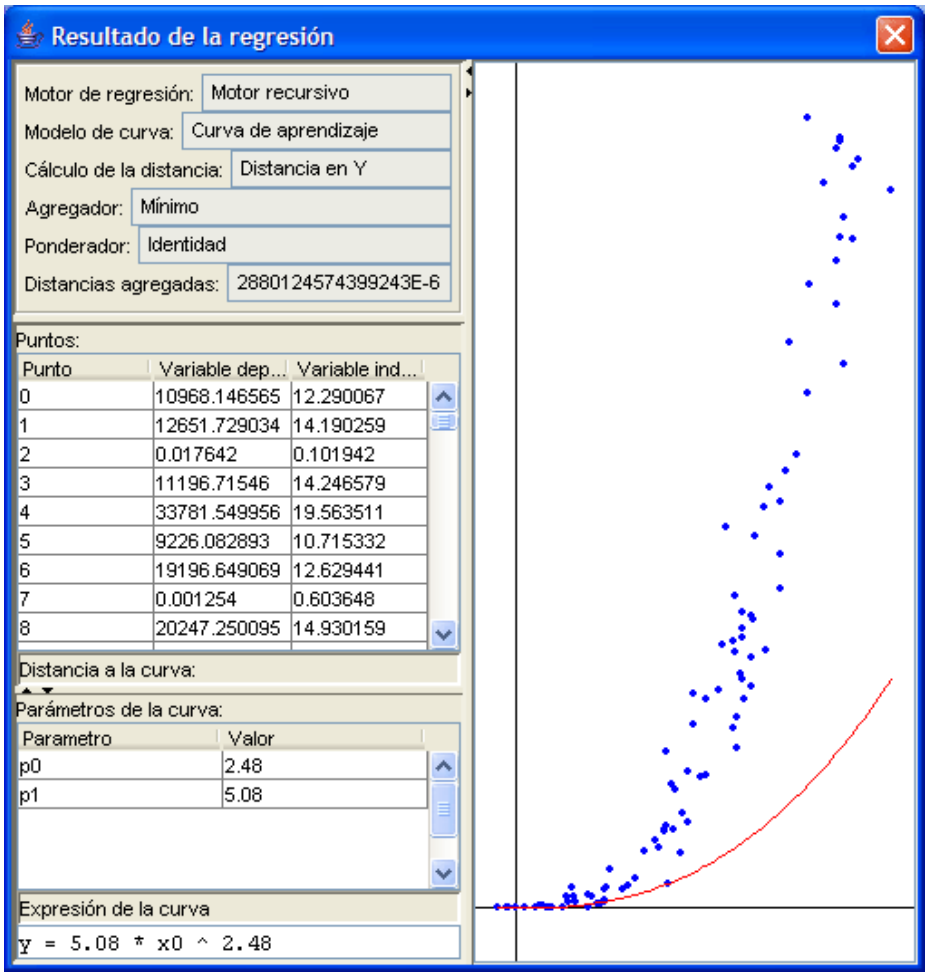


De nuevo se cambia el agregador. En este caso por la media, por lo que los parámetros obtenidos dan la curva de menor distancia media a los puntos.

Parámetros:

- Motor de regresión: motor recursivo
    - Rango: 2
    - Número de pasos: 100
    - Parámetro 0: 3
    - Parámetro 1: 5
- Modelo de curva: curva de aprendizaje
- Método de cálculo de la distancia: distancia en Y
- Ponderador de distancias: identidad
- Agregador de distancias: mínimo

Resultados



Al utilizar en este ensayo el agregador mínimo, se está obteniendo los parámetros que minimizan la distancia de la curva al punto más cercano, es decir, minimizan la distancia al de distancia mínima. Es por ello que la curva obtenida no sigue la distribución dada por los puntos, sino que se acerca lo más posible al que tiene distancia mínima, ignorando la distancia a los restantes.

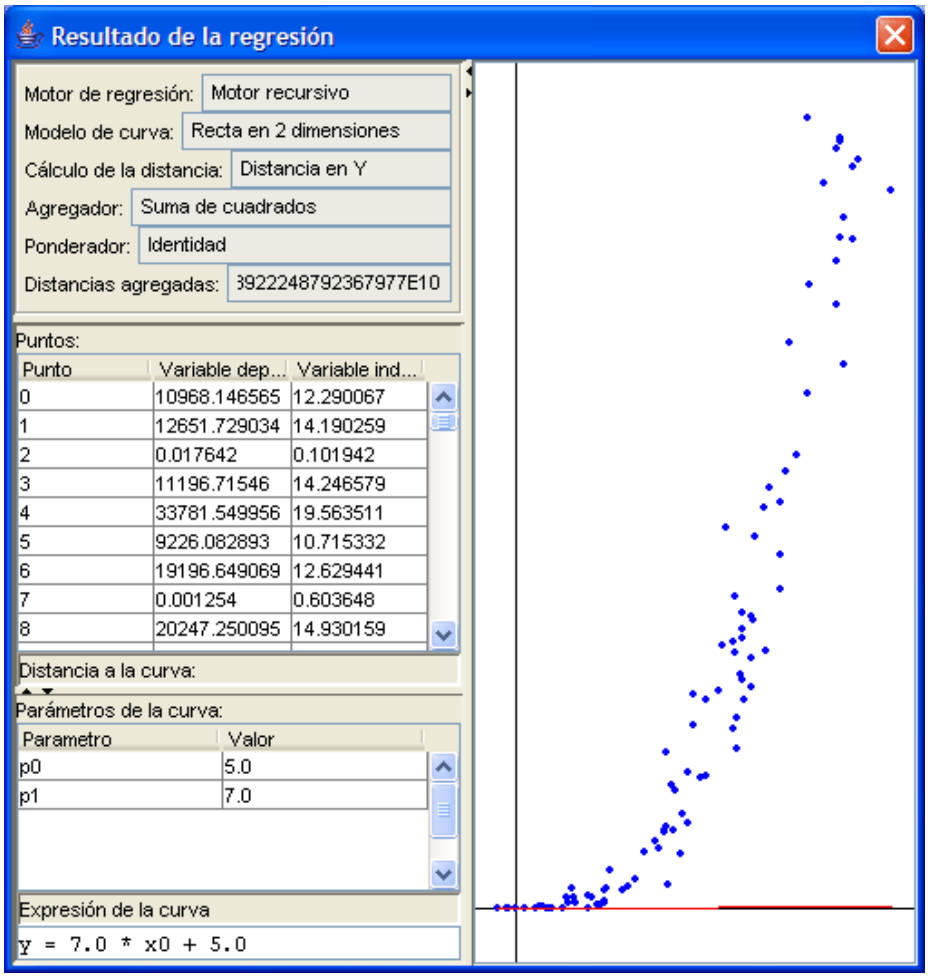
Parámetros:

- Motor de regresión: motor recursivo
    - Rango: 2
    - Número de pasos: 100
    - Parámetro 0: 3

- Parámetro 1: 5

- Modelo de curva: recta en 2 dimensiones
- Método de cálculo de la distancia: distancia en Y
- Ponderador de distancias: identidad
- Agregador de distancias: suma de cuadrados

Resultados

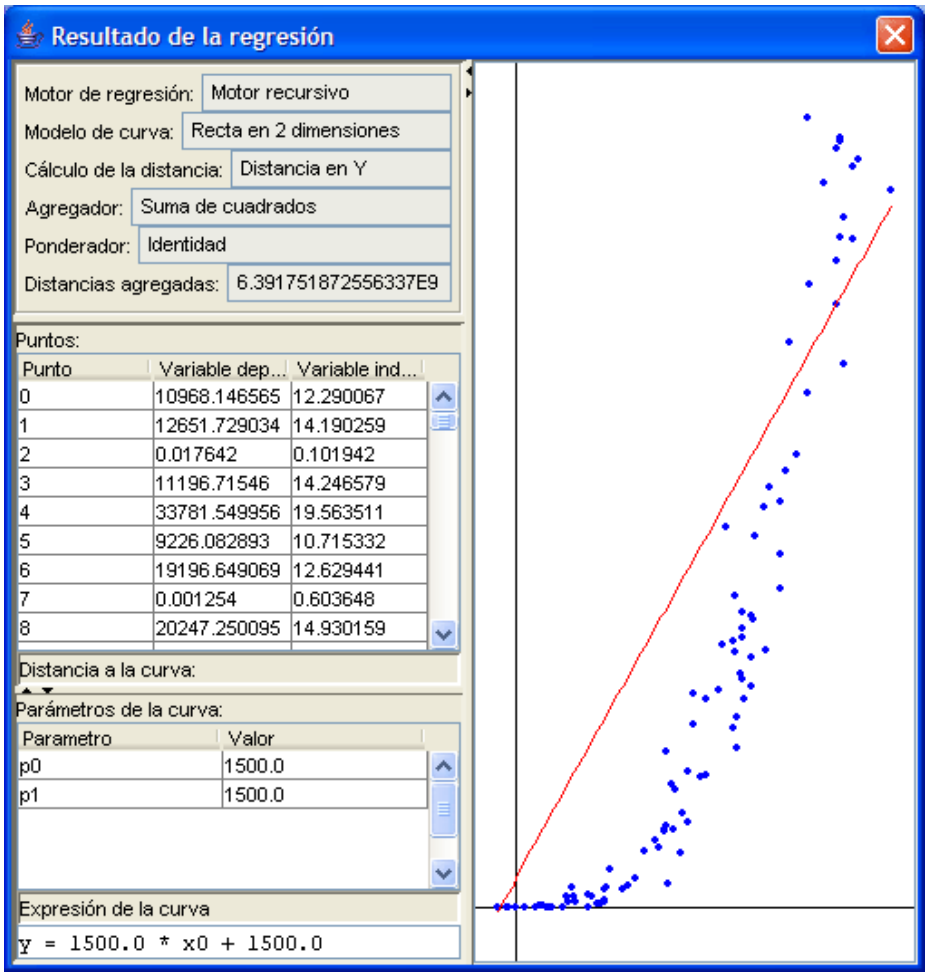


En este caso, la recta no sigue la distribución en absoluto. El motivo de esto son los parámetros de entrada introducidos en el motor recursivo. No son adecuados para este caso. Si se observan estos parámetros se puede comprobar que, al ser 2 el valor del rango, el parámetro p0 variará entre 1 y 5 y p1 entre 3 y 7. La recta necesitará mayor amplitud en los parámetros para poder encontrar un mejor ajuste.

Parámetros:

- Motor de regresión: motor recursivo
    - Rango: 200
    - Número de pasos: 100
    - Parámetro 0: 1700
    - Parámetro 1: 1700
- Modelo de curva: recta en 2 dimensiones
- Método de cálculo de la distancia: distancia en Y
- Ponderador de distancias: identidad
- Agregador de distancias: suma de cuadrados

Resultados



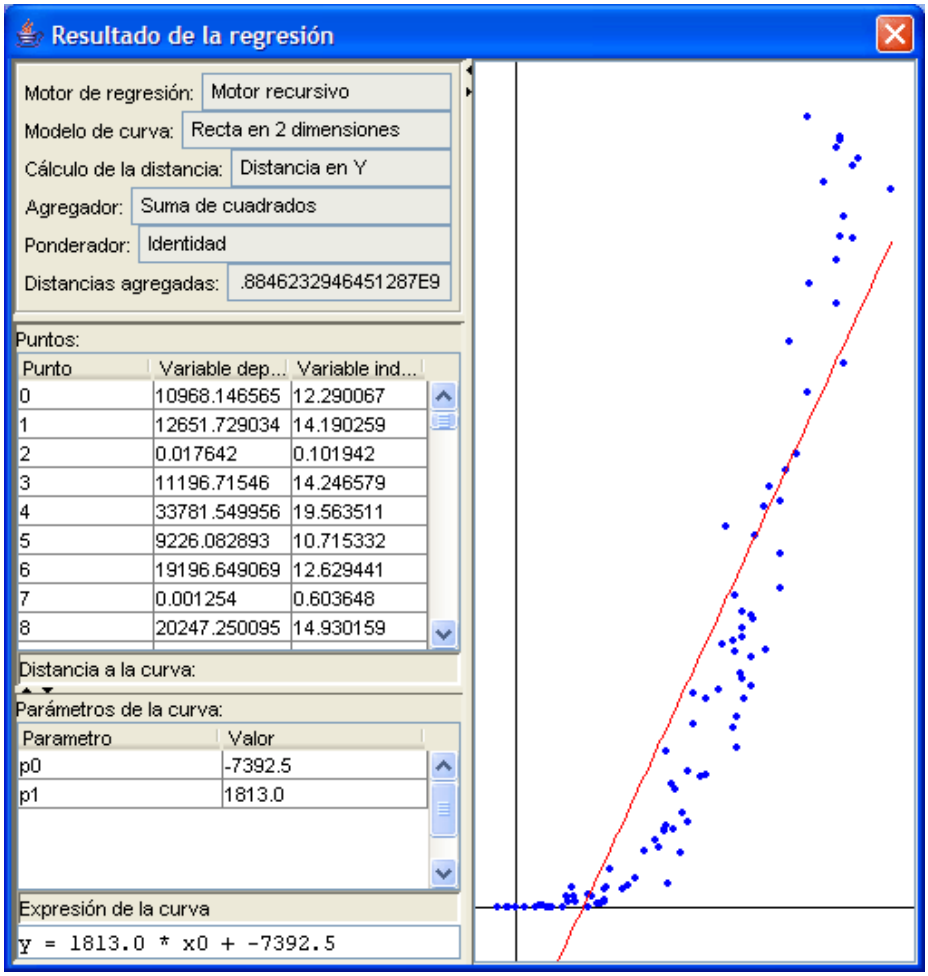
Ahora se obtiene una mejora con respecto al caso anterior. Como se puede observar, las distancias agregadas pasan, del orden de 10^{10} al orden de 10^{9}. Esto se ha conseguido al aumentar el rango de búsqueda de los parámetros de la curva y, sobretodo, al modificar los valores iniciales de los mismos.

Para ajustar mejor los parámetros se vuelve a realizar la regresión usando como valor inicial de los parámetros el resultado de esta regresión, es decir, 1500.

Se continúa este proceso hasta que los parámetros se estancan en torno a una cifra.

La sucesión de parámetros obtenida es la siguiente:

p0	p1
1500	1500
1300	1300
1100	1100
900	1200
500	1200
100	1200
-300	1300
-900	1300
...	...
-7395	1813



Se observa que la elección de parámetros iniciales es crucial para obtener buenos resultados con el motor recursivo.

Este proceso, realizado aquí manualmente, es el mismo que realiza automáticamente el motor recursivo con refinamientos.

7. Conclusiones y líneas de trabajo futuras

7.1. Conclusiones

Una vez desarrollado la herramienta de regresión borrosa y haber obtenido los resultados en los ensayos preliminares, se está en disposición de obtener las conclusiones que pueden extraerse del trabajo realizado.

Como se ha venido diciendo hasta ahora, al realizar un análisis de regresión, no siempre se dispone de un conjunto suficiente de observaciones, o se dispone de datos que no son “precisos”, es decir, que contienen algún tipo de imperfección, debido a la imprecisión o vaguedad de los mismos.

De cualquier forma, se tengan o no suficientes datos, contengan éstos imprecisiones o no, es posible encontrar la relación existente entre los datos de entrada y los de salida, decidiendo el mejor ajuste de los parámetros del modelo de curva que define dicha relación.

En el caso de que la muestra de observaciones no sea suficientemente grande o que las fuentes de información sean imprecisas, vagas o imperfectas, se deberán usar modelos paramétricos simbólicos.

En ocasiones el análisis de regresión borroso produce resultados que se ajustan mejor a los obtenidos mediante el análisis de regresión clásico. Dicha mejora es debida a la naturaleza difusa de los parámetros, que permiten una mejor representación de las observaciones.

El trabajo realizado ha permitido desarrollar una herramienta de regresión borrosa capaz de modelar la vaguedad o imprecisión en la información.

Debido a la inherente imprecisión del mundo real, se hace necesario utilizar modelos que permitan distintas formas de borrosidad en sus elementos. Dicha borrosidad se consigue mediante la asignación de funciones de pertenencia.

La herramienta desarrollada hace uso de modelos que permiten introducir borrosidad en las entradas, las salidas, los parámetros de los modelos de curva y en el cálculo de las distancias.

Dar borrosidad en las entradas y en las salidas, permite introducir la imprecisión o vaguedad en los valores observados de las variables independientes y en los valores observados para la variable dependiente.

La borrosidad en los parámetros de los modelos de curva representa las diferencias entre los valores observados y los estimados, debido a las inexactitudes en la estructura del sistema.

El grado de borrosidad en la medición de la distancia de un punto a la curva, vendrá dado por la fiabilidad que se espera en el resultado, o por la cantidad de aproximaciones realizadas.

En base al trabajo realizado y a los resultados obtenidos en los ensayos preliminares, se demuestra que el uso de la herramienta de regresión borrosa aporta evidencias suficientes de viabilidad.

La herramienta se ha desarrollado de forma específica para alcanzar de manera simple las necesidades expuestas en el trabajo.

Sin embargo, la simplicidad de su desarrollo, no disminuye la capacidad de posibilidades que ofrece la herramienta. El usuario podrá elegir entre diversos modelos de curvas, métodos de cálculo de distancias, ponderadores, agregadores, motores de regresión y tipos de borrosidad para los puntos que considere oportuno.

A la hora de elegir los elementos que formarán parte del proceso de regresión borrosa, el usuario, dependiendo de sus necesidades, deberá escoger los tipos de elementos más adecuados para llevarlo a cabo, ya que el resultado que se obtendrá estará determinado por dichos elementos.

Por ejemplo, el uso de los agregadores, ya sea al componer el resultado de la función de pertenencia de un punto borroso con respecto a sus componentes, o calcular la distancia nítida de un punto borroso a una curva, proporcionará resultados significativamente diferentes. Por lo que, el usuario, en función de sus necesidades deberá elegir el agregador más adecuado para obtener el resultado esperado.

Durante la fase de investigación del trabajo no se tuvo acceso a ninguna herramienta en el mercado que hiciera uso de la regresión borrosa, sin embargo, el estudio realizado, y a tenor de los resultados obtenidos, se pone de manifiesto una clara certeza de que la herramienta es necesaria para realizar determinados tipos de procesos de experimentación, como se ha podido ver a lo largo de este trabajo.

La herramienta de regresión borrosa es una generalización de las herramientas existentes en el mercado, realiza tanto regresión nítida como borrosa, es decir, si a los puntos de entrada no se les aplica ningún tipo de borrosidad, realizará la regresión nítida, característica propia de las herramientas encontradas.

La herramienta de regresión borrosa podrá ser aplicada en numerosos campos de la ciencia, tales como la Ingeniería del Software, Economía, Medicina, Psicología, Biología, Medio Ambiente, ... donde se precise una modelación borrosa , es decir, donde la medición de las variables presente dificultades de representación debido a la aparición de vaguedad, imprecisión o imperfección.

Se ha desarrollado esta herramienta sin ánimo de lucro, para ponerla a disposición de la comunidad de científicos y de todos aquellos que les pueda ayudar el uso de la regresión borrosa para llevar a cabo sus investigaciones.

7.2. Líneas futuras

La herramienta de regresión borrosa ha sido desarrollada con la intención de convertirse en el punto de partida de este tipo de aplicaciones, en este apartado se exponen las principales líneas de trabajo que establecen las bases del camino a seguir a partir de este trabajo.

Durante el desarrollo de la herramienta, se tuvo presente en todo momento uno de los objetivos fundamentales planteado desde un principio, desarrollar una herramienta abierta, con el fin de que pudiera ser utilizada y ampliada por futuros grupos de trabajo que siguieran la investigación.

El grupo de este trabajo, como primeros usuarios de la aplicación, ha detectado el primer cambio que se deberían realizar para mejorar la herramienta. Se trata de la flexibilidad de la gramática para el fichero de puntos.

En posteriores versiones se debería de implementar una gramática más flexible para el fichero de la lista de puntos, que se carga inicialmente. Además, se deberían implementar otros métodos alternativos para la introducción de dichos puntos, como puede ser la introducción manual.

La herramienta está diseñada de tal manera que se pueda añadir elementos a la aplicación de forma sencilla, cómoda y con la principal ventaja de reutilizar las clases desarrolladas. A continuación se detallan los elementos que permiten ampliar la herramienta, explicando las ventajas que aportaría a la misma.

Se podrán añadir distintos tipos de números borrosos, para ello será necesario implementar las funciones de pertenencia. Esto permitirá disponer de distintos tipos de borrosidad para que pueda ser aplicada a los elementos de la herramienta según considere oportuno el usuario.

Un modelo de curva representa un modelo paramétrico matemático simbólico, cuantos más modelos de curva estén implementados en la herramienta, más completa será y más posibilidades ofrecerá al usuario.

Se podrán añadir distintos métodos para calcular las distancias, ya sean métodos analíticos, por aproximación, para casos concretos o métodos para casos genéricos, que calculen la distancia de un punto n-dimensional a una curva arbitraria.

Otros elementos que se pueden incluir en la herramienta, son los agregadores. Los operadores de agregación son objetos matemáticos

cuya función es reducir un conjunto de números a un único número representativo o significativo. Tal y como se explicó en el apartado anterior, el uso de agregadores es un factor importante a la hora de usar la herramienta, por lo tanto, incluir otros tipos de agregadores, implica aumentar las posibilidades de uso.

Si las necesidades del usuario lo requieren, se pueden añadir distintos generadores de puntos, que generen distintas listas de puntos de acuerdo con los parámetros suministrados como entrada.

Se podrán incluir diferentes ponderadores tanto para los punto nítidos como para los puntos borrosos. Los nítidos permitirán al sistema dar diferente valor a las observaciones, y los ponderadores borrosos proporcionarán más credibilidad a determinadas observaciones. Con ello, se conseguirá dar pesos distintos a dichos valores en el proceso de regresión.

También se podrán incluir diferentes motores de regresión. El motor de regresión es el objeto que, a partir de los puntos, los modelos de curva, los métodos de cálculo de distancia, los agregadores y los ponderadores, dirige la regresión para obtener los parámetros que mejor se ajusten al modelo de curva.

Un motor de regresión muy interesante que se podría añadir, sería aquel que realizara el análisis de regresión para cada uno de los modelos de curva existentes en la aplicación, pudiendo de esta forma obtener, el modelo de curva que más se ajustara a las observaciones dadas, junto a los parámetros de dicho modelo.

Sin embargo, el añadir este tipo de motor de regresión implicaría realizar un número muy grande de cálculos, lo que hace pensar en el paralelismo. El proceso de regresión es inherentemente paralelo.

Además no hay que olvidar que la precisión del resultado es proporcional al número de cálculos realizados para su obtención, es decir, un mejor ajuste requerirá un mayor número de operaciones, por ejemplo, el motor recursivo con refinamiento alcanza un mejor ajuste que el normal, pero realiza más operaciones, por lo que tarda más tiempo en ejecutarse.

La computación paralela permite resolver problemas en menor tiempo de ejecución, ya que usa más procesadores, con mayor precisión, porque dispone de más memoria y además, permite resolver problemas más reales, haciendo uso de modelos matemáticos más complejos.

La herramienta de regresión borrrosa requiere mayor velocidad y memoria que la obtenida mediante la computación secuencial, por lo que una línea futura de trabajo será hacer uso de las ventajas que proporciona la computación paralela y que se describen en el Anexo I: Sistemas Distribuidos.

Por lo tanto, las principales líneas de continuación que se deberían seguir son el paralelismo y la ampliación de los elementos de la herramienta.

En el desarrollo de la aplicación se ha tenido un cuidado especial en conseguir que fuera lo más abierta posible, así mismo, se ha pretendido que el añadir elementos fuera una tarea sencilla, con el fin de que pudiera ampliarse con facilidad. Para incluir un elemento en la herramienta, simplemente hay que tener en cuenta las siguientes consideraciones.

- Se deberá crear una clase java que implemente el interfaz del elemento que se quiera añadir, dicha clase deberá tener como mínimo un constructor sin parámetros.

- Compilar la clase.

- Asegurarse de que clase compilada se encuentre en el classpath.

- Modificar el fichero xml correspondiente, que se encontrará en la carpeta xml. Dicho fichero deberá tener el siguiente formato:

    Nodo raíz elementos

    Y por cada elemento que exista en el gestor:

        Nodo hijo con los siguientes atributos

            - Clase completamente cualificada-calificada

            - Descripción corta

            - Descripción larga

Con formato: Numeración y viñetas

La herramienta de regresión borrosa desarrollada junto con las ampliaciones anteriormente explicadas, que deberán realizar futuros grupos, convertirán esta aplicación en una herramienta completa, con el fin de que sirva de ayuda para cualquier investigador que requiera su uso.

8. Anexos

8.1. Sistemas Distribuidos

Un sistema distribuido está compuesto por varios recursos informáticos de propósito general, tanto físicos como lógicos, que pueden asignarse dinámicamente a tareas concretas. Dichos recursos están distribuidos físicamente, tienen autonomía coordinada y funcionan gracias a una red de comunicaciones. [Ens78].

Las características de los sistemas distribuidos son:

- Cada elemento de computo tiene su propia memoria y su propio Sistema Operativo.
- Control de recursos locales y remotos.
- Sistemas Abiertos (Facilidades de cambio y crecimiento).
- Plataforma no standard (Unix, NT, Intel, RISC, Etc.).
- Medios de comunicación (Redes, Protocolos, Dispositivos, Etc.).
- Capacidad de Procesamiento en paralelo.
- Dispersión y parcialidad.

Los sistemas distribuidos permiten que los recursos disponibles en la red puedan ser utilizados simultáneamente por los usuarios y/o agentes que interactúan en la red.

Cada componente del sistema puede fallar independientemente, con lo cual los demás pueden continuar ejecutando sus acciones. Esto permite el logro de las tareas con mayor efectividad, pues el sistema en su conjunto continua trabajando.

Las ventajas de los sistemas distribuidos son las siguientes:

- Procesadores más poderosos y con menor coste: se desarrollan estaciones con mayor capacidad y satisfacen las necesidades de los usuarios.
- Avances en la tecnología de comunicaciones: permiten el desarrollo de nuevas técnicas mediante la disponibilidad de elementos de comunicación.
- Compartición de recursos: tanto dispositivos (hardware) como programas (software).

- Eficiencia y flexibilidad: respuesta rápida y ejecución concurrente de procesos mediante el empleo de técnicas de procesamiento distribuido.

- Disponibilidad y confiabilidad: sistema poco propenso a fallos y mayores servicios que elevan la funcionalidad.

- Crecimiento modular: rápida inclusión de nuevos recursos.

Pero no todo son ventajas, los sistemas distribuidos también tienen sus inconvenientes, son los siguientes:

- Requerimientos de mayores controles de procesamiento.

- Velocidad de propagación de información: es muy lenta a veces.

- Servicios de replicación de datos y servicios con posibilidades de fallos.

- Mayores controles de acceso y proceso.

- Administración más compleja.

- Mayor coste.

MPI (Message Passing Interface)

MPI es un estándar creado por un amplio comité de expertos y usuarios con el objetivo de definir una infraestructura común y una semántica específica de interfaz de comunicación.

El paso de mensajes es una tarea ampliamente usada en ciertas clases de máquinas paralelas, especialmente aquellas que cuentan con memoria distribuida.

Al diseñarse MPI, se tomaron en cuenta las características más atractivas de los sistemas existentes para el paso de mensajes y se intentó establecer un estándar práctico, portable, eficiente y flexible.

En el modelo de programación MPI, un computo comprende de uno o más procesos comunicados a través de llamadas a rutinas de librerías para mandar (send) y recibir (receive) mensajes a otros procesos.

En la mayoría de las implementaciones de MPI, se crea un conjunto fijo de procesos al inicializar el programa, y un proceso es creado por cada tarea. Sin embargo, estos procesos pueden ejecutar diferentes programas. De ahí que, el modelo de programación MPI es algunas veces referido como *MPMD* (multiple program multiple data) para distinguirlo del modelo *SPMD*, en el cual cada procesador ejecuta el mismo programa.

Debido a que el número de procesos en un cómputo de MPI es normalmente fijo, se puede enfatizar en el uso de los mecanismos para comunicar datos entre procesos. Los procesos pueden utilizar operaciones de comunicación punto a punto para mandar mensajes de un proceso a otro, estas operaciones pueden ser usadas para implementar comunicaciones locales y no estructuradas. Un grupo de procesos puede llamar colectivamente operaciones de comunicación para realizar tareas globales tales como broadcast, etc.

La habilidad de MPI para probar mensajes da como resultado el soportar comunicaciones asíncronas. Probablemente una de las características más importantes del MPI es el soporte para la programación modular. Un mecanismo llamado comunicador permite al programador del MPI definir módulos que encapsulan estructuras internas de comunicación (estos módulos pueden ser combinados secuencialmente y paralelamente).

Aunque MPI es un sistema complejo y multifacético, podemos resolver un amplio rango de problemas usando seis de sus funciones, estas funciones inician y terminan un cómputo, identifican procesos, además de mandar y recibir mensajes.

MPI_INT: Inicia un computo.

MPI_INT(int *argc,char ***argv), argc, argv son requeridos solo por el contexto del lenguaje C, en el cual son los argumentos del programa principal.

MPI_FINALIZE: Termina un computo.

MPI_FINALIZE().

MPI_COMM_SIZE: Determina el número de procesos en un computo.

MPI_COMM_SIZE(comm,size)

comm (IN): comunicador (manejador[handle])

size (OUT): número de procesos en el grupo del comunicador(entero).

MPI_COMM_RANK: Determina el identificador del proceso actual “mi proceso”.

MPI_COMM_RANK(comm,pid)

comm (IN): comunicador (manejador[handel])

pid (OUT): identificador del proceso en el grupo del comunicador(entero).

MPI_SEND: Manda un mensaje.

MPI_SEND(buf, count, datatype, dest, tag, comm).

buf (IN): dirección del buffer a enviar (tipo x)

count (IN): número de elementos a enviar del buffer (entero>=0)

datatype (IN): tipo de datos del buffer a enviar (handle)

dest (IN): identificador del proceso destino (entero)

tag (IN): message tag (entero)

comm (IN): comunicador (handle) .

MPI_RECV: Recive un mensaje.

MPI_RECV(buf,count,datatype,source,tag,comm.,status).

buf (OUT): dirección del buffer a recibir (tipo x)

count (IN): número de elementos a recibir del buffer (entero>=0)

datatype (IN): tipo de datos del buffer a recibir (handle)

source (IN): identificador del proceso fuente, o MPI_ANY_SOURCE(entero)

tag (IN): message tag, o MPI_ANY_TAG(entero)

comm (OUT): comunicador (handle)

status (OUT): estado del objeto (estado)

Los parámetros IN son parámetros que la función usa pero no modifica lo modifica y los parámetros OUT son aquellos que la función no usa pero puede modificar modificarlos.

OpenMP

El estándar OpenMP describe un API de programación SMP (Clusters de Multiprocesadores Simétricos), mediante directivas, rutinas de control y variables de entorno.

OpenMP reúne una serie de características que no poseen otros modelos de programación:

- Soporta paralelismo incremental.

- Apto para todo tipo de grano.
- Paralelismo en datos y en control parcialmente soportado.
- Totalmente portable.
- Soporta paralelismo anidado.
- Permite la creación de librerías paralelizadas gracias a la posibilidad de usar directivas huérfanas.

Las directivas de OpenMP informan de la paralelización entre fragmentos de código, bucles y subrutinas.

Actualmente OpenMP es soportado por la mayoría de los fabricantes. La implementación es libre y el uso correcto es responsabilidad del usuario.

Grid Computing

La necesidad de aprovechar los recursos disponibles en los sistemas informáticos conectados a Internet y simplificar su utilización ha dado lugar a una nueva disciplina en tecnología de la información conocida como Grid Computing.

El objetivo principal de la tecnología Grid Compuiting es la compartición de recursos en Internet de forma uniforme, transparente, segura, eficiente y fiable.

La tecnología Grid permite interconectar recursos en diferentes dominios de administración respetando sus políticas internas de seguridad y su software de gestión de recursos de Intarnet.

Uno de los problemas actuales es que un sistema no es capaz de almacenar el volumen de información que se necesita, mediante la arquitectura Grid se solventa el problema de adquirir mayor capacidad de procesamiento, ya que se servirá de la red para hacer uso de los recursos distribuidos de una manera eficiente, es decir, utilizará los recursos según se vayan necesitando, sin tener que preocuparse de donde se encuentren físicamente localizados.

La idea de Grid es ofrecer un único punto de acceso a un conjunto de recursos distribuidos geográficamente (supercomputadores, clusters, almacenamiento, fuentes de información, instrumentos, personal...). De este modo, los sistemas distribuidos se pueden emplear como un único sistema virtual en aplicaciones intensivas en datos o con gran demanda computacional.

Las aplicaciones de la tecnología Grid son innumerables, se ha utilizado para resolver grandes problemas de computo, como procesar datos de investigaciones, simulaciones científicas, procesamientos de datos estadísticos y simulaciones técnicas.

Las infraestructuras Grid permiten satisfacer las demandas de computación y gestión de datos de las aplicaciones de gran desafío y facilitan la compartición de recursos para resolver demandas puntuales.

Por otro lado, al igual que ocurrió con Internet, la tecnología Grid será decisiva para el desarrollo empresarial dando lugar a innumerables ideas de negocio como pueden ser el desarrollo de organizaciones virtuales, la comercialización en demanda de recursos, los proveedores de recursos... En definitiva, nos encontramos ante la tecnología que los analistas definen como la siguiente revolución en Internet.

8.2. Diccionario de clases (Javadoc)

Se ha empleado la herramienta **javadoc** para generar un diccionario de clases en el que se incluyen todos los paquetes, las clases y los métodos que componen la aplicación, así como las relaciones existentes entre ellos. Dado el volumen de la documentación (más de 200 páginas), se ha optado por incluirla en un documento aparte incluido en el CD entregado junto con este documento. En él se reproduce esta documentación, convertida del formato HTML en que es generada por la herramienta, y en inglés, ya que no se ha encontrado ningún **doclet** que dé la salida en castellano.

8.3. Artículo CEDI 2005

Una herramienta de regresión borrosa

Raquel Ballester	José I. del Campo	Gonzalo Flórez Puga	F. Javier Crespo
Alumno de la FdI	Alumno de la FdI	Alumno de la FdI	Profesor Tutor de SI
U .Complutense de Madrid	U .Complutense de Madrid	U .Complutense de Madrid	U Complutense de Madrid
raquelballester@terra.es	campo_ji@yahoo.es	gflorezpuga@yahoo.es	javier.crespo@fdi.ucm.es

Resumen

El uso de las técnicas de regresión sobre las observaciones experimentales ha permitido el estudio de numerosos fenómenos en diversos campos de la ciencia, y muy especialmente en las ciencias sociales. Dichas técnicas requieren de un número suficiente de observaciones "precisas", exactas y fiables. Sin embargo, no siempre es posible obtener el conjunto de observaciones necesario, o éstas contienen algún tipo de imperfección en los datos, debido a la imprecisión o vaguedad de los mismos.

En cualquier caso, con suficientes datos o no, con imperfecciones o no, los modelos obtenidos deberían proveer de capacidades predictivas y descriptivas [3]. Las actuales herramientas, o las más fácilmente accesibles, tienen limitado el uso de modelos y difícilmente usan las técnicas de la teoría de conjuntos borrosos.

Se propone en este trabajo una herramienta abierta de regresión que admita el uso de cualquier modelo de curva independientemente de su naturaleza. Además, esta herramienta permitirá el uso de diferentes formas de borrosidad y por su diseño permitiría cualquier modelo propuesto por el usuario si éste prevé que éstos tienen características que sean suficientemente predictivas y descriptivas.

Esta primera aproximación de una herramienta abierta de regresión se realiza un estudio sobre diferentes modelos paramétricos simbólicos, usados comúnmente en la práctica en disciplinas tan heterogéneas como pueden ser la Ingeniería del Software, la Economía o en cualquier campo en donde pueda aparecer imprecisiones en la información.

1. Introducción

El hombre tiene la capacidad de percibir el mundo que le rodea y expresarlo mediante el lenguaje natural. Dicho lenguaje es una forma de comunicación imprecisa y ambigua que se apoya en el conocimiento compartido por aquellos con los que se comunica [11].

En el lenguaje natural se describen objetos o situaciones en términos imprecisos: grande, joven, tímido, alto, bonito... La representación del conocimiento basado en estos términos no puede ser exacta, quizá impreciso, pero no exacta, ya que normalmente representan impresiones subjetivas o percepciones, tal y como analiza Lofti A. Zadeh, profesor de la Universidad de California en Berkeley, en su Teoría computacional de la percepción [23], en la cual trata un proceso de razonamiento automatizado en donde se opera sobre la base de percepciones en lugar de sobre medidas.

Diversos campos de la Ciencia como la Agricultura, Química, Medicina, Medio Ambiente, Psicología, Biología, Economía... usan modelos lineales para su desarrollo, lo que ha supuesto un gran avance, no solo por los desarrollos matemáticos alcanzados, sino también por su aplicación en situaciones reales. Sin embargo, estas relaciones no son de utilidad a la hora de usar variables lingüísticas, ya que los modelos lineales se basan en datos obtenidos por medio de la planificación y diseño de experimentos.

Los modelos paramétricos matemáticos usados en la actualidad trabajan bajo el dominio de los enteros, obviando las imprecisiones en los datos, por ello, si el objetivo es conseguir resultados más representativos, se hace necesario usar modelos paramétricos matemáticos simbólicos, que sean más predictivos y descriptivos [3].

En los modelos en donde los datos sean insuficientes o imperfectos, originados por la imprecisión o vaguedad, se ha demostrado útil el uso de un análisis borroso [1], [9], [10], [19], [13], [8], [21], [20].

El análisis de regresión borroso ha sido estudiado y aplicado en diferentes áreas tal como el modelado de datos económicos o financieros, la ingeniería del software, etc., [2], y se han obtenido resultados comparables y que aportan ajustes, en muchos casos superiores, a los obtenidos con el análisis de regresión clásico. Sin embargo, resulta extraño la dificultad de encontrar herramientas que permitan como entrada diferentes formas de borrosidad, con el fin de obtener una mayor descripción de la información que se pretende representar.

En este trabajo se propone una herramienta abierta mediante regresión nítida y borrosa, que mediante el uso de las técnicas de la Teoría de Conjuntos Borrosos, incrementa la complejidad de los modelos paramétricos existentes, para obtener un mejor reflejo de la realidad.

A continuación se presenta el desarrollo de la herramienta, quedando organizado el resto del artículo de la siguiente manera:

En la segunda sección, se verán algunos sistemas existentes en la actualidad, tanto nítidos como borrosos, y se presentarán las limitaciones que se tienen cuando se pretende realizar un modelo más preciso y descriptivo.

En la tercera sección, se presentará el modelo de la herramienta, la interfaz y los diagramas de clases y secuencia.

En la cuarta sección, se realizará una primera aproximación y se obtendrán algunos resultados de experimentos que aportarán evidencias suficientes de las cualidades de la herramienta que invitarán a continuar los ensayos.

La quinta sección contiene anotaciones y conclusiones.

2. Algunas herramientas de regresión actuales

El objetivo de esta sección es exponer el contexto actual del análisis de regresión mediante el estudio de las principales herramientas que actualmente se usan en el mercado. Se describen a continuación algunas de ellas:

SPSS (Statistical Product and Service Solutions) [5] es un paquete de software estadístico y tratamiento de datos con más de 35 años de experiencia en el sector. Engloba una línea de productos que es modular, integrada y con todas las funcionalidades necesarias para llevar a cabo cada paso del proceso analítico - planificación, recogida de datos, acceso y preparación de los datos, análisis, creación de informes y distribución de los mismos.

La interfaz gráfica de usuario lo hace sencillo de utilizar dado que le proporciona toda la gestión de los datos, los estadísticos y los métodos de creación de informes que necesita para realizar todo tipo de análisis.

SPSS proporciona los procedimientos estadísticos más usuales para análisis básico, incluyendo: sumas, frecuencias, tablas de contingencia, estadísticos descriptivos, análisis factorial y regresión.

En el módulo de base se dispone de una serie de modelos de regresión: lineal, logarítmico, inverso, cuadrático, cúbico, compuesto, potencial, S, creciente, exponencial y logístico. Respecto al ajuste de curvas, hay disponibles para especificar 11 tipos de curvas.

El módulo "SPSS Modelos de Regresión" [6] permite ir más allá del análisis de datos básico, con características añadidas como regresión logística multinomial, regresión no lineal restringida y no restringida, mínimos cuadrados ponderados, PROBIT...

Otra herramienta destacada es ***SAS/STAT*** [15]. Una de las características principales de SAS/STAT es su versatilidad, la cual hace posible evaluar datos provenientes de una gran variedad de disciplinas. La tecnología empleada permite aplicar un extenso conjunto de técnicas especializadas para cada tipo de industria.

SAS proporciona un gran conjunto de herramientas para el análisis estadístico, incluyendo análisis de varianza, regresión, análisis de dato por categorías, análisis multivariante, análisis no paramétrico...

Con respecto a la regresión, el procedimiento general que emplea SAS/STAT se basa en mínimos cuadrados para estimar los parámetros, incluyendo 9 técnicas de selección de modelo diferentes, pudiendo obtener diferentes diagnósticos y medidas. El uso de modelos más especializados se realiza ajustándolos a modelos lineales como los mixtos, no lineales, curvas cuadráticas (quadratic response surface models)...

Statgraphics [16] incluye las utilidades necesarias para el análisis estadístico de datos (análisis estándar para descripción, comparación de datos, análisis multivariante, análisis de series temporales, regresión avanzada, análisis para el control de calidad, y diseño de experimentos), gráficos interactivos, y gráficos e informes para presentaciones.

Statgraphics tiene una estructura modular constituida por 3 módulos diferentes, en los que se puede encontrar más de 150 procedimientos de distribución.

El módulo básico (Standard Edition) aporta todas las herramientas estadísticas básicas, entre ellas la regresión simple, múltiple y polinomial. A partir de éste se pueden seleccionar las funciones estadísticas adicionales necesarias en los otros módulos.

En la versión Professional, el módulo de regresión avanzada permite efectuar una exploración completa de los datos, formular modelos de regresión múltiple complejos, ajuste no lineal, validar los métodos de un laboratorio o simplemente buscar el mejor modelo de regresión.

R [14], [22] es un sistema para análisis estadísticos y gráficos. Tiene una naturaleza doble de programa y lenguaje de programación y es considerado como un dialecto del lenguaje S creado por los Laboratorios AT&T Bell. R se distribuye gratuitamente bajo los términos de la GNU General Public Licence (con website en http://www.gnu.org).

Consta de un "sistema base" y de paquetes adicionales que extienden la funcionalidad.

Permite realizar regresión múltiple y polinomial, y algunos modelos no lineales, aunque en la mayoría de los casos han de ser aproximados. Las funciones para ello son glm(), optim() y nlm(). También es posible realizar regresión robusta y regresión local aproximada con algunos paquetes adicionales.

Otras facilidades disponibles en R para regresión y análisis de datos son la posibilidad de emplear modelos mixtos, aditivos y arborescentes.

Matlab [12] es un lenguaje de computación técnica de alto nivel y un entorno interactivo para el análisis de datos y el desarrollo de algoritmos y aplicaciones. Matlab puede emplearse en un amplio rango de aplicaciones, incluyendo procesamiento de señal e imagen, comunicaciones, control digital, modelado financiero, análisis estadístico, biología computacional...

Diversas herramientas y add-on (colecciones de funciones de propósito específico, disponibles por separado) extienden el entorno base para resolver problemas específicos de ciertas áreas. Entre ellos se encuentra el "Statistics Toolbox", el cual da soporte a una gran variedad de tareas estadísticas comunes, desde la generación de números aleatorios al ajuste de curvas [18]. Este add-on distingue dos categorías de herramientas:

- Herramientas de probabilidad y estadística.

- Herramientas relacionadas con el aspecto gráfico e interfaces(GUI).

En la primera categoría, y dentro del área que nos ocupa, se encuentran:

- Modelos lineales: ANOVA, análisis de covarianza (ANOCOVA), regresión lineal múltiple, regresión por pasos, superficies de predicción, regresión contraída, análisis multivariante de la varianza (MANOVA). También da soporte a versiones no paramétricas de ANOVA.

- Modelos no lineales: Para los modelos no lineales, este add-on provee de funciones para la estimación de parámetros, predicción interactiva y visualización de ajustes no lineales multidimensionales, intervalos de confianza para predicciones de valores y parámetros. También contiene funciones para la clasificación y el establecimiento de árboles de regresión para aproximar cualquier tipo de relación que pudiera derivarse de ella.

Statistica [17] ofrece un gran conjunto de herramientas para el manejo, análisis y la visualización de datos, incluyendo procedimientos de data mining. Su tecnología incluye una amplia selección de módulos predictivos, clustering, clasificación y técnicas de exploración en una única plataforma software. Actualmente cuenta con más 20 años de experiencia en el sector.

Esta herramienta está disponible en 4 categorías (Enterprise, Web-based Analytic Applications, Data Mining Solutions, Desktop). La funcionalidad de cada categoría puede ser ampliada con módulos.

Con respecto a la regresión, el módulo base ofrece un conjunto de implementaciones de regresión linear, entre ellas simple, múltiple, paso a paso, jerárquica, no linear (polinómica, exponencial, logarítmica...). Las características de análisis de datos anómalos y de residuos incluyen una amplia selección de gráficos.

Métodos de regresión más avanzados se proporcionan en el módulo de Regresión General(GRM), como la regresión por subconjuntos, regresión paso a paso multivariante, para múltiples variables dependientes...

Otro módulo de interés es el de Modelos Avanzados, que ofrece una amplia elección de herramientas de modelado y previsión (por ej. modelos lineales, modelos lineales/no lineales generalizados, regresión Quick Logit/Probit, ANOVA/ANCOVA, análisis de supervivencia, series cronológicas y previsión), incluyendo selección automática de modelos y herramientas de visualización interactivas. Con este módulo se puede estimar prácticamente cualquier modelo no lineal definido por el usuario y caracterizar un conjunto de modelos predefinidos.

FuReA [7] es una herramienta para el análisis de regresión borroso, capaz de trabajar con datos imprecisos o inciertos.

La lógica borrosa es una técnica ampliamente usada con un gran número de aplicaciones. La teoría de conjuntos borrosos, en la cual se basa Furea, fue presentada por el Prof. Lotfi A. Zadeh en 1965.

La idea original de análisis de regresión borrosa fue propuesta por Tanaka et al. en 1982 [20] .

El análisis de regresión borrosa asume que alguno de los componentes del sistema es descrito mediante conjuntos borrosos, más concretamente mediante números borrosos.

Un conjunto borroso es aquel en el cual cada elemento posee un cierto grado de pertenencia al conjunto. En el análisis de regresión, los outliers siempre suponen una dificultad por el hecho de que modelan errores. Pero bajo ciertas circunstancias, también pueden ofrecernos valiosa información. Este es el motivo por el que la tarea de identificación y análisis de outliers presenta un doble interés desde el punto de vista técnico.

Se ha demostrado que el método de f-regresión empleado en Furea ofrece una mejor capacidad de detección de outliers y puede ser aplicado satisfactoriamente al análisis de regresión borrosa.

Furea permite introducir datos experimentales y añadir variables obtenidas indirectamente mediante transformaciones. También es posible construir arbitrariamente modelos lineales o no lineales usando variable definidas, y realizar cálculos que den valores más precisos de los parámetros del modelo y encontrar *outliers*.

Todo esto permite observar como el modelo calculado se ajusta a los datos experimentales.

2.1. Limitaciones

Según se ha expuesto, estas herramientas presentan una serie de limitaciones para el tratamiento de la imprecisión, la vaguedad y la incertidumbre:

- Están sometidas a procesos transparentes que no pueden ser controlados por el usuario.

- La propia potencia o rapidez de ejecución restringe el uso de otros procedimientos de la literatura del área, empleando la herramienta en cuestión métodos específicos implementados por los creadores de la misma.

- El desarrollo de herramientas de fácil acceso se ha enfocado en arquitecturas monoprocesador, lo que impide el empleo de éstas en proyectos de gran envergadura que exigen una gran cantidad de cómputo.

- Las herramientas expuestas en esta sección o bien no tratan la borrosidad, o bien no permiten una suficiente diversidad de métodos y modelos de regresión.

Por estos motivos se considera oportuno el desarrollo de una herramienta práctica que tenga en cuenta todas estas limitaciones.

3. Primera aproximación a una herramienta abierta para la regresión borrosa.

En esta sección se presenta un diseño de una "herramienta abierta" de regresión borrosa.

Esta herramienta es una extensión de la regresión paramétrica con números nítidos, que se generaliza haciendo uso de las técnicas de la Teoría de Conjuntos Borrosos y marcos matemáticos relacionados.

El objetivo principal planteado, que sea totalmente abierta, se concreta en que no presente limitaciones de ningún tipo en cuanto a incluir borrosidad. Este uso de las técnicas de la teoría de conjuntos borrosos debe poder concretarse tanto en los datos como en los procesos del procedimiento, no debe limitar los modelos a estudiar ni por su forma ni por el método que se desee emplear para mejorar su ajuste y debe permitir obtener las metas elegidas de predicción y descripción sin inconvenientes e inciertos pasos que oscurezcan el desarrollo u obtención de los parámetros de mejor ajuste del modelo o modelos propuestos.

A continuación se hace un estudio detallado de la aplicación desarrollada mediante la descripción de los paquetes que la componen.

Puntos. Este paquete contendrá las implementaciones de las clases que representan los puntos, tanto nítidos como borrosos, tanto las que actualmente están implementadas como las que incorporen futuros grupos de trabajos de Sistemas Informáticos de esta facultad que continúen el trabajo de investigación o colaboradores externos. También se incluyen en él otras clases utilizadas para su mantenimiento o representación gráfica.

Todos los puntos heredan de la clase abstracta *Punto*, en la que se define el comportamiento que debe poseer un punto en esta aplicación.

Esta clase permite implementar diferentes tipos de números borrosos, según su función de pertenencia, ya sea triangular, L-type, parabólica, etc. También permite representar puntos nítidos.

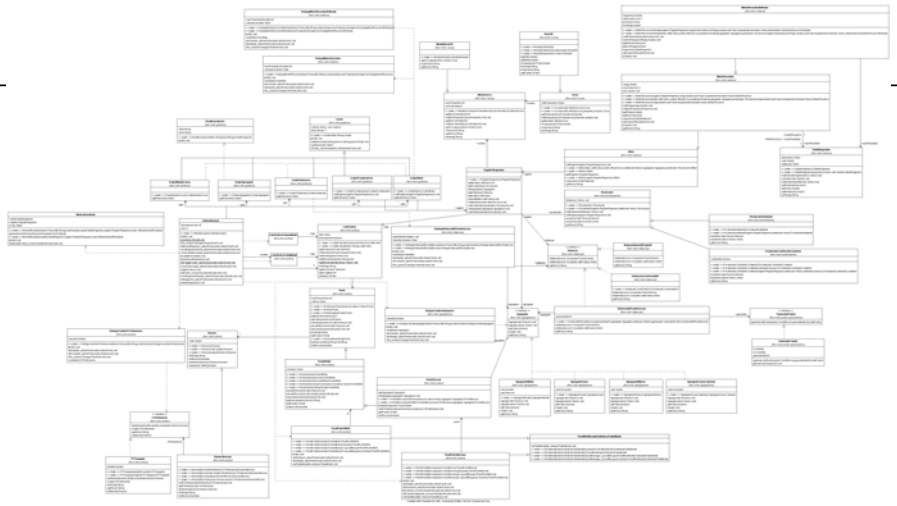
Se han realizado dos implementaciones de los puntos, una para puntos nítidos y otra para puntos borrosos.

La clase *PuntoNitido* representa los puntos nítidos n-dimensionales. Esta clase está formada por una lista de valores de la clase *Numero*. En ella existen operaciones que permiten cargar puntos desde un fichero, cambiar el número de dimensiones del punto, obtener y modificar los valores de cada una de sus componentes y asignar un nombre descriptivo a cada una de ellas.

La clase *PuntoBorroso* es una subclase de *PuntoNitido* y en ella se extiende esta clase para poder representar puntos borrosos. Cada instancia de esta clase está formada por una lista de instancias de la clase *NumeroBorroso* y tiene un agregador asociado. Este agregador se emplea para componer el resultado de la función de pertenencia del punto borroso como agregación de las funciones de pertenencia de cada una de sus componentes.

Todos los puntos que representan las observaciones realizadas por el usuario se incluyen en una lista de puntos, implementada en la clase *ListaPuntos*, y sobre ellos se realiza la regresión. El uso de la clase abstracta *Punto* permite a la herramienta realizar el proceso de regresión sobre una lista de puntos heterogéneos.

En este paquete, además de *Puntos* y *ListaPuntos*, se han incluido las clases que representan las funciones de pertenencia de los números borrosos; todas ellas deben implementar la interfaz *FcPertenencia*, que define los métodos básicos que deben poseer todas las funciones de pertenencia.



Curvas. Dentro del paquete *curvas* se incluyen las clases en que se implementan los modelos de curva que servirán para construir las instancias de las curvas de la aplicación.

Un modelo de curva representa un modelo paramétrico matemático simbólico de un tipo de curva. Este modelo, junto con un vector de parámetros de la clase *Número*, es una instancia de curva, y representa una curva determinada.

Por ejemplo, el modelo para una recta será:

$$y = t_1 \cdot x_0 + t_0$$

Si a este modelo se le asigna un vector de parámetros, por ejemplo (1, 2), se obtiene la siguiente instancia de curva:

$$y = 2 \cdot x_0 + 1$$

Los parámetros asignados a un modelo de curva para obtener una instancia de curva pueden ser tanto nítidos como borrosos.

En este paquete se encuentra la clase abstracta *ModeloCurva*, que determinará el comportamiento de todos los modelos de curvas simbólicas que se pueden usar en el proceso de regresión, con el fin de conseguir la modelización más descriptiva y predictiva.

Distancias. El paquete *distancias* se compone de distintos métodos de cálculo de la distancia existente entre una curva y un punto. Esta puede ser borrosa o nítida.

En este paquete tienen cabida todos los métodos de cálculo de la distancia, desde los más concretos, como la distancia existente entre un punto y una recta en dos dimensiones usando la fórmula analítica, hasta métodos más genéricos de cálculo de distancias de un punto n-dimensional a una curva arbitraria.

Dentro de este paquete se encuentra la interfaz *Distancia*. La implementación de esta interfaz permite utilizar diferentes tipos de métodos de cálculo de la distancia de un punto a una curva. Todos los métodos de cálculo de la distancia tienen que implementar la interfaz D*istancia*, contenido en este paquete. Así, se pueden implementar métodos analíticos, métodos por aproximaciones, métodos para casos concretos (distancia de un punto a una recta en dos dimensiones) o para casos genéricos (distancia de un punto a una curva cualquiera en n dimensiones).

Agregadores. En este paquete se incluye la jerarquía de clases que contiene los diversos operadores de agregación que se irán implementando de entre los existentes en la literatura al efecto, así como otras clases estrechamente relacionadas con su uso, construcción o mantenimiento.

Los operadores de agregación son objetos matemáticos cuya función es reducir un conjunto de números a un único número representativo (o significativo) [4]. Ejemplos de agregadores son la media, el máximo, la suma de cuadrados, el operador OWA o la integral de Choquet.

Dentro de la herramienta se emplean para diversas operaciones, tales como componer el resultado de la función de pertenencia de un punto borroso con respecto a sus componentes o calcular la distancia nítida de un punto borroso a una curva.

La clase principal de este paquete es la interfaz *Agregador*, que se encarga de dar un esqueleto al que se deberán ajustar todos los agregadores, con métodos para agregar valores y obtener su resultado.

La implementación de esta interfaz permite usar una jerarquía de herramientas de agregación que contengan cualquiera de los agregadores formalizados en la literatura al efecto.

Ponderadores. Este paquete está compuesto por clases que se utilizan para valorar los puntos nítidos y borrosos de forma que tengan pesos distintos en la operación de regresión. Este valor de ponderación podrá ser nítido o borroso.

Todos los ponderadores heredan de la clase *Ponderador*, que define los métodos principales que deben tener sus hijos.

Un ejemplo de uso de los ponderadores puede ser dar un valor de compensación a los puntos que representan observaciones atípicas o outlayers en la regresión.

Generadores. En este paquete se incluyen clases que se utilizan para generar puntos. Tendrán cabida en él aquellas clases que generen listas de puntos de acuerdo a unos parámetros suministrados como entrada.

Todos los generadores de puntos implementan la interfaz *GeneradorPuntos*.

Motores. En este paquete estarán incluidos todos los motores de regresión implementados. Un motor es un objeto que, utilizando los restantes elementos de la regresión (puntos, modelos de curva, métodos de cálculo de distancia, agregadores y ponderadores), dirige la regresión para obtener los parámetros para el mejor ajuste del modelo de curva.

La clase principal de este paquete es la clase abstracta *Motor*, de la que heredan todas las implementaciones de los distintos motores de regresión.

Las distintas implementaciones de esta clase abstracta son el núcleo de la herramienta, ya que serán las que guíen el proceso de regresión.

Se puede integrar en la aplicación toda una jerarquía de motores de regresión diferentes, cada uno de los cuales implementará un método de regresión distinto. Así, si se quiere realizar una regresión lineal se podría usar el método analítico clásico implementando un motor al efecto. De la misma manera se podrían añadir otros métodos más complejos de regresión no lineal, como Gauss-Newton, Marquardt o linear descent.

Gestores. Para conseguir una mejora en la explotación de la aplicación, aparte de la modularidad en el diseño, se ha realizado un conjunto de clases llamadas gestores, todas ellas incluidas en este paquete.

Un gestor es un componente gráfico que permite añadir, modificar y mostrar de forma sencilla las distintas implementaciones de cada una de las clases abstractas e interfaces que componen la aplicación. También poseen toda la información necesaria para crear, las instancias de las clases que representan los diferentes elementos que participan en el proceso de regresión.

Todos los gestores heredan de la clase abstracta *Gestor*, que a su vez es descendiente de la clase *JPanel*, por lo que todas ellas pueden representarse en una interfaz gráfica de usuario.

La mantenibilidad que permiten los gestores es uno de los puntos fuertes y reseñables de esta herramienta. Gracias a ella se ve poco afectada por los cambios que puedan producirse al añadir nuevos elementos a la aplicación.

El uso de los gestores y la forma en que están construidos permite generar motores que realicen cálculos en paralelo con distintos modelos y técnicas de regresión a la vez, usando técnicas de programación distribuida.

Por ejemplo, proporcionará a la herramienta la capacidad de realizar varias regresiones con los mismos datos de entrada usando distintos modelos de curva para obtener el modelo con el mejor ajuste para los datos.

4. Pruebas Preliminares

Para realizar una regresión, se procede a seleccionar "Nueva regresión" y a introducir los puntos para el análisis. La herramienta también permite cargar los puntos de un archivo de texto.

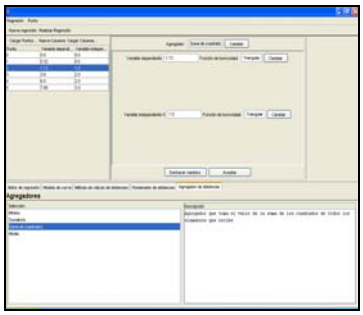


Figura 1. Interfaz de la herramienta

Las observaciones obtenidas al realizar este experimento han sido introducidas en el sistema como números borrosos. Por su simplicidad se han escogido funciones de pertenencia de forma triangular para estos puntos. Así mismo, se ha realizado la configuración de parámetros de la herramienta para el ajuste de la regresión de acuerdo a las opciones que se detallan:

- Motor de regresión recursivo con refinamiento: busca los parámetros para lograr la mejor regresión recorriendo todo el espacio de soluciones que se le indique mediante los parámetros, reduciendo el rango de búsqueda en cada refinamiento.

- Modelo de curva de aprendizaje: Instancia las curvas de tipo y = p1 * x ^ p0

- Cálculo de distancia en y: Mide la diferencia entre el valor de la variable dependiente de cada punto y el valor estimado por la curva para esa variable.

- Ponderador de distancia de producto de un coeficiente por una constante: Realiza el producto por un coeficiente suministrado por el usuario.

- Agregador de distancia de suma de cuadrados: Agregador que toma el valor de la suma de los cuadrados de todos los elementos que recibe.

- Para el motor recursivo usado en el ejemplo, se emplean las siguientes características:
  - Rango :10
  - Coeficiente del rango de búsqueda: 0.5
  - Nº de pasos:10
  - Nº de iteraciones:10
  - Valores iniciales de los parámetros: 0

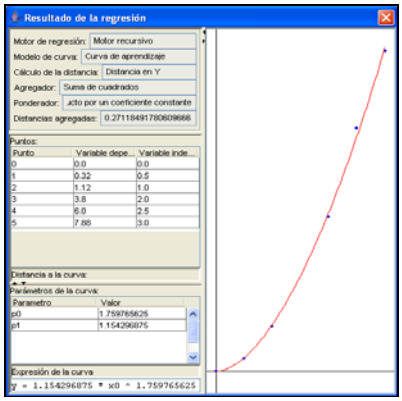


Figura 2. Resultado de la regresión

Como resultado del experimento se muestra en la fig.3 los valores de ajuste de la curva y una representación gráfica de la misma.

5. Conclusiones

De las pruebas preliminares de esta herramienta abierta se pueden emitir las siguientes conclusiones:

- En muchas ocasiones el análisis de regresión borroso produce mejores resultados que el análisis de regresión clásico.
- Sin embargo el uso de agregadores adecuados y específicos para cada entorno de la aplicación es muy importante.
- Esta herramienta está propuesta de forma específica para alcanzar de manera muy simple estas necesidades.

Referencias

[1] Y.O. Chang and B.M. Ayyub. "Fuzzy regression methodos – a comparative asseessment. Fuzzy Sets and Systems", 2001.

[2] S. Conte, H. Dunsmore, and V. Shen. "Software Engineering Metrics and Models". Benjamin Cummins Publishing company, 1986.

[3] Javier Crespo, "Modelo Paramétrico Matemático Difuso para la estimación de Esfuerzo de Desarrollo del Software", tesis doctoral, 2002.

[4] Marcin Detyniecki, "Fundamentals on Aggregation Operators". AGOP 2001

[5] "Familia de productos SPSS®", 2005.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.spss.com/es en FDI/UCM.

[6] "Folleto completo SPSS® Modelos de Regresión 13", 2005.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.spss.com/es en FDI/UCM.

[7] " Fuzzy Regression Analysis ", 2003.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.fuzzy.ru/ en FDI/UCM.

[8] Boris Izyumov, "Aplication of f-regression method to fuzzy classification problem",

[9] B. Izyumov, E. Kalinina, and M. Wagenknecht. "Software tool for regression analysis of fuzzy data". In 9th Zittau Fuzzy Colloquium, Germany 2001.

[10] G. Klir and T. A. Folger. "Fuzzy Sets, Uncertainty, and Information". Prentice Hall, 1988.

[11] Jonathan Lawry, “A Methodology for Computing with words”, International Journal of approximate reasoning, March 2001.

[12] “"MATLAB® 7, The Language of Technical Computing"”, 2004.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.mathworks.com en FDI/UCM.

[13] Nadipuram R. Prased, editor. “Fuzzy Modeling and Control: Selected Works of Sugeno”. CRC Press, 1999.

[14] “R Project for Statistical Computing”, 2005.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.r-project.org en FDI/UCM.

[15] “SAS/STAT® Software Fact Sheet”, 2004.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.sas.com en FDI/UCM.

[16] “STATGRAPHICS® PLUS v.5 para Windows”, 2005.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.statgraphics.net en FDI/UCM.

[17] “STATISTICA® Analytic Modules”, 2005.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.statsoft.com/ en FDI/UCM.

[18] “Statistics Toolbox For Use with MATLAB® :User’s Guide”, 2005.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.mathworks.com en FDI/UCM.

[19] Sugeno, M. “Industrial Aplications of Fuzzy Control”. North-Holland, 1985.

[20] Hideo Tanaka, Satoru Uejima and Kiyoji Asai, “Linear Regression Analysis with Fuzzy Model”, 1982.

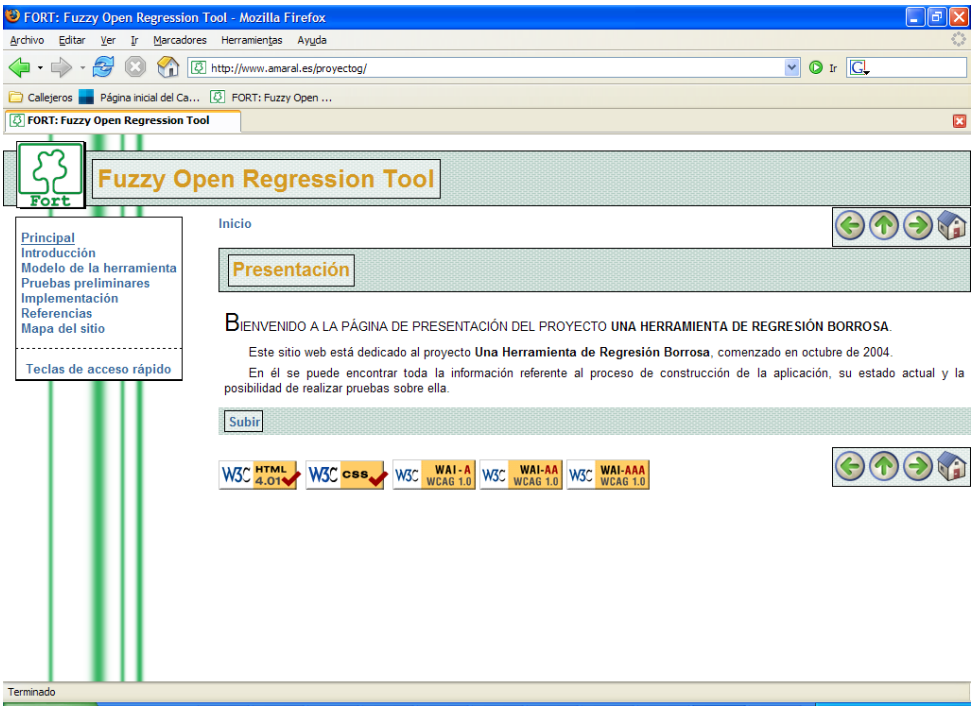
[21] R.C. Tsaur, H.F. Wang, “Outliers in Fuzzy Regression Analysis. In International Journal of Fuzzy Systems”, Vol. 1, N. 2, 1999.

[22] W. N. Venables, D. M. Smith

and the R Development Core Team, “An Introduction to R”, 1990-2004.

[23] Lofti A. Zadeh, “Teoría Computacional de la Percepción”, 2001.

8.4. WebSite de FORT

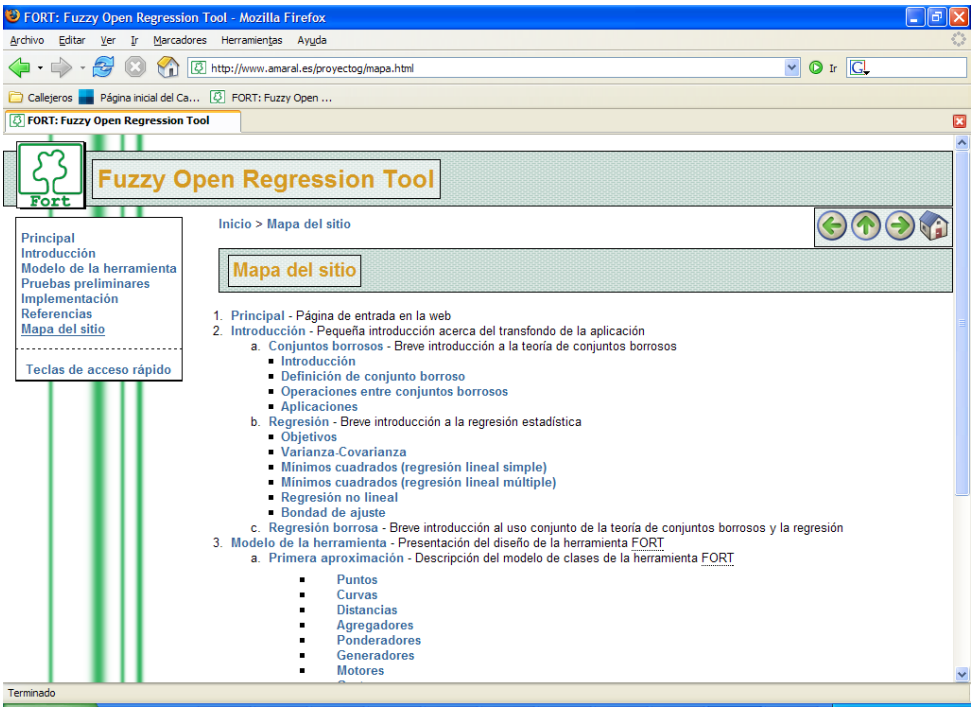


Paralelamente al desarrollo de la aplicación, se ha programado un sitio web con diversos contenidos relacionados con la misma, con el fin de abrir un canal de comunicación con otros miembros de la comunidad científica que estén estudiando temas relacionados.

Este sitio web es de acceso libre y se encuentra en la siguiente dirección:

http://www.fdi.ucm.es/profesor/jcrespo/web/index.html

A continuación se reproduce el índice del mismo para dar una pequeña idea de sus contenidos:



Contenido

Mapa del sitio

1. Principal - Página de entrada en la web
2. Introducción - Pequeña introducción acerca del transfondo de la aplicación
    1. Conjuntos borrosos - Breve introducción a la teoría de conjuntos borrosos
        - Introducción
        - Definición de conjunto borroso
        - Operaciones entre conjuntos borrosos
        - Aplicaciones
    2. Regresión - Breve introducción a la regresión estadística
        - Objetivos
        - Varianza-Covarianza
        - Mínimos cuadrados (regresión lineal simple)
        - Mínimos cuadrados (regresión lineal múltiple)
        - Regresión no lineal
        - Bondad de ajuste
    3. Regresión borrosa - Breve introducción al uso conjunto de la teoría de conjuntos borrosos y la regresión
3. Modelo de la herramienta - Presentación del diseño de la herramienta FORT

1. Primera aproximación - Descripción del modelo de clases de la herramienta FORT
    - Puntos
    - Curvas
    - Distancias
    - Agregadores
    - Ponderadores
    - Generadores
    - Motores
    - Gestores
2. El proceso de regresión - Descripción del proceso de regresión usado en la herramienta

4. Pruebas preliminares - Primeras pruebas realizadas y los resoltados obtenidos
5. Implementación - Acceso a la implementación de la herramienta
    1. Probar la aplicación - Descarga de la aplicación de regresión
    2. Ejecución en línea - Permite utilizar la herramienta en línea mediante un Servlet
6. Referencias - Referencias a otra documentación relacionada con el tema tratado por la herramienta
    1. Enlaces - Enlaces a páginas web relacionadas
    2. Bibliografía - Bibliografía relacionada

7. Mapa del sitio - Mapa de este sitio web

9. Bibliografía

[ACIS03] Asociación Colombiana de Ingeniería Sísmica, “Sistema experto para la toma de decisiones de habilidad y reparabilidad en edificios después de un sismo” Junio 2003.

[AdW95] I. Foster, “Designing and Building Parallel Progrmas”, Addison Wesley, 1995.

[APr93] G. Golub, J. M. Ortega, “Scientic Computing, An Introduction with Parallel Computing”, Academia Press, 1993.

[And00] Jorge de Andrés Sánchez, “Estimación de la estructura temporal de los tipos de interés mediante números borrosos. Aplicación a la valoración financiero-actuarial y análisis de la solvencia del asegurador de vida.”, Noviembre, 2000

[BCF05] Raquel Ballester, Jose I. del Campo, Gonzalo Flórez, “Una Herramienta de Regresión Borrosa”, Mayo, 2005

[Bla05] Paul E. Black, ed., NIST, "Dictionary of Algorithms and Data Structures”, Enero, 2005

http://www.nist.gov/dads/HTML/manhattanDistance.html

http://www.nist.gov/dads/HTML/lmdistance.html

[BMA04] Bioestadística: Métodos y Aplicaciones. U.D. Bioestadística. Facultad de Medicina.Universidad de Málaga.

[Boe81] Barry W. Boehm, “Software Engineering Economics”, 1981

[Calv03] Manuel Hernández Calviño, “Aclarando la Lógica Borrosa (Fuzzy Logic)” Revista Cubana de Física, Vol. 20, Nº. 2, 2003.

[Car01] Omar D. Carmona Arboleda, “Estimación Holística del Riesgo Sísmico Utilizando Sistemas Dinámicos Complejos”, Septiembre, 2001

[Cas02] J. Marcos Castro Bonaño, “Indicadores de Desarrollo Sostenible Urbano. Una Aplicación para Andalucía”, Abril, 2002

[CDS86] S. Conte, H. Dunsmore, and V. Shen. “Software Engineering Metrics and Models”. Benjamin Cummins Publishing company, 1986.

[ChA01] Y.O. Chang and B.M. Ayyub. “Fuzzy regression methodos – a comparative asseessment. Fuzzy Sets and Systems”, 2001.

[Det01] Marcin Detyniecki, “Fundamentals on Aggregation Operators”. AGOP 2001

[DNN93] S. Y. Kung ,"Digital Neural Network", 1993. PTR Prentice Hall, Inc.

[DTC02] T. Díaz Bravo, T. Torres Chávez ,"El EXCEL como apoyo a la enseñanza y la práctica de la Bioestadística.", 2002.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.cecam.sld.cu/pages/rcim/revista_3/articulos_html/articulo_tito.htm en FDI/UCM.

[EE04] Apuntes de Estadística empresarial. Diplomatura en ciencias empresariales. Departamento de economía general y estadística. U.Huelva.

[EST04] StatSoft, Inc. (2004). Electronic Statistics Textbook. Tulsa, OK: StatSoft. WEB: http://www.statsoft.com/textbook/stathome.html.

[FGL05] UNIPHIZ Lab software UNIPHIZ Lab, 2005 "Find Graph Quick and Easy

Graphing, digitizing and curve fitting software".Extraído en Junio de 2005 con navegador MS Internet Explorer de http://www.uniphiz.com/findgraph.htm

[FRA03] " Fuzzy Regression Analysis ", 2003.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.fuzzy.ru/ en FDI/UCM.

[Gal05] José Galindo, "Curso Introductorio de Conjutos y Sistemas Difusos", extraido en Junio de 2005 con navegador Opera 7 de http://www.lcc.uma.es/~ppgg/FSS/

[GSP05] GraphPad Software, Prism 4 for Windows and Macintosh 2005. Extraído en Junio de 2005 con navegador MS Internet Explorer de http://www.graphpad.com/prism

[IKW01] B. Izyumov, E. Kalinina, and M. Wagenknecht. "Software tool for regression analysis of fuzzy data". In 9th Zittau Fuzzy Colloquium, Germany 2001.

[IPE03] ”Informacion de Producto de Excel® 2003”, 2003.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.microsoft.com/spain/office/products/excel en FDI/UCM.

[Izy] Boris Izyumov, “Aplication of f-regression method to fuzzy classification problem”,

[JCr02] Javier Crespo, “Modelo Paramétrico Matemático Difuso para la estimación de Esfuerzo de Desarrollo del Software”, tesis doctoral, 2002.

[Jlaw01] Jonathan Lawry, “A Methodology for Computing with words”, International Journal of approximate reasoning, March 2001.

[KlF88] G. Klir and T. A. Folger. “Fuzzy Sets, Uncertainty, and Information”. Prentice Hall, 1988.

[MAT04] “"MATLAB® 7, The Language of Technical Computing"”, 2004.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.mathworks.com en FDI/UCM.

[MES03] "Modelos Estadísticos Aplicados", Juan M. Vilar Fernández,2003.Publicaciones de la UDC, monografía 101.

[MNT04] METHODS FOR NON-LINEAR LEAST SQUARES PROBLEMS 2nd Edition, April 2004

K. Madsen, H.B. Nielsen, O. Tingleff Informatics and Mathematical Modelling Technical University of Denmark

[MRS05] “Folleto completo SPSS® Modelos de Regresión 13”, 2005.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.spss.com/es en FDI/UCM.

[NPr99] Nadipuram R. Prased, editor. “Fuzzy Modeling and Control: Selected Works of Sugeno”. CRC Press, 1999.

[PID05] “Procesamiento de Imágenes Digitales. Preliminares Topológicos”, 2005.

Extraído en Junio de 2005 con navegador Opera 7 de http://www.us.es/gtocoma/pid/pid2/pid21.htm en FDI/UCM.

[PNJ01] “Procedimientos Numéricos en Lenguaje Java”. Angel Franco García. Escuela Universitaria de Ingeniería Técnica Industrial.2001

[PRE93] Pressman, Roger S., “Understanding Software Engineering Practices: Required at SEI Level 2 Process Maturity,” Software Engineering Training Series, Software Engineering Process Group, July 30, 1993

[PSer96]Pérez Serrada, Anselmo, “Una introducción a la Computación Evolutiva” Marzo, 1996.

[RLM00] Técnicas de regresión: Regresión Lineal Múltiple. Pértega Díaz, S., Pita Fernández, S.,Unidad de Epidemiología Clínica y Bioestadística. Complexo Hospitalario Juan Canalejo. A Coruña (España). CAD ATEN PRIMARIA 2000; 7: 173-176.

[RNA00] José R. Hilera y Victor J Martinez, "Redes Neuronales Artificiales”, 2000. Alfaomega. Madrid. España

[RPS05] “The R Project for Statistical Computing”, 2005.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.r-project.org en FDI/UCM.

[RSA95] de Carlos Coello Coello, Carlos Zozaya Gorostiza y David E. Goldberg “Revista Soluciones Avanzadas”, No. 17, enero.

[SAM05] “STATISTICA® Analytic Modules”, 2005.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.statsoft.com/ en FDI/UCM.

[SAS04] “SAS/STAT® Software Fact Sheet”, 2004.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.sas.com en FDI/UCM.

[SME99] Philippe SMETS ,"Varieties of ignorance and the need for well-founded theories", IRIDIA-Université Libre de Bruxelles

[SNH04] Sujit Nath Pant, Keith E. Holbert, “Fuzzy Logic in Decision Making and Signal Processing”, 2004, extraído en Junio de 2005 con navegador Opera 7 de http://ceaspub.eas.asu.edu/powerzone/

[SPS05] “Familia de productos SPSS®”, 2005.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.spss.com/es en FDI/UCM.

[SSi04] “SimStat v2.5 Simulation & Statistical Software”, 2004.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.simstat.com en FDI/UCM.

[STP05] “STATGRAPHICS® PLUS v.5 para Windows”, 2005.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.statgraphics.net en FDI/UCM.

[STU05] “Statistics Toolbox For Use with MATLAB® :User’s Guide”, 2005.

Extraído en Mayo de 2005 con navegador MS Internet Explorer de http://www.mathworks.com en FDI/UCM.

[Sug85] Sugeno, M. “Industrial Aplications of Fuzzy Control”. North-Holland, 1985.

[TJR95] T. J. Ross, “Fuzzy Logic with Engineering Applications”, McGraw-Hill, Inc, 1995.

[TSK82] Hideo Tanaka, Satoru Uejima and Kiyoji Asai, “Linear Regression Analysis with Fuzzy Model”, 1982.

[Twa99] R.C. Tsaur, H.F. Wang, “Outliers in Fuzzy Regression Analysis. In International Journal of Fuzzy Systems”, Vol. 1, N. 2, 1999.

[VSR04] W. N. Venables, D. M. Smith and the R Development Core Team, “An Introduction to R”, 1990-2004.

[Zad01] Lofti A. Zadeh, “Teoría Computacional de la Percepción”, 2001

Soft Computing

Las técnicas de Soft Computing engloban básicamente, la lógica borrosa, las redes neuronales, la computación evolutiva, los algoritmos genéticos y el razonamiento probabilístico.

El Soft Computing desempeña un papel muy importante en las ciencias y en la ingeniería, aunque su aplicación se extenderá en otros muchos campos, debido a los resultados satisfactorios que se han ido obteniendo con su uso,

Las técnicas de Soft Computing destacan por su tolerancia a la imprecisión, la incertidumbre, grado de credibilidad y la aproximación.

Estas técnicas usan la mente humana como modelo. Las líneas de investigación que lleva a cabo son:

Una nueva generación de motores de búsqueda en Internet, que usan técnicas de Soft Computing y tratan de mejorar la búsqueda lexicográfica actual, usando una búsqueda conceptual.

Técnicas avanzadas para descubrir “perfiles de usuario” que permitan un uso de internet más inteligente “a la carta”.

Comercio electrónico basado en técnicas de Soft Computing, por ejemplo, lo que el profesor Mandani denomina Soft Knowledge.

Semantic Web.

En los últimos años se ha podido comprobar un rápido crecimiento de las aplicaciones de la lógica borrosa y las redes neuronales, en diversos campos: electrónica de consumo, control de procesos industriales, reconocimiento del habla, visión artificial, tratamiento de la señal, reconocer y clasificar imágenes, manejar vehículos en trafico denso y un largo etcétera.

Lógica borrosa y sus aplicaciones

La lógica borrosa es básicamente una lógica multievaluada que permite valores intermedios para poder definir evaluaciones convencionales como sí / no, verdadero / falso, negro / blanco, etc. De esta forma se ha realizado un intento de aplicar una forma más humana de pensar en la programación de computadoras. La lógica borrosa se inició en 1965 por Lotfi A. Zadeh, profesor de ciencia de computadoras en la Universidad de California en Berkeley.

Aunque la lógica borrosa se inventó en Estados Unidos el crecimiento rápido de esta tecnología ha comenzado desde Japón y ahora nuevamente ha alcanzado USA y también Europa.

La intención original del profesor Zadeh era crear un formalismo para manipular de forma más eficiente la imprecisión y vaguedad del razonamiento humano expresado

lingüísticamente, pero el éxito de la lógica borrosa llegó en el campo del control automático de procesos. Esto se debió principalmente al “boom” de lo borroso en Japón, iniciado en 1987 y que alcanzó su máximo apogeo a principios de los noventa.

Desde entonces, han sido infinidad los productos lanzados al mercado que usan tecnología borrosa, muchos de ellos utilizando la etiqueta “fuzzy” como símbolo de calidad y prestaciones avanzadas.

En 1974 el profesor Mamdani experimentó con éxito un controlador borroso en una máquina de vapor, pero la primera implantación real de un controlador de este tipo fue realizada en 1980 por F. L. Smidth & Co. en una planta cementera en Dinamarca.

En1983, Fuji aplica lógica borrosa para el control de inyección química para plantas depuradoras de agua, por primera vez en Japón.

En 1987 la empresa OMRON desarrolla los primeros controladores borrosos comerciales con el profesor Yamakawa. A partir de ese momento, el control borroso ha sido aplicado con éxito en muy diversas ramas tecnológicas, por ejemplo la metalurgia, los robots de fabricación, controles de maniobra de aviones, ascensores o trenes (tren-metro de Sendai, Japón, 1987), sensores, imagen y sonido (sistema de estabilización de imagen en cámaras fotográficas y de video Sony, Sanyo, Cannon...), electrodomésticos (lavadoras de Panasonic o Bosch, aire acondicionado Mitsubishi, rice-cooker...), automoción (sistemas de ABS de Mazda o Nissan, Cambioautomático de Renault, controlautomático de velocidad, climatizadores...) y una larga lista de aplicaciones comerciales.

Redes neuronales y sus aplicaciones

Las redes neuronales surgen de los estudios sobre neurofisiología debidos a Rosenblatt. En estos trabajos se busca un modelo matemático para la neurona, de manera que sea posible reproducir por técnicas artificiales la capacidad de interpretación de información del cerebro.

Se considera el cerebro como una computadora capaz de procesar información imprecisa a un ritmo increíblemente veloz, y sobre todo, que aprende sin instrucciones explícitas de ninguna clase a crear las representaciones internas que hacen posible tales habilidades.

La estructura de la red se establece por imitación de las estructuras neuronales, tales como el cerebro. Estas redes deben tener capacidad de aprendizaje, es decir, deben ser capaces de modificar sus conexiones con tal de adaptarse de la mejor forma posible al comportamiento requerido.

En una red neuronal se distinguen dos etapas: una primera de aprendizaje, en la cual son presentadas a la red un conjunto de patrones de entrada y de salida. Por medio de algún algoritmo de optimización se modifican sus conexiones con el fin de que ésta imite este comportamiento. Y una segunda etapa, de funcionamiento, en la cual, ante cualquier entrada, la red debe ser capaz de responder con una salida lo más similar posible a las aprendidas [RNA00].

En general, la aplicación más extendida de las redes neuronales es la clasificación, también conocida como reconocimiento de patrones. La red es

empleada como un clasificador, de manera que ante una entrada estima la mayor o menor afinidad de ésta a los patrones aprendidos [DNN93].

Actualmente, las redes neuronales se emplean con éxito en numerosas aplicaciones, tales como el reconocimiento de caracteres escritos, reconocimiento del habla y sistemas de identificación.

Las redes neuronales y la lógica borrosa pueden aportar soluciones muy favorables al proceso de traducción Automática, ya que el problema principal que se presenta a la hora de realizar la traducción de un texto es su adecuada comprensión e interpretación.

Algoritmos genéticos y sus aplicaciones

Un algoritmo genético es una técnica de programación que imita a la evolución biológica como estrategia para resolver problemas.

Dado un problema específico a resolver, la entrada del algoritmo genético es un conjunto de soluciones potenciales a ese problema, codificadas de alguna manera, y una métrica llamada función de aptitud que permite evaluar cuantitativamente a cada candidata. Estas candidatas pueden ser soluciones que ya se sabe que funcionan, con el objetivo de que el algoritmo genético las mejore, pero se suelen generar aleatoriamente.

Seguidamente, el algoritmo genético evalúa cada candidata de acuerdo con la función de aptitud. En un acervo de candidatas generadas aleatoriamente, por supuesto, la mayoría no funcionarán en absoluto, y serán eliminadas. Sin embargo, por puro azar, unas pocas pueden ser prometedoras -pueden mostrar actividad, aunque sólo sea actividad débil e imperfecta, hacia la solución del problema.

Estas candidatas prometedoras se conservan y se les permite reproducirse. Se realizan múltiples copias de ellas, pero las copias no son perfectas; se introducen cambios aleatorios durante el proceso de copia.

Esta descendencia digital prosigue con la siguiente generación, formando un nuevo acervo de soluciones candidatas, y son sometidas a una ronda de evaluación de aptitud. Las candidatas que han empeorado o no han mejorado con los cambios en su código son eliminadas de nuevo; pero, de nuevo, por puro azar, las variaciones aleatorias introducidas en la población pueden haber mejorado a algunos individuos, convirtiéndolos en mejores soluciones del problema, más completas o más eficientes.

De nuevo, se seleccionan y copian estos individuos vencedores hacia la siguiente generación con cambios aleatorios, y el proceso se repite. Las expectativas son que la aptitud media de la población se incrementará en cada ronda y, por tanto, repitiendo este proceso cientos o miles de rondas, pueden descubrirse soluciones muy buenas del problema [PSer96].

Los algoritmos genéticos han demostrado ser una estrategia enormemente poderosa y exitosa para resolver problemas, demostrando de manera espectacular el poder de los principios evolutivos.

Se han utilizado algoritmos genéticos en una amplia variedad de campos para desarrollar soluciones a problemas tan difíciles o más difíciles que los abordados por los diseñadores humanos [RSA95].

Las soluciones que consiguen los algoritmos genéticos son a menudo más eficientes, más elegantes o más complejas que nada que un ingeniero humano produciría.

Existen diversos tipos algoritmos evolutivos que se pueden aplicar a numerosos problemas, dependiendo de su naturaleza, ya sea un problema de búsqueda, de optimización, de predicción o de clasificación (Parametrización, Configuración, Maximización, Minimización).

La Computación Evolutiva puede aplicarse en diversos dominios, por ejemplo, en un proceso de generación de conocimiento interpretable por el humano.

Actualmente, la generación automática de conocimiento es un proceso realizable en una computadora con un mínimo de supervisión, retroalimentación y supuestos.

Existen aplicaciones prácticas donde este proceso automático ha sido de gran ayuda, como en el campo del control de plantas depuradoras de agua, en donde se ha logrado revisar las clases de situaciones de una planta propuestas por un experto humano, recordándole situaciones pasadas que se le olvidó mencionar en una entrevista.