A methodology for building fuzzy rules from data

J. Tinguaro Rodríguez
Faculty of Mathematics, Complutense University of Madrid, Plaza de Ciencias 3 28040 Madrid, Spain
jtrodrig@mat.ucm.es
Victoria López
Faculty of Informatics, Complutense University of Madrid, C/ Profesor José García Santesmases, s/n 28040 Madrid, Spain
vlopez@fdi.ucm.es
Javier Montero
Faculty of Mathematics, Complutense University of Madrid, Plaza de Ciencias 3 28040 Madrid, Spain
javier_montero@mat.ucm.es
Begoña Vitoriano
Faculty of Mathematics, Complutense University of Madrid, Plaza de Ciencias 3 28040 Madrid, Spain
bvitoriano@mat.ucm.es

Abstract

Extraction of rules for classification and decision tasks from databases is an issue of growing importance as automated processes based on data are being required in these fields. Interpretability of rules is improved by defining classes for independent variables. Moreover, though more complex, a more realistic and flexible framework is attained when fuzzy classes are considered. In this paper, an inductive approach is taken in order to develop a general methodology for building fuzzy rules from databases. Three types of rules are built in order to be able of dealing with both categorical and numerical data.

Keywords: Rules induction, Fuzzy classification, Data Mining, Decision Support Systems

1 Introduction

Fuzzy rules and algorithms are widely used in several real-life applications of computational intelligence, as targeted e-commerce marketing and advertising [9], natural disaster and emergency management (see for instance [3] and [6]) or control [4]. In many of these applications, decision and classification rules are obtained by means of experts and the subsequent knowledge engineering.

Nevertheless, in many cases, the necessary information to build these rules is contained in databases rather than in experts' heads. Moreover, as the underlying realities of these applications are evolving, it is also necessary to undergo a continuous learning process in order to adapt to these changing situations. Last but not least, in many fields this learning process is needed to be automated.

For all these reasons, procedures are needed to extract and build a set of fuzzy rules from raw data contained in databases. In this paper, we propose a general methodology to do so, based upon an inductive approach in which rules are conceived as successively experienced relations among variables.

This paper is organized as follows: first, we describe the model of knowledge representation, which takes a database as input and produces, by means of a set of classes (crisp or fuzzy), a matrix representing the data in a categorical way. Next, we describe how to build up three types of inference and classification rules, using that matrix along with the raw data as inputs. Finally, some remarks and conclusions are exposed with relation to certain characteristics of the exposed methodology and its applicability.

2 Knowledge representation

In order to build up rules from data and carry out a useful inference process, it is necessary to previously define the general framework and mathematical models that are used to represent the information and knowledge we are going to work with. In other words, a mathematical model of knowledge representation is needed to give the data an appropriate shape or structure, in agreement with those required for the input of the rules building process.

Basic raw data is intended to be a database, which could be viewed as a real-valued matrix $D = \left(d_{ki} \right)_{mxn}$, having m instances and n variables $X_1,..,X_n$.

The range of each variable X_i is then partitioned into a set of c_i classes $A_{i1},...,A_{ic_i}$, which can be fuzzy or crisp. In this paper, these classes are intended to be linearly ordered, i.e. $A_{ij} < A_{ij'}$ iff $j < j'$, but a different structure could be given as explained in [5].

We will use capital letters to denote the values of variables in the database, this is, $X_{ki} = d_{ki}$ for $k = 1,..,m$ and $i = 1,..,n$. Lower case letters will denote values of categories, i.e., $x^k_{ij_i} = \mu_{A_{ij}}(X_{ki})$ for $j_i = 1,...,c_i$ and $\mu_{A_{ij}}$ being the membership function of the class A_{ij}. In the crisp case, it is supposed that the value of X_{ki} lies in exactly one class j', i.e., $\mu_{A_{ij'}}(X_{ki}) = 1$ and $\mu_{A_{ij}}(X_{ki}) = 0$ if $j \neq j'$. In the fuzzy case, $\mu_{A_{ij}}(X_{ki}) \in [0,1]$ and the classes not necessarily form a fuzzy partition in the sense of Ruspini [7], i.e., $\sum_{j=1}^{c_i} \mu_{A_{ij}}(X_{ki})$ need not to sum exactly 1 (see [1]). In fact, missing values of any variable are modelized assigning the value 0 to every class.

In this way, first level of knowledge representation is constituted by a matrix $H = \left(h_{kj} \right)_{mxl}$, $l = \sum_{i=1}^{n} c_i$ being the total number of categories or classes and such that $h_{kj} = x^k_{ij_i} = \mu_{A_{ij}}(X_{ki}) = \mu_{A_{ij}}(d_{ki})$, for all $k = 1,..,m$, $i = 1,..,n$, $j_i = 1,...,c_i$ and $j = 1,..,l$. Reference to the i-th variable is removed in the h_{kj}'s as it is intended that categories are sorted by the variables to which they correspond.

3 Data-based rules building

Matrix H constitutes the first level of knowledge. However, in order to have some inference capability, a second level knowledge, or meta-knowledge, is needed. This second level knowledge, to which we will refer as *rules*, has to be extracted from the first one, and therefore this is the reason for we say that these rules are *data-based*.

Conceptually, the methodology for rule extraction described in this paper is based upon the idea that a rule is built up through the successive repetition and experience of similar situations. It is usually accepted that whenever a relation is experienced or successively repeated, its rule condition is strengthened.

The approach presented in this paper follows these ideas. Each instance of the database in which the same classes of different variables appear together is considered as a case for the existence of a relationship between these categories. In this sense, what is going to be measured and translated into the rules is the trend of some variables as other variables appear. Another methodology for building interpretable fuzzy rules from data is the one described in [2].

Rules need some variables to play the role of *premises* or independent variables, being the rest called *consequences* or dependent variables. Thus, from the set of n variables $X_1,..,X_n$, a subset of p premises variables is extracted, which left us with another subset of $q = n - p$ consequence variables. Since in this approach the conclusion for each consequence variable is independent of the conclusion for the rest of them, for the sake of simplicity in the exposition we will suppose without loss of generality that $q = 1$, i.e., that there exist only one consequence or dependent variable. In the subsequent, this one will be denoted by *Y*, being $\left\{ X_1,..,X_p \right\}$ the set of premises or independent variables.

In this paper, three types of rules and the algorithms to compute them are described.

Formally, for a rule we understand an expression of the type

$$R: \ if\ X_1 \text{ is } A_1 \text{ and ... and } X_p \text{ is } A_p$$
$$then\ Y \text{ is } B,$$

where each A_i is a class of the *i-th* premise variable and *B* is the conclusion assigned to the dependent variable *Y*. Thus, what is understood for three different groups of rules is that three different types of conclusions *B* are going to be assigned to the dependent variable *Y*:

- In the first type of rules, a degree of possibility π_j is assigned to each one of the $d := c_n = c_{p+1}$ classes B_j $(j = 1,..,d)$ defined in last section for the variable *Y*. Therefore, $B = \pi = (\pi_1,..,\pi_d)$. This group of rules is useful to deal with categorical variables and also with numerical variables previously classified in classes.
- The second group of rules assigns to *Y* a mean value $\overline{y}$ in the range of the dependent variable. The algorithm that computes this value makes use of the possibilities π of the previous group in order to weight the values of *Y*. Therefore, these rules are dependent of the rules in the last group, and in this case $B = \overline{y}$. This group of rules works with numerical data, and therefore is devised for the task of predicting numerical variables.
- Finally, the last group of rules assigns to *Y* an interval $[b_1, b_2]$ of possible values of the dependent variable. The lower and upper extremes of this interval are computed by means of fuzzy (resp. crisp) order statistics, and therefore the algorithm to compute these rules is in fact an algorithm to compute those statistics. Thus, for this group of rules $B = [b_1, b_2]$. Intervals and order statistics fit well when working with numerical variables which have a huge variability and/or present outliers.

Case 1 $B = \pi$. Calculation of dependent classes possibilities π

Given the matrix *H*, a class B_j of the dependent variable *Y* and a combination $(j_1,...,j_p)$ of classes of the *p* premises variables, $j \in \{1,..d\}$ and $j_i \in \{1,..,c_i\}$ for all $i = 1,..,p$, we define the possibility in *H* of the class B_j when $A_{1j_1} \wedge ... \wedge A_{pj_p}$ is true as a weighted aggregation of its membership degrees through all the *m* instances of the database, this is,

$$\pi_H(j \mid j_1,...,j_p) = \frac{\sum_{k=1}^{m} T(x^k_{1j_1},..,x^k_{pj_p})\, y^k_j}{\sum_{k=1}^{m} T(x^k_{1j_1},..,x^k_{pj_p})},$$

where $y^k_j = \mu_{B_j}(Y^k)$ is the membership degree to its *j*-th class of the *k*-th instance of variable *Y* and *T* is the logical operator (usually a t-norm) that modelizes the conjunction "and".

Proposition 1.- If variable *Y* does not have missing values and the classes B_j, $j = 1,..d$, form a Ruspini partition, then the possibilities of the *d* classes of *Y* given any combination of premises $(j_1,...,j_p)$ sum up to one.

Proof:

$$\sum_{j=1}^{d} \pi_H(j \mid j_1,...,j_p) = \frac{\sum_{k=1}^{m} T(x^k_{1j_1},..,x^k_{pj_p}) \sum_{j=1}^{d} y^k_j}{\sum_{k=1}^{m} T(x^k_{1j_1},..,x^k_{pj_p})} =$$

$$= \frac{\sum_{k=1}^{m} T(x^k_{1j_1},..,x^k_{pj_p})}{\sum_{k=1}^{m} T(x^k_{1j_1},..,x^k_{pj_p})} = 1$$

Thus, if a variable *Y* has missing values and its classes form a Ruspini partition, last proposition is saying that possibilities of *Y* sum less than one. In this way, we can define the *ignorance* associated with variable *Y* given a combination of premises $(j_1,...,j_p)$ as one minus the sum of its possibilities for that combination of premises, i.e., $I_Y(j_1,...,j_p) = 1 - \sum_{j=1}^{d} \pi_H(j \mid j_1,...,j_p)$. As explained in [5], *ignorance I* is a necessary class that should be added to the current ones in order to better modelize the underlying learning process.

Case 2 $B = \bar{y}$. Calculation of dependent variable mean $\bar{y}$.

Given H and a combination $\left(j_1,\dots,j_p \right)$ of classes of the p premises variables, $j_i \in \left\{ 1,..,c_i \right\}$ for all $i = 1,..,p$, the fuzzy mean of a crisp variable Y when $A_{1j_1} \wedge \dots \wedge A_{pj_p}$ is true could be easily defined as $\sum_{k=1}^{m} T(x_{1j_1}^k,..,x_{pj_p}^k)Y^k / \sum_{k=1}^{m} T(x_{1j_1}^k,..,x_{pj_p}^k)$. However, as variable Y could have not only missing values but also outliers lying in classes with low possibility, it seemed more realistic to us to define this mean as

$$\bar{y}_H(j_1,\dots,j_p) = \frac{\sum_{k=1}^{m} W^k(j_1,\dots,j_p)T(x_{1j_1}^k,..,x_{pj_p}^k)Y^k}{\sum_{k=1}^{m} W^k(j_1,\dots,j_p)T(x_{1j_1}^k,..,x_{pj_p}^k)}$$

where the weights are defined as $W_H^k(j_1,\dots,j_p) = \sum_{j=1}^{d} y_j^k \pi_H(j \mid j_1,\dots,j_p)$. This way, missing values of Y do not affect the computation. Moreover, if outliers of Y lie in classes with low possibility or possibility equal to 0, their values will not affect so much the resulting mean. Anyway, this mean $\bar{y}_H(j_1,\dots,j_p)$ is biased towards the classes B_j with higher possibility $\pi_H(j \mid j_1,\dots,j_p)$, which is a realistic assumption in our opinion.

Case 3 $B = \left[b_1, b_2 \right]$. Calculation of fuzzy order statistics.

Given H and a combination $\left(j_1,\dots,j_p \right)$ of classes of the p premises variables, $j_i \in \left\{ 1,..,c_i \right\}$ for all $i = 1,..,p$, if the classes of premises variables are fuzzy it is not obvious how to calculate the percentile $\alpha \in \left\{ 1,..,99 \right\}$ of the values of a crisp variable Y for which $A_{1j_1} \wedge \dots \wedge A_{pj_p}$ is true. For each instance of Y_k in H, the truth value of the conjunction $A_{1j_1} \wedge \dots \wedge A_{pj_p}$ could take a different value $T(x_{1j_1}^k,..,x_{pj_p}^k)$, so we can not simply take as the α -percentile the value of Y below which α percent of the observations for which $A_{1j_1} \wedge \dots \wedge A_{pj_p}$ is true may be found. However, it seems natural to generalize this idea to the fuzzy case by defining the α -percentile as the value of Y below which we can found the α percent of the total amount of membership $w(j_1,\dots,j_p) = \sum_{k=1}^{m} T(x_{1j_1}^k,..,x_{pj_p}^k)$ to the conjunction class $A_{1j_1} \wedge \dots \wedge A_{pj_p}$ .The algorithm used to find this value is the following:

1. Sort H by the values of Y, removing the instances for which the value of Y is missing.
2. Define $$k_\alpha(j_1,\dots,j_p) = \min\left\{ k / \sum_{s=1}^{k} T(x_{1j_1}^s,..,x_{pj_p}^s) > \frac{\alpha}{100} w(j_1,\dots,j_p) \right\}$$
3. Define the α -percentile as $PC_\alpha(j_1,\dots,j_p) = Y_{k_\alpha(j_1,\dots,j_p)}$.

Thus, in order to build the interval that constitutes the conclusion of the rules of this third group, two values $\alpha_1, \alpha_2 \in \left\{ 1,..,99 \right\}$, $\alpha_1 < \alpha_2$ have to be chosen, leading to the interval $\left[PC_{\alpha_1}(j_1,\dots,j_p), PC_{\alpha_1}(j_1,\dots,j_p) \right]$.

These three groups of rules are then stored as vectors or multi-dimensional matrices, which constitute a second level of knowledge representation.

4 Conclusion

For each combination $(j_1,\dots,j_p)$ of the premises variables, the importance or influence over a rule $R(j_1,\dots,j_p)$ of a given instance *k* in the dataset is directly proportional to $T(x_{1j_1}^k,..,x_{pj_p}^k)$. This is obvious for the two first types of rules described above. For the third ones, note that an instance with $T(x_{1j_1}^k,..,x_{pj_p}^k) = 0$ have no influence over the α -percentile computation. On the other hand, if $T(x_{1j_1}^k,..,x_{pj_p}^k)$ is close to 1 then removing the instance *k* could have a quite important effect over the percentile, especially when the values of $T(x_{1j_1}^{k'},..,x_{pj_p}^{k'})$, $k \neq k'$, are small.

Therefore, these rules measure the behaviour of a dependent variable for each combination $(j_1,...,j_p)$ of the premises, giving more importance or weight to those instances in the dataset for which the values $X_1,..,X_p$ of premises lie inside the conjunction class $A_{1j_1} \wedge ... \wedge A_{pj_p}$. That behaviour is then measured by averaging membership degrees of the dependent classes (case 1), numerical values of the variables (case 2) or by order statistics (case 3). It has to be pointed that no one of these operations measure or give importance to the prevalence of any specific combination of premises classes, i.e., conjunction classes with higher frequency does not produce stronger rules, although logically they lead to more stable and sound ones.

Another important remark concerns the number of rules to create. Many times, we have a previously defined set of rules to which give attention, as it is the case when analysis and considerations previously carried out by experts make possible knowing those premises which allows to give the best prediction over the decision variables. In these cases, the methodology exposed in this paper can be seen as a mechanism to compute the conclusions *B* of that set of rules from data containing the appropriate information.

On the other hand, when confronted with the problem of building rules from data it is of course possible that the set of rules to create is not previously defined. This is the same to say that we are not given the set of combinations of independent variables and/or classes of these variables that have to be used as premises of the rules. Furthermore, every possible combination of classes could occur in the practice and constitute in fact an important premise for explaining the data. For this reason, we give here algorithms to build every possible rule, this is, a rule for each possible combination of classes of the premises variables. In fact, creating all possible rules could be a successful strategy for some practical developments (see for example [6]) in which the number of premises *p* is relatively small.

If *p* is large, making use of statistical methods as clustering or principal components should be taken into account in order to devise a suitable set of rules or for dimensionality reduction purposes. Future research will concerns these and others issues related with rules building and continuous learning processes. Introduction of a bipolar approach [8] to allow detection of contradictory information will be explored. A more advanced prototype of the decision support system [6] is also being developed making use of the exposed methodology.

Acknowledgements

This research has been partially supported by grants TIN2006-06190, CCGO7-UCM/ESP-2576 and I-Math Consolider C3-0132.

References

[1] A. Amo, J. Montero, G. Biging, V. Cutello (2004). Fuzzy classification systems *European Journal of Operational Research* 156 (2): 495-507

[2] S. Destercke, S. Guillaume, B. Charnomordic (2007). Building an interpretable fuzzy rule base from data using Orthogonal Least Squares Application to a depollution problem, *Fuzzy Sets and Systems*, 158 (18): 2078-2094

[3] L.S. Iliadis (2005). A decision support system applying an integrated fuzzy model for long-term forest fire risk estimation, *Environmental Modelling & Software*, 20 613-621

[4] E.H. Mamdani (1974). Application of Fuzzy Algorithms for the Control of a Dynamic Plant, *Proc. IEE*, 121 (12): 1585-1588

[5] J. Montero, D. Gómez, H. Bustince (2007). On the relevance of some families of fuzzy sets, *Fuzzy sets and systems*, 158 (22): 2439-2442

[6] J.T. Rodriguez, B. Vitoriano, J. Montero, A. Omaña (2008). A decision support tool for humanitarian organizations in natural disaster relief, in: D. Ruan et al. (eds), *Computational Intelligence in Decision and Control*. World Scientific, Singapore: p. 600-605

[7] E.H. Ruspini (1969). A new approach to clustering, *Inform. Control* 15 22–32.

[8] M. Öztürk, A. Tsoukiàs (2007). Modelling uncertain positive and negative reasons in decision aiding, *Decision Support Systems*, 43 (4): 1512-1526

[9] R.R. Yager (2000). Targeted e-commerce marketing using fuzzy intelligent agents *Intelligent Systems and their Applications*, 15 (6): 42-45