Available online at www.sciencedirect.com
SCIENCE DIRECT®
Fuzzy Sets and Systems 149 (2005) 105–129
FUZZY sets and systems
www.elsevier.com/locate/fss

A definition for fuzzy approximate dependencies

F. Berzal[a,*], I. Blanco[b], D. Sánchez[a], J.M. Serrano[c], M.A. Vila[a]

[a]*Department of Computer Science A.I., University of Granada, 18071 Granada, Spain*
[b]*Department of Languages and Computer Science, University of Almeria, 04120 Almeria, Spain*
[c]*Department of Computer Science, University of Jaen, 23071 Jaen, Spain*

Available online 23 August 2004

Abstract

In the analysis of data stored in databases, a very interesting issue is the detection of possible existing relations between attribute values and, at an upper level, relations between attributes themselves. In case uncertainty is present in data, or it is introduced in a pre-processing step, specific data mining and knowledge discovery techniques and methodologies must be provided. The theory of fuzzy subsets is a helpful tool to reach this goal. In this paper we introduce a new definition and an algorithm for computing fuzzy approximate dependencies, a type of relations that can be found between attributes in a fuzzy database, on the basis of a previous definition of fuzzy association rule. We will discuss about possible applications of this new tool.

Keywords: Fuzzy relational databases; Fuzzy association rules; Fuzzy approximate dependencies; Data mining

1. Introduction

Knowledge discovery in databases (KDD) is concerned with finding previously unknown and potentially useful knowledge from databases. Roughly, it consists of three main steps: data preprocessing (preparing data), data mining (finding interesting patterns in data) and interpretation of data mining results to provide the final knowledge.

There are several ways in which fuzzy set technology is useful in KDD, see [43]. First, participation of users in all the steps of the KDD process is crucial, and in particular the ultimate objective of KDD is to

* Corresponding author. Tel.: +34-958-240599; fax: +34-958-243317.
E-mail addresses: fberzal@decsai.ugr.es (F. Berzal), iblanco@ual.es (I. Blanco), daniel@decsai.ugr.es (D. Sánchez), jschica@ujaen.es (J.M. Serrano), vila@decsai.ugr.es (M.A. Vila).

0165-0114/$ - see front matter © 2004 Published by Elsevier B.V.
doi:10.1016/j.fss.2004.07.012

provide users with *understandable* information. Fuzzy set technology is a suitable tool for this purpose, specially in those cases where we want to express summaries or relations between numerical data by means of linguistic terms. Also, linguistic assessments of patterns are helpful in order to judge them in the last step of KDD.

However, there is another important fact that links KDD and fuzzy sets: in many cases data is inherently imprecise or uncertain, and several fuzzy relational, deductive and object-oriented database models have been developed in order to cope with this. A more usual case, that provides a similar scenario, is that of fuzzy data obtained from crisp data in the preprocessing step by aggregation, summarization or change of granularity level. The analysis of such information requires the development of specific tools as fuzzy extensions of existing ones.

To illustrate these claims, let us use a simple example. Let *Cat* and *Sal* be two attributes storing the job category and salary of a set of employees. Suppose the category takes values in a set of labels indicating the kind of work, for example {Manager, Commercial ...} and the salary is a number. It is usual to find a relation between category and salary (the higher the category, the higher the salary). However, if we want to describe this relation, rules like "If Cat=Manager then Sal=123455" are not the best solution, since there are many different numerical values for the salary of a Manager, and the accuracy and semantic content of this kind of rules will be very poor. What we could prefer are rules like "If Cat=Manager then Sal=*High*", where *High* is a linguistic label that can be represented by a fuzzy set on the domain of the salary. To discover and to assess this kind of rules requires the development of specific tools.

The objective of our work is to provide a definition of *fuzzy approximate dependencies* (FAD for short), an extension of the concept of *approximate dependencies* to the fuzzy case. Roughly speaking, approximate dependencies (AD for short) are functional dependencies with exceptions. AD's can be useful to represent relations between attributes for several purposes, as we shall see later. Our approach is based on previous results on association rule assessment and existing definitions of crisp AD's [23,10] and fuzzy association rules [26]. The definition generalizes several existing relaxations, by means of fuzzy sets, of the concept of functional dependencies. In addition, we shall provide a methodology to obtain algorithms to discover FADs by adapting existing algorithms to discover association rules. A valuable feature of the methodology is that it does not increase time and space complexity, though both are multiplied by a constant.

The paper is organized as follows. Section 2 introduces related work, in particular some concepts we employ in the definition of FADs. Section 3 is devoted to our definition of FAD and applications. We describe how to adapt association rule mining algorithms to the task of finding FADs in Section 4, together with algorithms. In Section 5, a real problem where fuzzy approximate dependencies could be suitable is faced and discussed. Finally, Section 6 contains some concluding remarks and future tasks.

2. Related works

A very common issue in database analysis is the study of existing relations between data stored in a database. Mainly, we can distinguish two basic types of relations. On the one hand, there can be implicit or hidden relations between attribute values, not clear at a first moment. These can be obtained by means of a database analysis. One of the most known examples of this are association rules, defined in [2]. Association rules are "implications" that relate the presence of itemsets (set of items) in a given set of transactions (a T-set). A classical example consider that items are things we can buy in a market, and

transactions are market baskets containing several items. These rules take the form of, for example, "80% of people that buy milk, also buy flour".

On the other hand, we can find explicit relations between data, where implications between attributes can be easily detected (i.e., we can affirm that a job class determines a salary class, or that a given postal code determines the city). This kind of relations are usually integrity constraints or restrictions, imposed during the design phase of a database, according to the model of a real problem. In these cases, we can say that there exists a functional dependency between attributes. Formally, let $R = \{At_1, \ldots, At_m\}$ be a set of attributes and let r be a table with attributes in R such that $|r| = n$. Also, let $X, Y \subset R$ with $X \cap Y = \emptyset$, and let $dom(X) = \{x_1, \ldots, x_K\}$ and $dom(Y) = \{y_1, \ldots, y_M\}$ be the values of X and Y appearing in r. A functional dependency $X \rightarrow Y$ holds in R if and only if for every instance r of R

$$\forall t, s \in r \quad \text{if } t[X] = s[X] \text{ then } t[Y] = s[Y]. \tag{1}$$

Mining for functional dependencies in relational databases have been an object of interest in the field of data mining, because they are very informative about the structure of data. However, it is difficult to discover perfect functional dependencies in a database because one single exception to rule 1 turns the dependency not to hold. But indeed, if the number of exceptions is not very high, such "functional dependencies with exceptions" are showing us interesting regularities that hold in data. Moreover, usual problems such as the presence of noisy data can hide functional dependencies by introducing false exceptions. The proposed solution to these problems is the relaxation of the rule that defines a dependency, in order to accept some exceptions.

2.1. Extensions to the classical model of functional dependencies

We can distinguish two main approaches for extending the concept of functional dependencies, fuzzy functional dependencies and approximate dependencies (also known as partial determinations). The former typically introduces some degree of imprecision in the definition by changing either the granularity level of the attribute domains to a higher level, or the equality into a fuzzy resemblance relation, or the quantifier and implication into fuzzy ones, or several at a time. See [11] for a review, and [16,17,19] for further approaches. Another interesting issue is the search for functional dependencies in fuzzy relational databases (as seen in [60]). The latter will be discussed in the next section.

2.2. Approximate dependencies

Approximate dependencies [13,37,46] can be roughly defined as functional dependencies with exceptions. The definition of approximate dependencies is then a matter of how to define exceptions, and how to measure the accuracy (that is; the proportion of tuples in a relation where the dependency holds) of the dependency (see [11]). We shall follow the approach introduced in [10,23,47], where we applied the same methodology employed in mining for association rules to the discovery of approximate dependencies. The idea is that it is interesting to measure not only the accuracy of the dependency (as other existing approaches do [32,37,46]) but also its support (that is, the proportion of tuples in a relation where the dependency appears), in order to see the empirical evidence associated to the dependency in data. This way, we can avoid to obtain trivial dependencies.

To assess the dependencies, we apply the same measures of interest and accuracy introduced in [1], that is support (the joint probability $p(X \cup Y)$, noted $S(X \rightarrow Y)$) and confidence (the conditional probability $p(Y|X)$, noted $Conf(X \rightarrow Y)$).

Some authors have shown that confidence can yield misleading results in some cases. Basically, the problem with confidence is that it does not take into account the support of Y, hence it is unable to detect statistical independence or negative dependence, i.e., a high value of confidence can be obtained in those cases. This problem is specially important when there are some items with very high support. In the worst case, given an itemset Y such that $S(Y) = 1$, every rule of the form $X \Rightarrow Y$ will be strong provided that $S(X) > minsupp$. It has been shown that in practice, a large amount of rules with high confidence are misleading because of the aforementioned problems.

A summary of papers discussing this problem and the alternative measures proposed is in [7]. There, confidence is used in order to compute an accuracy measure based on certainty factors (see [54] for the definition, and [6,7] for the explanation). Formally, we obtain the certainty factor of a rule as follows,

$$CF(X \Rightarrow Y) = \begin{cases} \frac{(Conf(X \Rightarrow Y)) - S(Y)}{1 - S(Y)} & \text{if } Conf(X \Rightarrow Y) > S(Y), \\ \frac{(Conf(X \Rightarrow Y)) - S(Y)}{S(Y)} & \text{if } Conf(X \Rightarrow Y) < S(Y), \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Certainty factors take values in $[-1, 1]$, indicating the extent to which our belief that the consequent is true varies when the antecedent is also true. It ranges from 1, meaning maximum increment (i.e., when X is true then Y is true) to -1, meaning maximum decrement.

Notice that the two possible extreme cases occur when $S(Y) = 0$ or $S(Y) = 1$. Both cases result onto trivial rules, since no new information can be obtained from them. So, it seems reasonable to give these rules a value of CF = 0.

Returning to our definition of AD, the idea is that, since a functional dependency "$X \rightarrow Y$" can be seen as a rule that relates the equality of attribute values in pairs of tuples (see Eq. (1)), and association rules relate the presence of items in transactions, we can represent approximate dependencies as association rules by using the following interpretations of the concepts of item and transaction:

- An item is an object associated to an attribute of R. For every attribute $At_k \in R$ we note it_{At_k} the associated item.
- We introduce an itemset I_X to be

$$I_X = \{it_{At_k} \mid At_k \in X\}.$$

- T_r is a T-set that, for each pair of tuples $\langle t, s \rangle \in r \times r$ contains a transaction $ts \in T_r$ verifying

$$it_{At_k} \in ts \Leftrightarrow t[At_k] = s[At_k].$$

It is obvious that $|T_r| = |r \times r| = n^2$.

For example, let us consider the relation r shown in Table 1. By means of our definition, the resulting T-set T_r would be the one shown in Table 2.

Then, an approximate dependency $X \rightarrow Y$ in the relation r is an association rule $I_X \Rightarrow I_Y$ in T_r (see [10,23]). The support and certainty factor of $I_X \Rightarrow I_Y$ measure the interest and accuracy of the dependency $X \rightarrow Y$. In particular, the following property holds:

Table 1
A relation, r

	A	B	C
t_1	a_1	b_1	c_1
t_2	a_2	b_1	c_2
t_3	a_1	b_1	c_3
t_4	a_3	b_2	c_3

Table 2
T-set T_r obtained from r

	it_A	it_B	it_C
t_1t_1	1	1	1
t_1t_2	0	1	0
t_1t_3	1	1	0
t_1t_4	0	0	0
t_2t_1	0	1	0
t_2t_2	1	1	1
t_2t_3	0	1	0
t_2t_4	0	0	0
t_3t_1	1	1	0
t_3t_2	0	1	0
t_3t_3	1	1	1
t_3t_4	0	0	1
t_4t_1	0	0	0
t_4t_2	0	0	0
t_4t_3	0	0	1
t_4t_4	1	1	1

Proposition 2.1 (*Blanco et al. [10]*). *If* $CF(X \to Y) = 1$ (*it also implies that* $Conf(X \to Y) = 1$) *then* $X \to Y$ *is a functional dependency*.

The support and accuracy of an approximate dependency $X \to Y$ can be interpreted as an aggregation of the support and accuracy of the association rules that relate values of X to values of Y. Therefore, approximate dependencies can be seen as a summary of the information provided by those associations.

2.3. Fuzzy association rules

Several authors have proposed fuzzy association rules as a generalization of association rules when data is fuzzy or has been previously fuzzyfied ([5,26,31,38,39]). Though most of these approaches have been introduced in the setting of relational databases, we think that most of the measures and algorithms proposed can be employed in a more general framework. A somewhat complete review, including references to papers on extensions to the case of quantitative attributes and hierarchies of items, can be found in [27].

Additional approaches to this problem can be found in [12,14,15,20,30,33,36,40,61]. In [28], several fuzzy data mining measures are discussed. Ref. [62] also relates fuzzy functional dependencies with clustering problems in data bases by means of fuzzy association rules, although this approach is different from ours.

In this paper we shall employ the model proposed in [26]. This model considers a general framework where data is in the form of fuzzy transactions, i.e., fuzzy subsets of items. A (crisp) set of fuzzy transactions is called an FT-set, and fuzzy association rules are defined as those rules extracted from an FT-set.

Let $I = \{i_1, \ldots, i_m\}$ be a set of items and T be a set of fuzzy transactions, where each fuzzy transaction is a fuzzy subset of I. Let $\tilde{\tau} \in T$ be a fuzzy transaction, we note $\tilde{\tau}(i_k)$ the membership degree of i_k in $\tilde{\tau}$. A fuzzy association rule is an implication of the form $A \Rightarrow C$ such that $A, C \subset R$ and $A \cap C = \emptyset$. A and C are called antecedent and consequent, respectively.

It is immediate that the set of transactions where a given item appears is a fuzzy set. We call it *representation* of the item. For item i_k in T we have the following fuzzy subset of T:

$$\tilde{\Gamma}_{i_k} = \sum_{\tilde{\tau} \in T} \tilde{\tau}(i_k)/\tilde{\tau}. \tag{3}$$

This representation can be extended to itemsets as follows: let $I_0 \in R$ be an itemset, its representation is the following subset of T:

$$\tilde{\Gamma}_{I_0} = \bigcap_{i \in I_0} \tilde{\Gamma}_i = \min_{i \in I_0} \tilde{\Gamma}_i. \tag{4}$$

In order to measure the interest and accuracy of a fuzzy association rule, we must use approximate reasoning tools, because of the imprecision that affects fuzzy transactions and, consequently, the representation of itemsets. A semantic approach based on the evaluation of quantified sentences (see [64]) is proposed in [26]. Let Q be a fuzzy coherent quantifier. As defined in [18], Q is a fuzzy coherent quantifier when it verifies the following properties,

- $Q(0) = 0$ and $Q(1) = 1$
- Monotonicity: If $x < y$, $Q(x) \leqslant Q(y)$.

Definition 2.1. (Delgado et al. [26]) The support of an itemset is equal to the result of evaluating the quantified sentence Q of T are $\tilde{\Gamma}_{I_0}$.

Definition 2.2. (Delgado et al. [26]) The support of the fuzzy association rule $A \Rightarrow C$ in the FT-set T, $Supp(A \Rightarrow C)$, is the evaluation of the quantified sentence Q of T are $\tilde{\Gamma}_{A \cup C} = Q$ of T are $(\tilde{\Gamma}_A \cap \tilde{\Gamma}_C)$.

Definition 2.3. (Delgado et al. [26]) The confidence of the fuzzy association rule $A \Rightarrow C$ in the FT-set T, $Supp(A \Rightarrow C)$, is the evaluation of the quantified sentence Q of $\tilde{\Gamma}_A$ are $\tilde{\Gamma}_C$.

The sentences can be evaluated for instance by means of method *GD*, defined in [22] as

$$GD_Q(G/F) = \sum_{\alpha_i \in \triangle(G/F)} (\alpha_i - \alpha_{i+1}) Q\left(\frac{|(G \cap F)_{\alpha_i}|}{|F_{\alpha_i}|}\right), \tag{5}$$

where $\triangle(G/F) = \wedge(G \cap F) \cup \wedge(F)$, $\wedge(F)$ being the set of levels in F, and $\triangle(G/F) = \{\alpha_1, ..., \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ for every $i \in \{1, ..., p\}$. The set F is assumed to be normalized. If not, F is normalized and the normalization factor is applied to $G \cap F$ (see Algorithm 2).

We choose the quantifier Q_M, defined by $Q_M(x) = x$, since it verifies the conditions we request for a quantifier (that is, to be a coherent quantifier) and it has a valuable property: the values obtained by using it in definitions 2.1, 2.2 and 2.3 in the case of crisp transactions, are the ordinary measures of support and confidence in the crisp case. This way, the proposed method is a generalization of the ordinary association rule assessment framework in the crisp case. However, we shall see in Section 3.5 that it can be useful to consider other quantifiers when assessing FADs in order to generalize existing approaches.

Fuzzy relational databases could be seen as a particular case of FT-set. For example, let $R = \{At_1, \ldots, At_m\}$ be a set of attributes, and let $Lab(At_k) = \{a_{k_1}, \ldots, a_{k_n}\}$ be a set of linguistic labels defined on $dom(At_k)$ $\forall At_k \in R$. Let r be a relation with attributes in R. Then, a fuzzy transaction could be obtained from each $t \in R$ as the following fuzzy set:

$$\tilde{\tau}_t = \sum_{k \in \{1,\ldots,m\}} a_{k_i}(t[At_k])/a_{k_i},$$

where each item is a pair (At_k, a_{k_i}) representing 'At_k is a_{k_i}'. In the following, and for the sake of simplicity, we have reduced to the particular case of considering only one label at a time. As a future task, we will study the general case of fuzzy partitions over the attribute domain, that is, a fuzzy value as intersection of several adjacent labels.

3. A new definition: fuzzy approximate dependencies

As discussed in Section 2.1, it is possible to extend the concept of functional dependencies in several ways by smoothing some of the elements of the rule in Eq. (1).

As far as we know, the proposed methodology in this paper is relatively new and original, as we could not find any analogous work in the existing bibliography. Anyway, some approaches can be found that must be mentioned. Fuzzifying the definition of approximate dependencies proposed by [32], based on partitioning the set of tuples in a relation, we must mention the works discussed in [58–60].

3.1. Wang et al. approach

Wang et al. introduce a new data mining technique for extracting approximate dependencies in fuzzy databases in which a set of resemblance relations is defined. In following works, this relations are extended to similarity relations.

According to [53], databases based on resemblance or similarity relations are specially suitable for describing and managing categorical information over discrete domains. Opposing to that, fuzzy set-based models are more appropriate for applying over numeric domains (Table 3).

The definition proposed by the authors is the following. An approximate dependency over a relational scheme R can be expressed as $X \to A$, where $X \subseteq R$ and $A \in R$. Informally, an approximate dependency $X \to A$ holds if all tuples that agree on X approximately also agree on A approximately.

Formally, the dependency holds or is valid in a given fuzzy relation r over R if for all pair of tuples t_i and $t_l \in r$ we have:

$$\text{If } [t_i]_{D_j}^{\alpha_j} = [t_l]_{D_j}^{\alpha_j} \quad \text{for all } D_j \in X, \text{ then } [t_i]_A^{\alpha_j} = [t_l]_A^{\alpha_j}, \tag{6}$$

Table 3
Fuzzy database relation

Emp#	Job	Exp.	Salary
1	Salesman	3	37*K*
2	Design engineer	10	40*K*
3	System engineer	5	45*K*
4	Software engineer	5	45*K*
5	Accountant	12	47*K*
6	Accountant	5	50*K*
7	Secretary	10	53*K*
8	Secretary	15	55*K*

where $[t_i]_{D_j}^{\alpha_j}$ represents the equivalence class of tuple t_i with respect to an attribute D_j with level value α_j. The notation is explained as follows.

Two tuples t_i and t_l are equivalent with respect to an attribute D_j for a given level value α_j if t_{ij} and t_{lj} belong to the same equivalence class of D_j. The equivalence classes of D_j are determined by the level value α_j and defined by the similarity relation. In general, an attribute D_j partitions the tuples of a relation into a set of equivalence classes. The authors denote the equivalence class of a tuple $t_i \in r$ with respect to an attribute D_j with level value α_j by $[t_i]_{D_j}^{\alpha_j}$, i.e.,

$$[t_i]_{D_j}^{\alpha_j} = \{t_l \in r | t_{lj} \approx_{\alpha_j} t_{ij}\}. \tag{7}$$

The set $\pi_{D_j}^{\alpha_j} = \{[t_i]_{D_j}^{\alpha_j} | t_i \in r\}$ of equivalence classes is a partition of r under D_j with level value α_j. That is, $\pi_{D_j}^{\alpha_j}$ is a collection of disjoint sets (equivalence classes) of tuples, such that each set has values belonging to an equivalence class in D_j, and the union of the sets equals the relation r. The rank $|\pi|$ of a partition is the number of equivalence classes in π (Table 4).

Authors start from the concept of partition refinement to obtain approximate dependencies. A partition π is a refinement of another partition π' if every equivalence class in π is a subset of some equivalence class of π'. According to [32], an approximate dependency $X \to A$ holds if and only if π_X refines $\pi_{\{A\}}$.

There is an even simpler test for determining the approximate dependency $X \to A$. If π_X refines $\pi_{\{A\}}$, then adding A to X does not increase any equivalence classes of π_X, thus $\pi_{X\cup\{A\}} = \pi_X$. Consequently, we can find in [32] that an approximate dependency $X \to A$ holds if and only if $|\pi_X| = |\pi_{X\cup\{A\}}|$.

The main improvement introduced by Wang et al. works is the application of resemblance and similarity relations when working on fuzzy relational databases, following the proposed idea in [32], and implementing an extended version of the algorithm proposed in the cited work.

Against that, the main disadvantage found in these works is that of there is no definition of any measure of the interest or certainty of the obtained results.

3.2. Our definition

We want to consider as much cases as we can, integrating both approximate dependencies (exceptions) and fuzzy dependencies. For that purpose, in addition to allowing exceptions, we have considered the relaxation of several elements of the definition of functional dependencies, that allows us to take into account several of the approaches described in [11]. In particular we consider membership degrees

Table 4
Similarity relations over attribute domains in Table 3

Job	*Sw.eng*	*Acct*	*Sys.eng*	*Sales*	*Dn.eng*	
Secr	0.6	0.7	0.6	0.5	0.6	
Sw.eng		0.6	0.8	0.5	0.8	
Acct			0.6	0.5	0.6	
Sys.eng				0.5	0.8	
Sales					0.5	
Exp.	5	10	12	15		
3	0.9	0.7	0.7	0.5		
5		0.7	0.7	0.5		
10			0.9	0.7		
12				0.7		
Sal.	40	45	47	50	53	55
37	0.9	0.7	0.7	0.5	0.5	0.5
40		0.7	0.7	0.5	0.5	0.5
45			0.9	0.5	0.5	0.5
47				0.5	0.5	0.5
50					0.9	0.9
53						0.9

Table 5
Fuzzy relation *r*

	A	*B*	*C*
$\tilde{t}_1$	$a_1, 0.46$	$b_1, 0.76$	$c_1, 0.53$
$\tilde{t}_2$	$a_1, 0.73$	$b_2, 0.06$	$c_1, 0.31$
$\tilde{t}_3$	$a_1, 0.4$	$b_2, 0.28$	$c_1, 0.66$
$\tilde{t}_4$	$a_2, 0.41$	$b_1, 0.49$	$c_1, 0.34$

associated to pairs (attribute, value) as in the case of fuzzy association rules, and also fuzzy similarity relations to smooth the equality of the rule in Eq. (1).

Formally, let $R = \{At_1, \ldots, At_m\}$ be a relational scheme, and r a fuzzy relation over R in the following terms: the intersection between an attribute At_k and a fuzzy tuple $\tilde{t}$ is a pair $\langle \tilde{t}(At_k), \mu_{\tilde{t}}(At_k)\rangle$, being $\tilde{t}(At_k)$ the value of At_k en $\tilde{t}$, and $\mu_{\tilde{t}}(At_k)$ the related membership degree. Table 5 shows an example of a fuzzy relation, r, defined over a relational scheme $R = \{A, B, C\}$.

We consider S_{At_i} a fuzzy similarity relation over $dom(At_i)$. Let $S_R = \{S_{At_k} | At_k \in R\}$. To be more precise, relations in S_R are assumed to be max–min transitive, i.e.

$$S_{At_k}(x_i, x_j) \geqslant \bigvee_{l=1}^{n} min(S_{At_k}(x_i, x_l), S_{At_k}(x_l, x_j)), \forall x_i, x_j \in dom(At_k). \tag{8}$$

We shall define fuzzy approximate dependencies in a relation as fuzzy association rules on a special FT-set obtained from that relation, in the same way that approximate dependencies are defined as association rules on a special T-set.

Let $I_R = \{it_{At_k} | At_k \in R\}$ be the set of items associated to the set of attributes R. We define a FT-set T'_r associated to table r with attributes in R as follows: for each pair of rows $\langle \tilde{t}, \tilde{s} \rangle$ in $r \times r$ we have a fuzzy transaction $\widetilde{ts}$ in T'_r defined as

$$\widetilde{ts}(it_{At_k}) = min(\mu_{\tilde{t}}(At_k), \mu_{\tilde{s}}(At_k), S_{At_k}(\tilde{t}(At_k), \tilde{s}(At_k))) \; \forall it_{At_k} \in T'_r \tag{9}$$

This way, the membership degree of a certain item it_{At_k} in the transaction associated to tuples $\tilde{t}$ and $\tilde{s}$ takes into account the membership degree of the value of At_k in each tuple and the similarity between these values. This value represents the degree to which tuples $\tilde{t}$ and $\tilde{s}$ agree in At_k, i.e., the kind of items that are related by the rule in Eq. (1). On this basis, we define fuzzy approximate dependencies as follows:

Definition 3.1. Let $X, Y \subseteq R$ with $X \cap Y = \emptyset$ and $X, Y \neq \emptyset$. The fuzzy approximate dependency $X \to Y$ in r is defined as the fuzzy association rule $I_X \Rightarrow I_Y$ in T'_r.

The support and certainty factor of $I_X \Rightarrow I_Y$ are calculated from T'_r as explained in Section 2.3, and they are employed to measure the interest and accuracy of $X \to Y$.

Definition 3.2. The support of the fuzzy approximate dependency $X \to Y$ ($I_X \Rightarrow I_Y$ in T'_r), $Supp(X \to Y)$, equals to the evaluation of the quantified sentence Q of T'_r are $\tilde{\Gamma}_{I_X \cup I_Y} = Q$ of T'_r are $(\tilde{\Gamma}_{I_X} \cap \tilde{\Gamma}_{I_Y})$.

Definition 3.3. The confidence of the fuzzy approximate dependency $X \to Y$ ($I_X \Rightarrow I_Y$ in T'_r), $Conf(X \to Y)$, corresponds to the result of evaluating the quantified sentence Q of $\tilde{\Gamma}_{I_X}$ are $\tilde{\Gamma}_{I_Y}$.

Finally, computing the certainty factor is very simple and, as seen before in the case of fuzzy association rules, we can still compute it in the same way we did for the crisp case, applying Eq. (2).

From Eq. (9) it is obvious that $n' = |T'_r| = n^2$ being $n = |r|$. However, we shall see later that it is possible to calculate the support of an itemset I_X in time $O(n)$ with respect to the number of tuples.

Following [26], the FAD $X \to Y$ holds with total accuracy (certainty factor $CF(X \to Y) = 1$) in a relation r iff $\widetilde{ts}(I_X) \leqslant \widetilde{ts}(I_Y) \; \forall \widetilde{ts} \in T'_r$ (let us remember that $\widetilde{ts}(I_X) = \min_{At_k \in X} \widetilde{ts}(it_{At_k}) \; \forall X \subseteq R$). Moreover, since fuzzy association rules generalize crisp association rules, FADs generalize ADs.

3.3. Examples

The following subsection is devoted to show an example to see how our definition works in practice. Table 5 shows a toy fuzzy relation r with attributes in $R = \{A, B, C\}$. Each cell contains both a value and the corresponding membership degree. For every attribute, a fuzzy similarity relation is defined for all possible values. These relations are showed in Table 6. Finally, Table 7A shows the obtained FT-set T'_r. On T'_r, it is possible to apply a fuzzy association rule extraction algorithm.

According to our definition, FARs in T'_r are FADs in r. Table 7B lists some fuzzy approximate dependencies that can be obtained from r, with their respective support (expressed in %) and certainty factor.

Table 6
Fuzzy similarity relations for A, B, and C

a_2	0.3		b_2	0.8		c_2	0.4	
a_3	0.3	0.5	b_3	0.5	0.5	c_3	0	0
	a_1	a_2		b_1	b_2		c_1	c_2

Table 7
(A) The FT-set T'_r obtained from r (B) FADs in r (fuzzy association rules in T'_r)

	it_A	it_B	it_C
$\widetilde{t_1t_1}$	0.46	0.76	0.53
$\widetilde{t_1t_2}$	0.46	0.06	0.31
$\widetilde{t_1t_3}$	0.4	0.28	0.53
$\widetilde{t_1t_4}$	0.3	0.49	0.34
$\widetilde{t_2t_1}$	0.46	0.06	0.31
$\widetilde{t_2t_2}$	0.73	0.06	0.31
$\widetilde{t_2t_3}$	0.4	0.06	0.31
$\widetilde{t_2t_4}$	0.3	0.06	0.31
$\widetilde{t_3t_1}$	0.4	0.28	0.53
$\widetilde{t_3t_2}$	0.4	0.06	0.31
$\widetilde{t_3t_3}$	0.4	0.28	0.66
$\widetilde{t_3t_4}$	0.3	0.28	0.34
$\widetilde{t_4t_1}$	0.3	0.49	0.34
$\widetilde{t_4t_2}$	0.3	0.06	0.31
$\widetilde{t_4t_3}$	0.3	0.28	0.34
$\widetilde{t_4t_4}$	0.41	0.49	0.34

A

$[B] \rightarrow [A]$, $supp$ 20.56%, $conf$ 48.35%, CF 0.35
$[A] \rightarrow [B]$, $supp$ 20.56%, $conf$ 30.86%, CF 0.13
$[C] \rightarrow [A]$, $supp$ 33.44%, $conf$ 60.29%, CF 0.40
$[A] \rightarrow [C]$, $supp$ 33.44%, $conf$ 49.79%, CF 0.24
$[C] \rightarrow [B]$, $supp$ 21.0%, $conf$ 37.82%, CF 0.21
$[B] \rightarrow [C]$, $supp$ 21.0%, $conf$ 53.62%, CF 0.41
$[B, C] \rightarrow [A]$, $supp$ 20.12%, $conf$ 84.61%, CF 0.81
$[C] \rightarrow [A, B]$, $supp$ 20.12%, $conf$ 34.34%, CF 0.18
$[B] \rightarrow [A, C]$, $supp$ 20.12%, $conf$ 46.05%, CF 0.32
$[A, C] \rightarrow [B]$, $supp$ 20.12%, $conf$ 60.38%, CF 0.50
$[A] \rightarrow [B, C]$, $supp$ 20.12%, $conf$ 29.76%, CF 0.12
$[A, B] \rightarrow [C]$, $supp$ 20.12%, $conf$ 92.39%, CF 0.90

B

3.4. Comparison with Wang et al. approach

As an additional example, let us take the same set of objects described in [59]. Table 3 shows us a fuzzy relation in which the job category, experience and salary of eight employees are represented. The defined similarity relations over attributes domains are shown in Table 4. Finally, in order to maintain the same notation used in our definition, let us suppose a membership degree of one for every pair (*attribute*, *value*) in the relation.

Applying our proposed methodology over the set of objects shown in Table 3, and taking into consideration the existing similarity relation between attributes values (Table 4), we obtained the set of fuzzy approximate dependencies that can be found in Table 8.

We shall use this example to show a first difference between our methodology and the one proposed in [59]. In this work, as no measure is defined to inform us about the goodness of the obtained dependencies, the example just concludes that the dependency $[Job, Exp] \rightarrow [Sal]$ holds for this particular set of employees. In this sense, we believe that our methodology brings more richness to the obtained results. In Table 8, we must remark that not only the dependency $[Job, Exp] \rightarrow [Sal]$ is obtained, looking at its certainty factor of 0.82. Moreover, another approximate dependency to be considered is found (Table 7). $[Job, Sal] \rightarrow [Exp]$ has a certainty factor CF = 0.86. In particular, by means of our methodology it is

Table 8
Fuzzy approximate dependencies obtained from Table 3

$[Exp] \rightarrow [Job]$, $supp$ 64.69%, $conf$ 74.07%, CF 0.26
$[Job] \rightarrow [Exp]$, $supp$ 64.69%, $conf$ 86.92%, CF 0.63
$[Sal] \rightarrow [Job]$, $supp$ 61.87%, $conf$ 81.53%, CF 0.51
$[Job] \rightarrow [Sal]$, $supp$ 61.87%, $conf$ 84.56%, CF 0.59
$[Sal] \rightarrow [Exp]$, $supp$ 64.69%, $conf$ 87.91%, CF 0.66
$[Exp] \rightarrow [Sal]$, $supp$ 64.69%, $conf$ 75.37%, CF 0.30
$[Exp, Sal] \rightarrow [Job]$, $supp$ 60.62%, $conf$ 89.12%, CF 0.72
$[Sal] \rightarrow [Job, Exp]$, $supp$ 60.62%, $conf$ 78.53%, CF 0.45
$[Exp] \rightarrow [Job, Sal]$, $supp$ 60.62%, $conf$ 69.42%, CF 0.22
$[Job, Sal] \rightarrow [Exp]$, $supp$ 60.62%, $conf$ 94.56%, CF 0.86
$[Job] \rightarrow [Exp, Sal]$, $supp$ 60.62%, $conf$ 80.61%, CF 0.51
$[Job, Exp] \rightarrow [Sal]$, $supp$ 60.62%, $conf$ 92.95%, CF 0.82

possible to extract all existing fuzzy approximate dependencies for this example. This set of dependencies can be later ordered and reduced, according to the user's necessities, and to the certainty factor.

3.5. Some particular cases

There are several possible scenarios where the concept of FAD can be useful. In each case, specific instantiations of the concept are possible depending on the similarity relations we employ, the presence or not of fuzzy degrees, and even the quantifier employed to calculate the support and confidence (and hence the certainty factor) of the FAD. Some examples are:

- Let us suppose we are interested in looking for ordinary functional dependencies. In this case, let S_{At_k} be the ordinary equality $\forall At_k \in R$, and let r be a crisp relation. In addition, let us employ in expression 2.3 (confidence) the fuzzy quantifier $\forall$ defined as $\forall(x) = 1$ iff $x = 1$ and 0 otherwise. Then we will be looking for ordinary functional dependencies, and the certainty factor of $X \rightarrow Y$ will be 1 iff the functional dependency $X \rightarrow Y$ holds in r, and 0 otherwise.
- Let r be a crisp relation, let S_{At_k} be the ordinary equality $\forall At_k \in R$ and let us employ Q_M in expression 2.3 (confidence). Then we will be looking for ADs as introduced in [23,10].
- Let us suppose that the cardinality of $dom(At_k)$ is very high compared to the number of tuples in r (a typical case is $dom(At_k) \subseteq \mathbb{R}$ such as the attribute *Sal* (salary) in the example in the introduction). One usual way to analyze relations between At_k and other attributes is to employ a set of linguistic labels $Lab(At_k)$ to replace the domain, or to diminish the granularity of the description of At_k in general (again, consider the example in the introduction where $Lab(Sal) = \{High, Medium, \ldots\}$). In this point, we must remark that our intention is not to define a fuzzy partition (i.e., a Ruspini partition) *sensu strictu*, but to establish a set of linguistic labels (according to experts' aid) in order to decrease granularity in data as well as to increase data semantics. Usually, in order to look for dependencies involving At_k, similarity relations can be provided by domain experts in a coherent way with the following resemblance relation

$$R_{At_k}(L_1, L_2) = \max_{x \in dom(At_k)} \min\{L_1(x), L_2(x)\} \tag{10}$$

A similarity relation can be obtained by computing the convex hull of R_{At_k} (see Section 4 for details) if necessary. This is similar to perform a fuzzy clustering on $dom(At_k)$ and then to relate the obtained

clusters with values of other attributes. This way, we can obtain FADs involving At_k that will summarize the information given by all the fuzzy association rules that relate clusters in the domain of At_k to values (or clusters) of other attributes.

- Similarity relations can be useful when the domain of an attribute takes values whose semantics overlaps. For example, consider the attribute *Hair color* and suppose we find in the database values such as blonde, yellow, light, red, orange, etc. (a possible cause is that different users have introduced data in the database without agreeing a set of values for the attribute). If we want to relate hair color to other attributes, we could be interested in taking into account that "*blonde*" and "*yellow*" are similar to some extent, among other similarities. This can be accomplished by using a suitable similarity relation in the domain of the attribute. This way, dependencies involving this attribute should reflect better the possible relations involving hair color.

The aforementioned examples consider we are working with crisp data, that is the most usual case. In the case of fuzzy databases containing fuzzy data (fuzzy degrees and similarity relations), the utility (and even necessity) of a definition of FAD is more clear. Some other possibilities are described in [22,26].

Regarding the final application and utility of FADs, they can provide information about relations (smoothed functional dependencies in general) between attributes in the database. This kind of relations can be seen as the result of an exploratory analysis, and they provide very useful information since when an FAD $X \to Y$ holds with high accuracy we know that there is a set of association rules relating values of X and Y that hold with high accuracy, i.e., we obtain a summary of the accuracy and support of relations between X and Y. Therefore, the process of finding interesting links between attributes in a database could start by looking for FADs and after that looking for association rules, either to obtain a description of a FAD that hold, or to look for possible local associations between values of attributes if the FAD does not hold. This methodology has been employed (using crisp ADs) by experts in the analysis of real databases containing information about soils in [48].

A specific application for FADs is the analysis of correspondences between different fuzzy partitions of the same set of objects. In [48] we have described this application in the case of crisp partitions using ADs to analyze correspondences, and some results in real databases containing data about soils in an agricultural environment have been provided. The extension to the case of FADs and the analysis of existing fuzzy databases will be dealt with in the future.

4. Algorithms

If we want FAD to be a useful concept in practice, we must provide efficient algorithms able to obtain them from real databases. This is not an easy task since we are dealing with a set of transactions of size n^2 with n being the number of tuples in the relation. Since n is usually large in real databases, n^2 can be a too large number, so trying to compute the FT-set T'_r from r and then to search in this set of fuzzy transactions is too expensive. In addition, we must deal with fuzzy degrees, fuzzy similarity relations and computation of quantified sentences, that increase the complexity of the task.

In the field of data mining and knowledge discovery, several algorithms to discover association rules have been presented. We follow the steps of the early algorithm Apriori, introduced in [2], for the sake of simplicity, though the modifications we propose can be applied to more recent and efficient algorithms. Since we extract fuzzy approximate dependencies in terms of association rules, this algorithm remains

the same as the one in [10], with the added complexity of managing fuzzy transactions. In order to achieve this, we must perform similar changes to those in [26].

Usually, an algorithm for association rule extraction comprises two phases. The first one computes the set of frequent itemsets, that is, interesting itemsets with a support greater than a certain threshold, called *minsupp*. The process runs iteratively, computing all the 1-itemsets (itemsets containing 1 item), 2-itemsets, and so on. Each iteration requires a pass over the set of transactions, and because of this, this phase is the most expensive in time.

After all interesting itemsets are extracted, the analysis of them reveals all the association rules with accuracy greater than a certain threshold, called *minconf* (*mincf* in our case). This step usually remains the same in all extraction algorithms and will not be discussed in this paper.

In this section we provide a methodology to adapt existing algorithms to discover association rules, specifically the first step, to the task of discovering FADs. The methodology concerns how to calculate efficiently the support of attributes (our items) by taking into account fuzzy values and similarity relations. A summary of the main aspects of this methodology are the following (we shall describe algorithms later):

- Let us consider first a crisp case. In order to calculate the support of an attribute X (itemset I_X) in T'_r, we store the support in r of each value $x \in dom(X)$, that can be obtained in $\mathcal{O}(n)$. This is the usual information stored by any algorithm looking for association rules in r (items are pairs (attribute,value) in that case, while items are attributes when looking for ADs). From these values, and assuming we employ equality as S_X, the support of I_X in T'_r can be obtained easily as

$$S(I_X) = \frac{1}{n^2} \sum_{x \in dom(X)} x^2. \tag{11}$$

Specifically, once the support for each value of X is obtained, we only must obtain the addition of the square of those values to obtain the support of I_X. We shall discuss the case of fuzzy similarity relations later. Let us remark that, since in the worst case $K = n$, the whole process takes up to $\mathcal{O}(n)$.

In the crisp case, it is even possible to obtain the support of every x and the support of I_X at exactly the same time by using the following result and algorithm 1:

Proposition 4.1 (*Blanco et al. [10]*). *The support of a crisp itemset I_X is*

$$S(I_X) = \frac{1}{n^2} \sum_{i=1}^{K} \sum_{p=1}^{n_{x_i}} (2p - 1). \tag{12}$$

- If fuzzy degrees are associated with values of X in tuples, we employ a fixed set of equidistributed α-cuts for each $x \in dom(X)$. This depends on the precision level we require (a constant), but we chose to employ $k = 100$ α-cuts, that we consider to be sufficient. To do that, we must round or truncate the fuzzy degrees. During the scanning of the tuples in r, what we store is the number of times that a given value x appears with a certain degree. We use a vector we name $N(X, x)$ for that purpose. Calculating $N(X, x)$ takes time $\mathcal{O}(n)$. From $N(X, x)$ it is possible to obtain a similar vector for I_X as seen before, we name V_X. Each position in this vector stores the number of transactions in T'_r where I_X appears with a given degree.

  The support of each x and I_X can be obtained (as the evaluation of the corresponding quantified sentences) from those vectors in time $\mathcal{O}(1)$ by using algorithm 2. Again, the final time complexity is

$\mathcal{O}(n)$. The required storage (a long integer for each $x \in dom(X)$ in any association rule discovery algorithm) is multiplied by a constant (the number of *alpha*-cuts considered).

- Finally, we introduce similarity relations. Alpha-cuts of fuzzy similarity relations (max–min transitive) provide crisp equivalence relations in $dom(X)$. This information can be taken into account during the calculation of V_X from the set of vectors $N(X, x)$ with $x \in dom(X)$. The idea is that the vectors $N(X, x)$ of those values $x \in dom(X)$ that are equal at a certain level according S_X are added to form a single vector at that level. That means that if two values $x_1, x_2 \in dom(X)$ are similar with degree β (i.e. $S_X(x_1, x_2) = \beta$) then for those levels $\alpha \leqslant \beta$ we treat them as the same value. This way, Eq. (11) is applied at each level on the equivalence classes induced by S_X at that level. This information can be incorporated to the process of calculating V_X, from which the support I_X is obtained, without increasing time complexity, see algorithm 5. In fact, it can be calculated before the mining process start, see algorithm 4.

  Fuzzy similarity relations impose a strong restriction as it is the max–min transitivity, that cannot be accomplished by all fuzzy relations defined over a certain domain. Nevertheless, in order to grant this requirement, it is possible to compute the transitive closure of a resemblance relation, in order to obtain the fuzzy similarity relation that we need. In [8], three possible algorithms to obtain the transitive closure of the symmetric matrix of a given fuzzy relation are discussed:
  - By means of the iterative composition $(\vee, \wedge)$, as is described in works like [63] and [56].
  - A column-row exploration algorithm, as the one that can be found in [35].
  - The Prim minimum expansion tree procedure, described in [29].

  According to the followed representation for our similarity relations, the algorithm proposed in [35] seems to be the simplest to take into practice. Algorithm 3 describes the procedure.
- As it was initially expressed, a cell in our fuzzy relation has the following structure, $\langle \tilde{t}(At_k), \mu_{\tilde{t}}(At_k) \rangle$, showing that the degree in which attribute At_k takes value $\tilde{t}(At_k)$ is $\mu_{\tilde{t}}(At_k)$. But an usual case in real problems affected by imprecision is that of an attribute taking more than one value simultaneously, each one with a certain degree (for example, attribute *Hair color* could take values (0.8/*blond*, 0.3/*brown*) for the same person).

  A first solution to face this problem could be the consideration of every pair $(value, degree)$ as belonging to distinct attributes (columns), and then apply the data mining algorithm. Nevertheless, it must be taken into account the restriction of no appearance of a given attribute (even having distinct values) more than one time (for example, simultaneously in the antecedent and in the consequent) in the final rules. The basic procedure of rule (or dependencies) generation could be modified in order to consider this restriction. Nevertheless, the problem appears to be more complex when fuzzy similarity relations are considered over linguistic labels, and not only over attribute values. This particular aspect would be studied in detail as a future task.

Algorithm 6 is the adaptation, following our proposed methodology, of a simple algorithm to find frequent itemsets to the specific case of finding FADs (in particular, Aprori algorithm [2]). In summary, the previous modifications do not increase the complexity of any association rule mining algorithm, though space and time can be increased by a constant that depends mainly on the number of α-cuts considered.

In Algorithm 6 the function $\rho(z, k)$ maps the real value z to the nearest value in the fixed set of levels we are using for the fuzzy degrees. Itemsets are computed ordered by size. Variable l shows the actual size, and acts as a counter of the current stage. The set L_l stores de l-itemsets that are being analyzed and, at the end, it stores the frequent l-itemsets. The procedure $CreateLevel(i, L)$ generates a set of i-itemsets

such that every proper subset with $i-1$ items is frequent (i.e., is in L_{i-1}), and the associated counters. Since every proper subset of a frequent itemset is also a frequent itemset, with this procedure we avoid analyzing itemsets that do not verify this property, hence saving space and time. This valuable property holds also in the fuzzy case, since we compute support by means of *GD* method (see Section 2.3). The following property holds,

Proposition 4.2. *Let X be a set of objects* (*i.e., items*), *and* $A, A' \subset X$. *Then, if* $A \subseteq A'$, $GD_Q(A/X) \geqslant GD_Q(A'/X)$.

Proof. Trivial, since Q is a coherent quantifier, that is, monotonic and non-decreasing. □

4.1. Efficiency study

For our particular case of searching for fuzzy approximate dependencies, Apriori algorithm (introduced in [2] for association rules extraction) is used and properly extended. We chose this algorithm because of its simplicity and because it is one of the most well-known algorithms in this area. Originally, being n the number of transactions (or tuples) and m the number of items, a total number of 2^m itemsets must be considered, in the worst case. As we must compute the support of every considered itemset, that is, count each appearance in the set of transactions, the algorithm total efficiency order can be up to $\mathcal{O}(n \cdot 2^m)$. Nevertheless, if the order is only expressed according to the number of tuples, the previous expression can be reduced to $\mathcal{O}(n)$. In the following, it must be noticed that the order is expressed according to the number of tuples.

In order to extract fuzzy association rules, Apriori algorithm can be extended (as shown in [26]), multiplying the efficiency order by a constant value k, corresponding to the number of considered α-cuts for the storage of fuzzy degrees.

Our definition starts from the one proposed in [10], where in order to obtain approximate dependencies from a relational table, it is possible to apply the corresponding transformation over the original table, and extract association rules that can be viewed as approximate dependencies. The main disadvantage appeared as a number of n^2 transactions (from n tuples) must be considered. Nevertheless, the paper shows how it is possible to maintain the algorithm efficiency order of $\mathcal{O}(n)$.

The proposed algorithm (Algorithm 6) extends Apriori algorithm in the same terms discussed in the previous paragraph, adding the α-cuts factor (for the fuzzy degrees consideration). This way, in a normal case, we can affirm that an acceptable efficiency order for our algorithm could be $\mathcal{O}(k \cdot n)$. As the number of α-cuts, k, is constant, the algorithm order remains basically $\mathcal{O}(n)$.

Nevertheless, we must include an additional factor, that of considering fuzzy similarity relations between attribute values. According to our algorithm, we use these relations when computing the total support of a set of attributes. This process is achieved by Algorithm 5. Being m the number of attributes, and n the number of tuples in the relational table, a total number of $m \cdot \binom{n}{2}$ possible pairs of related values must be considered, in the worst case, that is, when n is the maximum size of every attribute domain.

As we must perform this step in every iteration, the resulting order, for the worst case, can be up to $\mathcal{O}(n \cdot m \cdot \binom{n}{2})$ (considering also the multiplier factor k). This order can be very expensive for databases large enough.

As seen before, our algorithm bottle-neck appears in the management of fuzzy similarity relations. In future works, our efforts will be specially devoted to this aspect, in order to study the convenience and necessity of this type of relations, and how to improve the relations computation and, in particular, the efficiency order.

5. Experiments on real data

The following is an example of fuzzy approximate dependencies extraction over real data. As discussed in the introduction, fuzzy sets can be applied in knowledge discovery tasks in several ways and with interesting results. In many cases data is inherently imprecise or uncertain. A more usual case is that of fuzzy data obtained from crisp data in the preprocessing step by aggregation, summarization or change of granularity level.

An example of both cases is soil data, and more concretely, soil color information. On the one hand, some similarities can be established between attribute values according to semantic relations. On the other hand, the definition of sets of linguistic labels over numeric domains can help us in the reduction of granularity information. Color is a very remarked characteristic of soils. It can be easily determined with little expert aid, and it lets us to qualitatively estimate the sets of materials conforming soil horizons and soil-forming processes [9].

Several authors have studied the existing relations between soil color and soil components ([57,51,52]). In [50], a deeper study of the so called "*Mediterranean red soils*", typical of Mediterranean climate, can be found. Here, statistic tools are applied to suggest and contrast a certain number of hypothesis, relating some soil components with soil color. Unfortunately, most statistical techniques can not be applied over data modelled by means of fuzzy sets. Our intention here is to extend these previous studies by means of fuzzy data mining techniques.

5.1. Bibliographic sources and databases

The studied database consists of information about three mesoenvironments from the South and Southeast of the Iberian Peninsula under Mediterranean climate: Sierra Nevada, Sierra of Gádor and Southeast (involving part of the provinces of Murcia and Almería). We used two Ph.D. Thesis and five cartographic sheets from LUCDEME, scale 1:100000.

Data from Sierra of Gádor was extracted from [42] and consists of 70 soil profiles and 176 horizons. Altitude fluctuates from 100 to 2200 m, and rainfall from 213 mm/year (semiarid climate) to 813 mm/year (wet climate), with a mean annual rainfall of 562 mm/year. Lower annual mean temperature is 6.4 °C and higher is 21.0 °C, with a mean of 12.7 °C. Original material of soils are of carbonated type, mainly limestones and dolomites. Data from Southeast was extracted from LUCDEME soil maps, specifically from sheets 1041 from Vera, Almería [21], 911 from Cehegin, Murcia [4], 1030 from Tabernas, Almería [44], 912 from Mula, Murcia [3] and 1031 from Sorbas, Almería [45]. There is a total of 89 soil profiles and 262 horizons. Altitude fluctuate from 65 to 1120 m, and rainfall from 183 mm/year (arid climate) to 359 mm/year (semiarid climate), with a mean annual rainfall of 300 mm/year. Lower annual mean temperature is 13.2 °C and higher is 19.0 °C, with a mean of 17.0 °C. Geological environment and original materials of soils are extremely different, we can find carbonated, acids and volcanic rocks.

Data from Sierra Nevada was extracted from [49]. There is a total of 35 soil profiles and 103 horizons. Altitude fluctuates from 1320 to 3020 m, and rainfall from 748 mm/year (semihumid climate) to 1287 mm/year (hyperhumid climate), with a mean annual rainfall of 953 mm/year. Lower annual mean temperature is 0.0 °C and higher is 12.1 °C. Geological environment and original materials of soils are mainly acids, but it is not strange to find basic rocks.

Soil colors can be quantified by means of several color systems. The most extended of these systems is the Munsell Color System [41,55]. It is based on three parameters: *Hue*, *Chroma* and *Value*. *Hue* is related with the dominant length wave in reflected radiation, *Value* (or lightness) expresses the proportion of reflected light, and finally, *Chroma* means the chromatic intensity or relative purity on color.

Starting from the correlation matrix appeared in [49], we selected those soil components more correlated (positive or negatively) with *Hue*, *Value* and *Chroma*. The studied components were: *Clay percentage*, *Sand percentage*, and *Organic Carbon percentage*. Database values had to be preprocessed before the analysis. In order to reduce the granularity degree, attributes with numeric domains were discretized, following the discussed techniques in [34], under supervision of domain experts. A set of linguistic labels {*Low*, *Medium*, *High*} was defined for every numeric attribute. Later, these labels were associated to fuzzy sets. Attributes with categorical domains were fuzzified considering fuzzy similarity relations according to semantics between values.

5.2. Results and interpretation

In this section, we apply the domain experts aid in order to give an interpretation of the obtained results. First, as a previous exploratory step, we applied a crisp approximate dependencies (Section 2.2) extraction algorithm. We reduced to the case of one antecedent and one consequent, and fixed a minimum threshold of 0.7 for CF measure. According to this value and to the experts' opinion, the "*best*" obtained approximate dependencies were the following,

$[Dry\ Hue] \rightarrow [Wet\ Hue]$, $supp$ 17.35%, CF 0.91
$[Wet\ Hue] \rightarrow [Dry\ Hue]$, $supp$ 17.35%, CF 0.88
$[Altitude] \rightarrow [Mean\ Annual\ Rainfall]$, $supp$ 35.51%, CF 0.80
$[Mean\ Annual\ Temp.] \rightarrow [Mean\ Annual\ Rainfall]$, $supp$ 31.87%, CF 0.78
$[Free\ iron\ percentage] \rightarrow [Mesoenvironment]$, $supp$ 28.62%, CF 0.76
$[Mean\ Annual\ Rainfall] \rightarrow [Altitude]$, $supp$ 35.51%, CF 0.75
$[Mesoenvironment] \rightarrow [Mean\ Annual\ Rainfall]$, $supp$ 31.16%, CF 0.71

From these dependencies, the first one is trivial, from an expert's point of view, since it just relates the two existing moisture states (wet and dry) for the *Hue* color parameter. This property, as seen before, gives us information about the reflected radiation wave length by the soil sample. *Hue*, opposing to other properties as *Value* and *Chroma*, is hardly modifiable by moisture changes. In this way, the relation is logical for *Hue*, but not for *Value* of *Chroma*.

The second dependency shows the narrow relation between three climatic properties. Following the regional climatic pattern, a higher rainfall corresponds to a higher altitude, and viceversa, as shown by the results. Another revealed relation is the one between temperature and rainfall. In the geographical zone studied, the higher the temperature, the lower the rainfall, almost invariably.

Another result reveals a relation between rainfall and mesoenviroment. This is very reasonable, from the experts' point of view. Looking at the previous dependencies and knowing the existing altitude

gradient in Southeast-Sierra of Gádor-Sierra Nevada, a relation like this was expected. Moreover, *Free iron percentage* seems to be related to soil evolution, that is, the obtained dependency shows that a general evolutive relation between mesoenvironments exists.

For all, it must be remarked that, by means of crisp techniques, it is not possible to obtain dependencies involving color properties (*Hue*, *Chroma*, *Value*) and any other soil attribute, as it was our first objective.

Nevertheless, applying fuzzy approximate dependencies, a higher number of results (up to six times, maintaining the same CF threshold of 0.7) is obtained. This fact, in particular, means a higher possibility of discovering potentially useful information in a database. Within the obtained results, we have selected those dependencies involving soil color properties and other attributes,

$[\%\ Organic\ Carbon] \rightarrow [Wet\ Value]$, $supp$ 27.6%, CF 0.75
$[CEC] \rightarrow [Wet\ Value]$, $supp$ 26.31%, CF 0.7
$[\%\ Clay] \rightarrow [Dry\ Chroma]$, $supp$ 28.9%, CF 0.99
$[\%\ Organic\ Carbon] \rightarrow [Dry\ Chroma]$, $supp$ 31.4%, CF 0.91
$[\%\ Calcic\ Carbonate] \rightarrow [Dry\ Chroma]$, $supp$ 27.12%, CF 0.8
$[CEC] \rightarrow [Dry\ Chroma]$, $supp$ 27.11%, CF 0.71
$[\%\ Useful\ water] \rightarrow [Dry\ Chroma]$, $supp$ 29.12%, CF 0.97

Organic carbon percentage, Cation exchange capacity (CEC), Clay percentage, Calcic Carbonate percentage and Useful water in soil conform the most relevant properties group in soil analysis. Because of this, these obtained dependencies have a very high interest, according to experts' opinion.

Relations between *CEC* and *Organic carbon* are well-known in the studied knowledge area. A higher *Organic carbon percentage* (humus content in soil), implies a higher *Value* (lower luminosity, darker soils). This fact is always present in studied soils, as experts verify. This effect is clearly more remarkable in wet soils (attribute *Wet Value*) than in dry soils (obtained dependencies involving *Dry Value* gave only CF under 0.54). For this particular case, organic matter shows a higher *CEC* (about 300 cmol(+)/kg) than the remaining soil components (i.e., clay has a value of 30 cmol(+)/kg), so a narrow relation appears between these attributes. For that reason, the fuzzy approximate dependency $[CEC] \rightarrow [Wet\ Value]$ is easily explainable, although with a lower CF.

In other order of things, the dependency between *Clay percentage* and *Dry Chroma* is almost perfect. In every soil in the world, a higher value for *Clay percentage* implies a higher *Chroma*, since this is invariably related to iron oxide fine particle releasing (clay-scale particles), acting as pigments and intensifying soil color. This effect is even more remarkable on dry soils, since sample humidification reduces reflection. The meaning of variation of *Dry Chroma* with *Organic carbon* is not so clear, even though they appear to be very related. A higher quantity of humus generates an intensity loss in soil color, but a local study by means of association rules between attribute values would be desirable, in order to verify this hypothesis. Then, experts could confirm the direction of the dependencies involving *Dry Chroma* with *CEC* and *Useful water*, as both are very dependent of *Clay* and *Organic carbon percentages*. Nevertheless, the dependencies seems to be reasonable in some measure.

Finally, the fuzzy approximate dependency between *Calcic carbonate percentage* and *Dry Chroma* seems to be very interesting, according to experts. A priori, it could be argued that a high carbonate percentage should lead to a low intense and whitish soil color.

For this particular case, asked experts were highly satisfied as knowledge extracted by means of fuzzy data mining was more suitable to "*fusion*" or comparison with expert knowledge that crisp. Moreover, fuzzy data mining was sensitive to low support dependencies, that were discarded in crisp data mining.

Agricultural information, and in particular, soil data, is inherently affected by imprecision of uncertain factors and can be modelled very efficiently in fuzzy databases.

6. Concluding remarks and future tasks

Data mining can benefit from fuzzy set technology, since the latter allows to obtain more understandable relations in data. In this paper we have proposed a methodology to obtain what we call fuzzy approximate dependencies (FAD) from databases. FADs generalize several existing ways to smooth functional dependencies, and provide information about relations at the attribute level. We have enumerated several scenarios where the concepts introduced can be useful, both for analyzing crisp and, obviously, fuzzy data. The proposed methodology can be implemented by modifying existing algorithms to discover association rules without increasing the theoretical complexity, though time and space are increased by a constant related to the number of α-cuts we consider.

We have employed this methodology to adapt A priori in order to discover FADs. Moreover, we have discussed a real problem where our methodology can be suitable. Our preliminary experiments suggests that both time and space employed in the mining process are acceptable, though more detailed reports will be provided as the result of currently ongoing experiments. As a future work we plan to adapt recent and more efficient algorithms to discover association rules for the purpose of mining FADs.

Additional future tasks will be to study different kinds of fuzzy relations that can be necessary in order to apply FADs in the analysis of fuzzy databases, in particular when different ways to represent uncertainty and imprecision are employed in the database model. We also plan to employ FADs to analyze correspondences between fuzzy partitions of the same set of objects in real databases, as suggested in [48].

Acknowledgements

This work is part of the research project Fuzzy-KIM CICYT TIC2002-04021-C02-02.

Appendix Algorithms

Algorithm 1 (Blanco et al. [10]) Algorithm to obtain the support of a crisp itemset I_X

$S(I_V) \leftarrow 0$
for all $i \in \{1, .., K\}$ **do**
  $N(V, v_i) \leftarrow 0$
end for
for all $t \in r$ **do**
  $N(V, t[V]) \leftarrow N(V, t[V]) + 1$
  $S(I_V) \leftarrow S(I_V) + 2N(V, t[V]) - 1$
end for
Exit: $S(I_V)/n^2$ is the support of the itemset I_V

Algorithm 2 (Delgado et al. [24]) Algorithm to obtain $GD_Q(C/A)$ from V_A and $V_{A\cup C}$

$j \leftarrow k$; $GD \leftarrow 0$; $nf(A)^* \leftarrow k$; $acum_A \leftarrow 0$; $acum_{A\cup D} \leftarrow 0$
{Calculate $nf(A)^* = nf(A)^* \times k$}
{This is the normalization factor}
while $nf(A)^* > 0$ y $V_A(nf(A)^*) = 0$ **do**
  $nf(A)^* \leftarrow nf(A)^* - 1$
end while
if $nf(A)^* = 0$ **then**
  return ("Error"); End
end if
while $j > 0$ **do**
  $acum_{A\cup D} \leftarrow acum_{A\cup D} + V_{A\cup D}(j)$
  $acum_A \leftarrow acum_A + V_A(j)$
  if $j \leqslant nf(A)^*$ **then**
    $GD \leftarrow GD + Q(\frac{acum_{A\cup D}}{acum_A})$
  end if
  $j \leftarrow j - 1$
end while
{Normalization}
$GD \leftarrow \frac{GD}{nf(A)^*}$
Return(GD); End

Algorithm 3 (Kandal and Yelowtiz [35]). Computes the transitive closure in a fuzzy relation matrix

Label all possible values in $1, \ldots, N$
Build the resemblances primitive matrix ρ, where entry ij represents the resemblance degree between i and j values
for $K = 1$ to N **do**
  for $I = 1$ to N **do**
    if $\rho(I, K) \neq 0$ **then**
      for $J = 1$ to N **do**
        $\rho(I, J) = max(\rho(I, J), min(\rho(I, K), \rho(K, J)))$
      end for
    end if
  end for
end for

Algorithm 4 Algorithm to obtain the set of equivalence classes for a given fuzzy itemset I_X

Require: I_X a fuzzy itemset of attributes, S_X a fuzzy similarity relation, deg_X, an α-cut.
Ensure: Υ, a set of equivalence classes.

for all $x \in dom(X)$ **do**
  $\overline{x} \leftarrow \{x' | x' \in dom(X)\ and\ S_X(x, x') \geqslant deg_X\}$
  $\Upsilon \leftarrow \Upsilon \cup \{\overline{x}\}$
end for
Exit: return Υ, the set of equivalence classes for itemset I_X at α-cut deg_X.

Algorithm 5 Algorithm to compute the support of a given fuzzy itemset I_X of attributes

$j \leftarrow k$
while $j > 0$ **do**
  $\Upsilon \leftarrow ComputeEquivClasses(I_X, S_X, j)$ (see Algorithm 4)
  for all $\overline{x} \in \Upsilon$ **do**
    $acum_{\overline{x}} \leftarrow 0$
  end for
  for all $x \in dom(X)$ **do**
    $acum_{\overline{x}} \leftarrow acum_{\overline{x}} + N_{(X,x)}[j]$
  end for
  $V_{(I_X)}[j] \leftarrow V_{(I_X)}[j] + \sum_{\overline{x} \in \Upsilon} acum_{\overline{x}}^2$
  $j \leftarrow j - 1$
end while
Return $V_{(I_X)}$; End.

Algorithm 6 Algorithm to obtain frequent itemsets from T'_r, i.e., first step when looking for FADs

Require: R, a set of attributes (our items); r a fuzzy relation in R; S_R, a set of similarity relations for each attribute in R.
Ensure: F, the set of all frequent fuzzy itemsets.

$F \leftarrow \emptyset$; $l \leftarrow 1$; $L_1 \leftarrow \emptyset$
for all attribute $At \in R$ **do**
  Allocate memory for $V_{(\{it_{At}\})}$, an array of $k+1$ positions initialized to 0
  $L_1 \leftarrow L_1 \cup \{\{it_{At}\}\}$
  for all $a \in dom(At)$ **do**
    Allocate memory for $N_{(At,a)}$, an array of $k+1$ positions initialized to 0
  end for
end for
while $l \leqslant m$ y $L_l \neq \emptyset$ **do**
  for all tuple $\tilde{t} \in r$ **do**
    for all itemset $I_X \in L_l$ **do**
      $N_{(X,\tilde{t}(x))}[\rho(\mu_{\tilde{t}}(X), k)] \leftarrow N_{(X,\tilde{t}(x))}[\rho(\mu_{\tilde{t}}(X), k)] + 1$
    end for
  end for
  for all itemset $I_X \in L_l$ **do**
    Compute $V_{(I_X)}$ (see Algorithm 5)
    Free memory of every $N_{(X,x)}$, $\forall x \in dom(X)$
    Compute $GD_Q(\tilde{I}_{I_X}, T'_r)$ (see Algorithm 2)
    if $GD_Q(\tilde{I}_{I_X}, T'_r) < minsupp$ **then**
      $L_l \leftarrow L_l \backslash \{I_X\}$
      Free memory of $V_{(I_X)}$
    end if
  end for
  $F \leftarrow F \cup L_l$; $L_{l+1} \leftarrow CreateLevel(l+1, L_l)$; $l \leftarrow l+1$
end while
Return F, the set of all frequent fuzzy itemsets.

References

[1] R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in large databases, in: Proc. ACM SIGMOD Conf. on Management of Data, Washington, D.C., May,1993, pp. 207–216.

[2] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: Proc. 20th Internat. Conf. on Very Large Databases, Santiago, Chile,September 1994, pp. 478–499.

[3] J. Alias, Mapa de suelos de Mula, Mapa 1:100000 y memoria, LUCDEME; MAPA-ICONA-University of Murcia, 1986.

[4] J. Alias, Mapa de suelos de Cehegin, Mapa 1:100000 y memoria, LUCDEME; MAPA-ICONA, University of Murcia, 1987.

[5] W.H. Au, K.C.C. Chan, An effective algorithm for discovering fuzzy rules in relational databases, in: Proc. IEEE Internat. Conf. on Fuzzy Systems, Vol. II, 1998, pp. 1314–1319.

[6] F. Berzal, I. Blanco, D. Sánchez, M.A. Vila, A new framework to assess association rules, in: F. Hoffmann (Ed.), Advances in Intelligent Data Analysis, Fourth International Symposium, IDA'01, Lecture Notes in Computer Science 2189, Springer, Berlin, 2001, pp. 95–104.

[7] F. Berzal, I. Blanco, D. Sánchez, M. Vila, Measuring the accuracy and interest of association rules: a new framework, Intelligent Data Analysis 6 (2002) 221–235.

[8] J.C. Bezdek, J.D. Harris, Fuzzy partitions and relations: an axiomatic basis for clustering, Fuzzy Sets and Systems 1, North-Holland, Amsterdam, 1978, pp. 111–127.

[9] J.M. Bigham, E.J. Ciolkosz, Eds. Soil color. Soil Sci. Soc. Am. Spec. Publ. No. 31, (1993) 159.

[10] I. Blanco, M.J. Martín-Bautista, D. Sánchez, M.A. Vila, On the support of dependencies in relational databases: strong approximate dependencies. Data Mining and Knowledge Discovery, submitted.

[11] P. Bosc, L. Lietard, O. Pivert, Functional dependencies revisited under graduality and imprecision, 1997 Annual Meeting of NAFIPS, 1997, pp. 57–62.

[12] P. Bosc, D. Dubois, O. Pivert, H. Prade, On fuzzy association rules based on fuzzy cardinalities, FUZZ-IEEE (2001) 461–464.

[13] P.D. Bra, J. Paredaens, Horizontal decompositions for handling exceptions to functional dependencies, Adv. Database Theory 2 (1983) 123–144.

[14] K.C.C. Chan, W.-H. Au, Mining Fuzzy Association Rules, CIKM (1997) 209–215.

[15] G. Chen, Q. Wei, Fuzzy association rules and the extended mining algorithms, Inform. Sci. 147 (1–4) (2002) 201–228.

[16] J.C. Cubero, M.A. Vila, A new definition of fuzzy functional dependency in fuzzy relational databases, Int. J. Intelligent Systems 9 (5) (1994) 441–448.

[17] J.C. Cubero, O. Pons, M.A. Vila, Weak and strong resemblances in fuzzy functional dependencies, in: Proc. IEEE Internat. Conf. on Fuzzy Systems, Orlando/FL, USA,1994, pp. 162–166.

[18] J.C. Cubero, J.M. Medina, O. Pons, M.A. Vila. The generalized selection: an alternative way for the quotient operations in fuzzy relational databases, in: B. Bouchon-Meunier, B. Yager, R.R. Zadeh, L.A. (Eds.), Fuzzy Logic and Soft Computing, Vol. 5, World Scientific, USA, 1995, 5, pp. 214–250.

[19] J.C. Cubero, F. Cuenca, I. Blanco, M.A. Vila, Incomplete functional dependencies versus knowledge discovery in databases, in: Proc. EUFIT'98, Aachen, Germany,1998, pp. 731–774.

[20] J.M. de Graaf, W.A. Kosters, J.J.W. Witteman (2001). Interesting Fuzzy Association Rules in Quantitative Databases, PKDD (2001) 140–151.

[21] G. Delgado, et al., Mapa de Suelos de Vera, LUCDEME, ICONA-Universidad de Granada, 1991.

[22] M. Delgado, D. Sánchez, M.A. Vila, Fuzzy quantified dependencies in relational databases, in: Proc. EUFIT'99, Aachen, Germany,1999.

[23] M. Delgado, M.J. Martín-Bautista, D. Sánchez, M.A. Vila, Mining strong approximate dependencies from relational databases, in: Proc. IPMU'2000, Vol. 2, Madrid, Spain, 2000a, pp. 1123–1130.

[22] M. Delgado, D. Sánchez, M.A. Vila, Fuzzy cardinality based evaluation of quantified sentences, Internat. J. Approx. Reason. 23 (2000) 23–66.

[24] M. Delgado, D. Sánchez, M.A. Vila, Fuzzy cardinality based evaluation of quantified sentences, Internat. J. Approx. Reason. 23 (2000b) 23–66.

[25] M. Delgado, D. Sánchez, J.M. Serrano, M.A. Vila, A survey of methods to evaluate quantified sentences, Mathware and Soft Comput. VII(2–3) (2000) 149–158.

[26] M. Delgado, N. Marín, D. Sánchez, M.A. Vila, Fuzzy association rules: general model and applications, IEEE Trans. Fuzzy Systems 11 (2) (2003) 214–225.
[27] M. Delgado, N. Marín, D. Sánchez, M.A. Vila, Mining fuzzy association rules: an overview. 2003 BISC International Workshop on Soft Computing for Internet and Bioinformatics, 2003, accepted.
[28] D. Dubois, E. Hüllermeier, H. Prade, A note on quality measures for fuzzy association rules. Fuzzy Sets and Systems—IFSA 2003, Lecture Notes in Artificial Intelligence, Vol. 2715, Springer, Berlin, pp. 346–353.
[29] J. Dunn, A graph theoretic analysis of pattern classification via Tamura's fuzzy relation, IEEE Trans. SMC-4 3 (1974) 310–313.
[30] A. Gyenesei, Interestingness measures for fuzzy association rules, in: Proc. PKDD,2001, pp. 152–164.
[31] T.P. Hong, C.S. Kuo, S.C. Chi, Mining association rules from quantitative data, Intelligent Data Analysis 3 (1999) 363–376.
[32] Y. Huhtala, J. Karkkainen, P. Porkka, H. Toivonen, Efficient discovery of functional and approximate dependencies using partitions, in: Proc. 4th Internat. Conf. on Data Engineering,1998, pp. 392–401.
[33] E. Hüllermeier, Implication-Based Fuzzy Association Rules, in: Proc. PKDD,2001, pp. 241–252.
[34] F. Hussain, H. Liu, C.L. Tan, M. Dash, Discretization: an enabling technique, Technical Report, The National University of Singapore, June 1999.
[35] A. Kandel, L. Yelowitz, Fuzzy chains, IEEE Trans. SMC-4 5 (1974) 472–475.
[36] M. Kaya, R. Alhajj, F. Polat, A. Arslan, Efficient Automated Mining of Fuzzy Association Rules, DEXA (2002) 133–142.
[37] J. Kivinen, H. Mannila, Approximate dependency inference from relations, Theoretical Computer Science 149 (1) (1995) 129–149.
[38] C.-M. Kuok, A. Fu, M.H. Wong, Mining fuzzy association rules in databases, SIGMOD Record 27 (1) (1998) 41–46.
[39] J.H. Lee, H.L. Kwang, An extension of association rules using fuzzy sets, in: Proc. IFSA'97, Prague, Czech Republic,1997.
[40] J. Luo, S. Bridges, Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection, Internat. J. Intelligent Systems 15 (8) (2000) 687–703.
[41] A. Munsell, Soil Color Charts, Munsell Color Company Inc., Baltimore, Maryland, 1954.
[42] C. Oyonarte, Estudio Edáfico de la Sierra de Gádor (Almería), Evaluación para usos forestales, Ph.D. Thesis, University of Granada, 1999.
[43] W. Pedrycz, Fuzzy Set Technology in Knowledge Discovery, Fuzzy Sets and Systems 98 (1998) 279–290.
[44] A. Pérez-Pujalte, Mapa de suelos de Tabernas, Mapa 1:100000 y memoria, LUCDEME; MAPAICONA-CSIC, 1987.
[45] A. Pérez-Pujalte, Mapa de suelos de Sorbas, Mapa 1:100000 y memoria, LUCDEME; MAPA-ICONA-CSIC, 1989.
[46] B. Pfahringer, S. Kramer, Compression-based evaluation of partial determinations, in: Proc. First Internat. Conf. Knowledge Discovery and Data Mining (KDD'95),1995, pp. 234–239.
[47] D. Sánchez, Adquisición de relaciones entre atributos en bases de datos relacionales, Ph.D. Thesis (in Spanish), University of Granada, 1999.
[48] D. Sánchez, J.M. Serrano, M.A. Vila, V. Aranda, J. Calero, G. Delgado, Using Data Mining Techniques to Analyze Correspondences Between User and Scientific Knowledge in an Agricultural Environment, in: M. Piattini, J. Filipe, J. Braz (Eds.), Enterprise Information Systems IV, Kluwer Academic Publishers, Dordrecht, MA, 2003, pp. 75–89.
[49] M. Sánchez-Marañón, Los suelos del Macizo de Sierra Nevada, Evaluación y capacidad de uso (in Spanish), Ph.D. Thesis, University of Granada, 1992.
[50] M. Sánchez-Marañón, G. Delgado, M. Melgosa, E. Hita, R. Delgado, CIELAB color parameters and their relationship to soil characteristic in Mediterranean Red Soils, Soil Sci. 11 (1997) 833–842.
[51] U. Schwertmann, Relations between iron oxides, soil color and soil formation, Soil Sci. Soc. Am. Spec. Publ. N? 31 (1993) 51–69.
[52] D.G. Schulze, J.L. Nagel, G.E. Van Scoyoc, T.L. Henderson, M.F. Baumgardner, D.E. Scott, Significance of organic matter in determining soil colors, Soil Sci. Soc. Am. Spec. Publ. N? 31 (1993) 71–90.
[53] S. Shenoi, A. Melton, Proximity relations in the fuzzy relational database model, Fuzzy Sets and Systems 31 (1989) 285–296.
[54] E. Shortliffe, B. Buchanan, A model of inexact reasoning in medicine, Math. Biosci. 23 (1975) 351–379.
[55] Soil Survey Staff Soil Taxonomy, U.S. Dept. Agri. Handbook No. 436, 754pp.
[56] S. Tamura, S. Higuchi, K. Tanaka, Pattern classification based on fuzzy relations, IEEE Trans. SMC-1 (1971) 61–66.
[57] J. Torrent, U. Schwertmann, D.G. Schulze, Iron oxide mineralogy of two river terraces sequences in Spain, Geoderma 23 (1980) 191–208.

[58] S.-L. Wang, J.-S. Tsai, Discovery of approximate dependencies from proximity-based fuzzy databases, in: Proc. 3rd Internat. Conf. on Knowledge-based Intelligent Information Engineering Systems, August 1999, Adelaide, Australia,1999, pp. 234–237.
[59] S.-L. Wang, J.-S. Tsai, B.-C. Chien, Mining Approximate Dependencies Using Partitions on Similarity-relation-based Fuzzy Databases, in: Proc. IEEE SMC'99, October 1999, Tokyo, Japan, V-871–V-875,1999.
[60] S.-L. Wang, J.-S. Tsai, T.-P. Hong, Mining functional dependencies from fuzzy relational databases, in: Proc. ACM SAC 2000, Fuzzy Application and Soft Computing Track, March 2000, Italy,2000, pp. 490–493.
[61] H.-T. Yang, S.-J. Lee, Mining Fuzzy Association Rules from Sequence databases with Quantitative data and Inter-Transaction Intervals, FSKD (2002) 606–610.
[62] Y. Yang, M. Singhal, Fuzzy functional dependencies and fuzzy association rules, DaWaK (1999) 229–240.
[63] L.A. Zadeh, Similarity relations and fuzzy orderings, Inform. Sci. 3 (1971) 177–200.
[64] L.A. Zadeh, A computational approach to fuzzy quantifiers in natural languages, Comput. Math. Appl. 9 (1) (1983) 149–184.