

# On Some New Types of Fuzzy Dependencies

C. Molina<sup>a</sup>, D. Sánchez<sup>1,b</sup>, J.M. Serrano<sup>a</sup>, M.A. Vila<sup>b</sup>

<sup>a</sup>Department of Informatics, University of Jaén, Spain

*e-mail* {carlosmo, jschica}@ujaen.es

<sup>b</sup>Department of Computer Science and A.I., University of Granada

C/ Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain

*e-mail*: {daniel,vila}@decsai.ugr.es

## Abstract

We introduce fuzzy dependencies with new semantics as a generalization of both fuzzy approximate and fuzzy gradual dependencies. The new dependencies are defined as fuzzy association rules relating different types of items on a particular set of transactions obtained from a database. We discuss the semantics and we illustrate the new types of dependencies with results of our experiments. Algorithmic issues are also briefly discussed.

**Keywords:** Fuzzy dependencies; gradual dependencies; approximate dependencies; association rules.

## 1 Introduction

Several types of dependencies and fuzzy dependencies have been introduced in the literature by many authors. Most of them are fuzzy extensions of functional dependencies in relational databases. Given two attributes<sup>2</sup>  $X, Y$  a functional dependence is an expression of the form  $X \rightarrow Y$ . It is said that  $X \rightarrow Y$  holds if for every set of data (instance)  $r$ , Eq. 1 holds.

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

<sup>1</sup>Corresponding author.

<sup>2</sup>The definition of functional dependence and the different extensions discussed in this paper consider that  $X$  and  $Y$  are sets of attributes in general.

The different fuzzy extensions of functional dependencies are called *fuzzy functional dependencies*, and are obtained by replacing some of the elements in Eq. 1 by their fuzzy counterparts in different ways (see [4] for a review of the main approaches). As a particular case, approximate dependencies (see [11]) replace the universal quantifier by a less restrictive one, allowing exceptions to the rule. Fuzzy approximate dependencies [2] are a fuzzy extension of approximate dependencies that consider a fuzzy similarity relation instead of a crisp equality between values of attributes. A more general kind of fuzzy functional dependence that unify many of the different approaches described in [4] can be found in [6].

A completely different kind of dependencies, called gradual dependencies, was proposed in [9]. Gradual dependencies represent tendencies in the variation of the degree of fulfillment of imprecise properties in a set of objects. Since the imprecise properties are described by means of fuzzy sets, these dependencies have been also called fuzzy gradual dependencies [10]. The variations in the membership degree considered in gradual dependencies can be of two types: *the more* and *the less*, meaning that the membership degree of the first object to the considered fuzzy set is greater or lower than the membership of the second one, respectively. Hence we can consider four types of gradual dependencies: *the more X is A, the more Y is B* (expressed as  $(>, X, A) \rightarrow (>, Y, B)$ ), *the more X is A, the less Y is B* (expressed as  $(>, X, A) \rightarrow (<, Y, B)$ ), and so on. As an ex-

$$\forall t, s \in r \text{ if } A(t[X]) *_{1} A(s[X]) \text{ then } B(t[Y]) *_{2} B(s[Y]) \quad (2)$$

ample, consider a database containing data about weight and speed of a set of trucks, and consider the restrictions *high* related to weight and *slow* related to speed, represented by means of suitable fuzzy sets on the domains of the attributes. Examples of gradual dependencies are *the higher the weight, the lower the speed*, meaning that as the weight of a truck increases, its speed tends to decrease, and *the higher the weight, the higher the speed*, meaning the opposite tendency.

In [10] and previous works, gradual dependencies are interpreted as rules. Following that approach, a gradual dependence  $(*_1, X, A) \rightarrow (*_2, Y, B)$  with  $*_1, *_2 \in \{<, >\}$  is defined by the rule in Eq. 2.

On the basis of Eq. 1 and 2, interpretations of the different types of dependencies that we have mentioned above as fuzzy association rules have been provided in [11, 2, 10]. These interpretations, based on suitable definitions of the abstract notions of item and transaction, have the advantage that the measures and algorithms to mine for fuzzy association rules can be employed in order to assess and to mine for dependencies of different types.

In this paper we put together the different types of items employed to interpret dependencies as association rules. The resulting dependencies are fuzzy dependencies of different types, with new semantics, that generalize both approximate and gradual dependencies. We introduce and discuss some of the possible resulting semantics, and we illustrate them with the results of some experiments.

The paper is organized as follows: in section 2 we introduce fuzzy association rules and the interpretation of dependencies as association rules provided in [11, 2, 10]. In section 3 we introduce new dependencies and discuss about their semantics. Algorithmic issues and some examples obtained in our experiments are shown in section 4. Finally, section 5 contains our conclusions and future work.

## 2 Dependencies as Association Rules

In this section we briefly remember some notions on association rules and fuzzy association rules, and the definition of fuzzy dependencies as fuzzy association rules introduced in [11, 2, 10].

### 2.1 Association Rules

Let  $I$  be a set of items and  $T$  a set of transactions with items in  $I$ , both assumed to be finite. An association rule is an expression of the form  $I_1 \Rightarrow I_2$ , where  $I_1, I_2 \subseteq I$ ,  $I_1, I_2 \neq \emptyset$ , and  $I_1 \cap I_2 = \emptyset$ . The rule  $I_1 \Rightarrow I_2$  means “every transaction of  $T$  that contains  $I_1$  contains  $I_2$ ”.

The usual measures to assess association rules are support and confidence, both based on the concept of support of an *itemset* (i.e. a subset of items). The support of an itemset  $I_0 \subseteq I$ , and the support and confidence of the rule  $I_1 \Rightarrow I_2$  in a set of transactions  $T$ , are those of Eq. 3, 4 and 5, respectively.

$$\text{supp}(I_0) = \frac{|\{\tau \in T \mid I_0 \subseteq \tau\}|}{|T|} \quad (3)$$

$$\text{Supp}(I_1 \Rightarrow I_2) = \text{supp}(I_1 \cup I_2) \quad (4)$$

$$\text{Conf}(I_1 \Rightarrow I_2) = \frac{\text{Supp}(I_1 \Rightarrow I_2)}{\text{supp}(I_1)} \quad (5)$$

Several authors have pointed out the drawbacks of confidence and have proposed alternative measures. Following the proposal in [3], we shall employ Shortliffe and Buchanan’s certainty factors, defined as in Eq. 6.

### 2.2 Fuzzy association rules

Association rules have been extended to fuzzy association rules in different ways (see [5, 8] for a review). In [5], fuzzy association rules are defined and assessed as follows: let  $I = \{i_1, \dots, i_m\}$  be a set of items and  $\tilde{T}$  be a set of fuzzy transactions, where each fuzzy transaction is a fuzzy subset of  $I$ . For every fuzzy

$$CF(I_1 \Rightarrow I_2) = \begin{cases} \frac{conf(I_1 \Rightarrow I_2) - supp(I_2)}{1 - supp(I_2)} & conf(I_1 \Rightarrow I_2) > supp(I_2) \\ \frac{conf(I_1 \Rightarrow I_2) - supp(I_2)}{supp(I_2)} & conf(I_1 \Rightarrow I_2) \leq supp(I_2) \end{cases} \quad (6)$$

transaction  $\tilde{\tau} \in \tilde{T}$  we note  $\tilde{\tau}(i_k)$  the membership degree of  $i_k$  in  $\tilde{\tau}$ . For an itemset  $I_0$  we note  $\tilde{\tau}(I_0) = \min_{i_k \in I_0} \tilde{\tau}(i_k)$  the degree to which  $I_0$  is in a transaction  $\tilde{\tau}$ . A fuzzy association rule in  $\tilde{T}$  is an implication of the form  $I_1 \Rightarrow I_2$  such that  $I_1, I_2 \subset I$  and  $I_1 \cap I_2 = \emptyset$ .

We call *representation* of the item  $i_k$ , denoted by  $\tilde{\Gamma}_{i_k}$ , to the (fuzzy) set of transactions where  $i_k$  appears, defined as in equation 7. This representation can be extended to itemsets as in equation 8.

$$\tilde{\Gamma}_{i_k}(\tilde{\tau}) = \tilde{\tau}(i_k) \quad (7)$$

$$\tilde{\Gamma}_{I_0}(\tilde{\tau}) = \min_{i_k \in I_0} \tilde{\Gamma}_{i_k}(\tilde{\tau}) = \min_{i_k \in I_0} \tilde{\tau}(i_k) = \tilde{\tau}(I_0) \quad (8)$$

In order to measure the interest and accuracy of a fuzzy association rule, we employ a semantic approach based on the evaluation of quantified sentences, using the fuzzy quantifier  $Q_M(x) = x$ , as follows:

- The support of an itemset  $I_0$  is the evaluation of the quantified sentence  $Q_M$  of  $\tilde{T}$  are  $\tilde{\Gamma}_{I_0}$ .
- The support of the fuzzy association rule  $I_1 \Rightarrow I_2$  in  $\tilde{T}$ ,  $Supp(I_1 \Rightarrow I_2)$ , is the evaluation of the quantified sentence  $Q_M$  of  $T$  are  $\tilde{\Gamma}_{I_1 \cup I_2} = Q$  of  $T$  are  $(\tilde{\Gamma}_{I_1} \cap \tilde{\Gamma}_{I_2})$ .
- The confidence of the fuzzy association rule  $I_1 \Rightarrow I_2$  in  $\tilde{T}$ ,  $Conf(I_1 \Rightarrow I_2)$ , is the evaluation of the quantified sentence  $Q$  of  $\tilde{\Gamma}_{I_1}$  are  $\tilde{\Gamma}_{I_2}$ .
- The certainty factor is obtained from support and confidence using equation 6.

We evaluate a quantified sentence of the form  $Q$  of  $F$  are  $G$  by means of method  $GD$  [7], obtaining the value  $GD_Q(G/F)$  as in Eq. 9:

$$\sum_{\alpha_i \in \Lambda(G/F)} (\alpha_i - \alpha_{i+1}) Q \left( \frac{|(G \cap F)_{\alpha_i}|}{|F_{\alpha_i}|} \right) \quad (9)$$

where  $\Delta(G/F) = \Lambda(G \cap F) \cup \Lambda(F)$ ,  $\Lambda(F)$  being the level set of  $F$ , and  $\Lambda(G/F) = \{\alpha_1, \dots, \alpha_p\}$  with  $\alpha_i > \alpha_{i+1}$  for every  $i \in \{1, \dots, p-1\}$ , and considering  $\alpha_{p+1} = 0$ . The set  $F$  is assumed to be normalized. If not,  $F$  is normalized and the same normalization factor is applied to  $G \cap F$ .

Let us remark that by using the quantifier  $Q_M$ , the measures described above yield the ordinary measures for support, confidence, and certainty factor in the crisp case. Hence, this approach reduces to ordinary association rules when applied on a set of crisp transactions, i.e., it is a consistent generalization of crisp association rules.

### 2.3 Approximate dependencies

Let  $r$  be a table with attributes  $X, Y, Z \dots$ . Approximate dependencies in  $r$  can be defined and assessed as association rules by introducing the following set of items and transactions [11]:

- Items:  $I_r = \{I_X, I_Y, I_Z \dots\}$ .
- Transactions:  $T_r$  obtained as follows: for every ordered pair of tuples  $(t, s)$  with  $t, s \in r$  there is one transaction  $ts \in T_r$ . An item of the form  $I_X$  is in the transaction associated to the pair  $(t, s)$  iff  $t[X] = s[X]$ .

Let us remark that these definitions do not correspond to the usual way of translating tables into transactions in order to obtain association rules (where items are pairs (attribute,value) and transactions correspond to tuples). The motivation of these alternative definitions is to obtain association rules of the

$$gt_{ts}([*, X, A]) = \begin{cases} |A(t[X]) - A(s[X])| & A(t[X]) * A(s[X]) \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

form  $I_X \Rightarrow I_Y$  with the semantics of Eq. 1, corresponding to approximate dependencies of the form  $X \rightarrow Y$ . This idea is employed as well in the definition of fuzzy approximate dependencies (see section 2.4) and fuzzy gradual rules (see section 2.5). Finally, support and certainty factor of  $I_X \Rightarrow I_Y$  are employed to assess the dependence  $X \rightarrow Y$  [11].

## 2.4 Fuzzy approximate dependencies

In [2], approximate dependencies as introduced in [11] are extended to fuzzy approximate dependencies by considering a set of fuzzy transactions. For this purpose, similarity relations  $S_X, S_Y, S_Z, \dots$  in the domain of attributes  $X, Y, Z, \dots$  respectively, are employed. Then, items and transactions are defined as follows:

- Items:  $I_r = \{[S_X, X], [S_Y, Y], \dots\}$ .
- Fuzzy transactions: a set of fuzzy transactions  $\widetilde{T}_r$  is obtained as follows: for every ordered pair of tuples  $(t, s)$  with  $t, s \in r$  there is one fuzzy transaction  $\widetilde{ts} \in \widetilde{T}_r$ . An item of the form  $[S_X, X]$  is in the fuzzy transaction associated to the pair  $(t, s)$  with degree  $\widetilde{ts}([S_X, X]) = S_X(t[X], s[X])$ .

With these definitions, a fuzzy association rule  $[S_X, X] \Rightarrow [S_Y, Y]$  in  $\widetilde{T}_r$  defines a fuzzy approximate dependence  $(S_X, X) \rightarrow (S_Y, Y)$  in  $r$ . Depending on the definition of fuzzy association rule, that vary mainly in the definition of the measures of support and accuracy, we can obtain different definitions of fuzzy approximate dependence. In [2] the definition in [5] (see section 2.2) is employed. Let us remark that in the crisp case (i.e., the similarity relation is crisp equality) these fuzzy approximate dependencies reduce to the crisp approximate dependencies introduced in [11] (section 2.3), considering  $I_X \equiv [=, X]$  and  $X \rightarrow Y \equiv [=, X) \rightarrow (=, Y)$ .

## 2.5 Fuzzy gradual dependencies

Following the ideas in previous sections, fuzzy gradual dependencies have been also defined in [10] as fuzzy association rules by providing definitions of items and fuzzy transactions in accordance with Eq. 2. Let  $A, B, C, \dots$  be fuzzy sets defined on the domain of  $X, Y, Z, \dots$ , respectively. Items and fuzzy transactions are defined as follows:

- Items: We define the set of items as  $GI_r = \{[*_1, X, A], [*_2, Y, B], [*_3, Z, C], \dots\}$   $\forall *_i \in \{<, >\}$ .
- Fuzzy transactions: a set of fuzzy transactions  $GT_r$  is obtained as follows: for every ordered pair of tuples  $(t, s)$  with  $t, s \in r$  there is one fuzzy transaction  $gt_{ts} \in GT_r$ . An item of the form  $[*_1, X, A]$  is in the fuzzy transaction associated to the pair  $(t, s)$  with degree given by Eq. 10.

With these definitions, a fuzzy association rule  $[*_1, X, A] \Rightarrow [*_2, Y, B]$  in  $GT_r$  defines a fuzzy gradual dependence  $(*_1, X, A) \rightarrow (*_2, Y, B)$  in  $r$ . Again, the definition in [5] (see section 2.2) is employed to define fuzzy gradual dependencies in [10], and the measures of support and certainty factor are employed in order to assess such dependencies.

## 3 New fuzzy dependencies

In this section we introduce new fuzzy dependencies whose semantics are a mixing of the semantics of fuzzy approximate dependencies and fuzzy gradual rules, extending and going beyond both.

### 3.1 Items

As seen in previous sections, it is possible to define different types of fuzzy dependencies as association rules by considering transactions associated to pairs of tuples. This is

common to all the dependencies we have described before. The main difference between fuzzy approximate dependencies (of which approximate dependencies are a particular case) and fuzzy gradual dependencies is the kind of items they consider.

We can obtain fuzzy dependencies with new semantics by adding more types of items and by mixing all the items in the set of fuzzy transactions. Given a pair of tuples  $(t, s)$ , we consider a fuzzy transaction  $ts$ . For every attribute  $X$ , let  $S_X$  and  $A$  be a similarity relation and a fuzzy set, respectively, both defined on the domain of  $X$ . We shall consider the following types of items:

- FAD-items: items of the form  $[S_X, X]$  employed to obtain fuzzy approximate dependencies.  $ts([S_X, X]) = S_X(t[X], s[X])$ .
- GD-items: items of the form  $[*, X, A]$  with  $* \in \{<, >\}$ .  $ts([*, X, A])$  is defined by Eq. 10.
- AR-items: items of the form  $[X, x]$  with  $x$  in the domain of  $X$ .  $ts([X, x])$  is defined by Eq. 11. This type of items correspond to the ordinary pairs (attribute,value) employed when mining for association rules in tables in the usual way.

$$ts([X, x]) = \begin{cases} 1 & t[X] = s[X] = x \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

### 3.2 New dependencies

Let  $r$  be a table with attributes  $X, Y, Z, \dots$ , let  $S_X, S_Y, S_Z, \dots, A, B, C, \dots$  be similarity relations and fuzzy subsets on the domain of the attributes<sup>3</sup>. On the basis of the items in the previous section, we can define fuzzy dependencies as follows:

**Definition 3.1** *Let  $T_r$  be a set of fuzzy transactions on a set of items  $I_r$  containing FAD-items, GD-items and AR-items. Let  $I_1, I_2 \subset$*

<sup>3</sup>In general, it is possible to consider more than a single similarity relation and more than a fuzzy set for each attribute.

$I$  with  $I_1 \cap I_2 = \emptyset$ . The fuzzy association rule  $I_1 \Rightarrow I_2$  in  $T_r$  defines a fuzzy dependence in  $r$ .

In practice, we shall consider that  $I_2$  contains one single item. Though in principle any kind of item is allowed both in  $I_1$  and  $I_2$ , not all the possible combinations have a clear semantics. This is the case, among others, in the following situations:

- $I_1 \cup I_2$  contains the itemset  $\{[X, x], [X, x']\}$  with  $x \neq x'$ . This is not allowed in tables under the first normal form. However, in databases that violate this normal form, those rules could make sense.
- $I_1 \cup I_2$  contains the itemset  $\{[X, x], [S_X, X]\}$  with  $S_X$  reflexive (usual in similarity relations). This is subsumed by the item  $\{[X, x]\}$ .
- $I_1 \cup I_2$  contains the itemset  $\{[<, X, A], [>, X, A]\}$ . The support of the rule is 0.
- $I_1 \cup I_2$  contains the itemset  $\{[X, x], [*, X, A]\}$  with  $* \in \{<, >\}$ . The support of the rule is 0.
- $I_1 \cup I_2$  contains the itemset  $\{[=, X], [*, X, A]\}$  with  $* \in \{<, >\}$ . The support of the rule is 0.

On the contrary, other combinations have a clear semantics. First of all, it is clear that approximate dependencies (section 2.3), fuzzy approximate dependencies (section 2.4) and fuzzy gradual dependencies (section 2.5) are particular cases of definition 3.1. In addition, new types of dependencies arise, in particular the following:

- Dependencies obtained by adding items of the form  $[Z, z]$  to the itemset  $I_1$  in fuzzy approximate dependencies and fuzzy gradual dependencies. The semantics of such dependencies is “the fuzzy approximate (resp. fuzzy gradual) dependence holds when  $Z = z$ ”.

- Dependencies where  $I_1$  contains only items of the form  $[*, X, A]$  or  $[S_X, X]$  and  $I_2$  is of the form  $[Z, z]$ .

These dependencies are useful in order to characterize what happens in those subsets of tuples where  $Z = z$ . For example, it could happen that a dependence  $[*_1, X, A] \rightarrow [*_2, Y, B]$  does not hold in a table, but holds in those tuples where  $Z = z$ , i.e., the dependence  $[Z, z], [*_1, X, A] \rightarrow [*_2, Y, B]$  holds. These dependencies are then restrictions of approximate and/or gradual properties in a certain subset of tuples of the original table.

## 4 Experiments

In this section we discuss first some algorithmic issues of mining fuzzy dependencies. Then we show some dependencies obtained from a database in order to illustrate the new types of dependencies introduced in this paper.

### 4.1 Algorithms

A straightforward way to obtain fuzzy dependencies from a table  $r$  according to definition 3.1 is to generate the set of fuzzy transactions  $T_r$  and then to apply any algorithm for mining association rules. The problematic cases of dependencies without semantics described in section 3.2 could be discarded either by modifying the association rule mining algorithm so that certain itemsets are not considered (more efficient), or discarding the problematic dependencies after they are obtained in a postprocessing stage.

The main drawback of this approach is that  $|T_r| = n^2$ ,  $n = |r|$ , and  $n^2$  can be a huge number. In order to diminish the complexity due to the number of transactions when mining for fuzzy approximate dependencies, an efficient algorithm was proposed in [2] as an extension of the algorithm proposed for crisp approximate dependencies in [11]. In this algorithm, the complexity due to the number of transactions is  $n$ , i.e., the same of an ordinary association rule mining algorithm; the same happens to the overall complexity in the

worst case (though, obviously, ordinary association rule mining  $r$  and approximate dependence mining in  $T_r$  are different tasks).

For mining fuzzy gradual dependencies, we propose to employ the algorithm 1. This algorithm considers a fixed number  $k$  of equidistributed membership degrees. Algorithm 1 calculates the support of an itemset of the form  $\{[*_1, X, A], [*_2, Y, B]\}$ . This algorithm can be recursively extended in order to calculate itemsets of larger size. The complexity of the extension for itemsets of size  $p$  is  $O(n + k^{(p+1)})$ . The goodness of this complexity depends on the relation between  $k^{(p+1)}$  and  $n^2$ . Typical values for  $k$  can be in the order of 20, though a lesser value of 10 can be sometimes acceptable. If we try to find gradual dependencies with at most two items in the antecedent, then  $p \leq 3$ . If we employ a number of levels  $k = 20$  then we have  $k^{(p+1)} \leq 20^4 = 16 \times 10^4$ , i.e., a complexity equivalent to mining using an ordinary association rule mining algorithm with  $n = 400$ . Since usually  $n \gg 400$  (and even  $n > 16 \times 10^4$ ), using our proposal is much better and we can consider reasonably a linear complexity in the number of tuples.

Let us remark that obtaining the support of items of the form  $[*, X, A]$  is the most complex task in general. Dependencies according to definition 3.1 where not all the items are of the form  $[*, X, A]$  is less complex; we shall deal with the task of mixing the algorithms for every type of item in a future paper.

Finally, the support of items of the form  $[X, x]$  is linear; specifically, if there are  $l$  tuples where  $t[X] = x$  (where  $l$  can be obtained in time  $O(n)$ ), then the support of  $[X, x]$  is  $l^2$ .

### 4.2 Results

We have used *Auto MPG* database from UCI Machine Learning Repository [1]. The data concerns city-cycle fuel consumption of cars in miles per gallon (MPG), and the dataset consists of 8 attributes (continuous and discrete) describing characteristics of cars and a total of 398 instances.

---

**Algorithm 1** Algorithm for 2-itemsets  $\{[*_1, X, A], [*_2, Y, B]\}$ .

---

```

 $\widetilde{Supp} \leftarrow \emptyset$ 
 $i \leftarrow k$ 
while  $i > 0$  do
   $elem \leftarrow 0$ 
   $j \leftarrow 0$ 
  while  $j \leq k - i$  do
     $m \leftarrow 0$ 
    while  $m \leq k - i$  do
       $elem \leftarrow elem + V[j + i][m + i] \times V[j][m]$ 
       $n \leftarrow i$ 
      while  $n \leq k$  do
        if  $n \neq j + i$  then
           $elem \leftarrow elem + V[n][m + i] \times V[j][m]$ 
        end if
         $n \leftarrow n + 1$ 
      end while
       $n \leftarrow i$ 
      while  $n \leq k$  do
        if  $n \neq j + i$  then
           $elem \leftarrow elem + V[j][m] \times V[j + i][n]$ 
        end if
         $n \leftarrow n + 1$ 
      end while
       $j \leftarrow j + 1$ 
    end while
     $i \leftarrow i + 1$ 
  end while
   $\widetilde{Supp} \leftarrow \widetilde{Supp} \cup \{i/k/elem\}$ 
   $i \leftarrow i - 1$ 
end while
return  $GD_Q(\widetilde{Supp}/\widetilde{GT}^D)$ 

```

---

In order to improve the knowledge discovery process, we defined a set of linguistic labels  $\{High, Medium, Low\}$  for continuous attributes by using *k-Means* clustering. Trapezoidal distributions are obtained by allowing an overlapping of 10% of the interval amplitude around the boundaries. As an example, the result for attribute *Acceleration* is shown in figure 1.

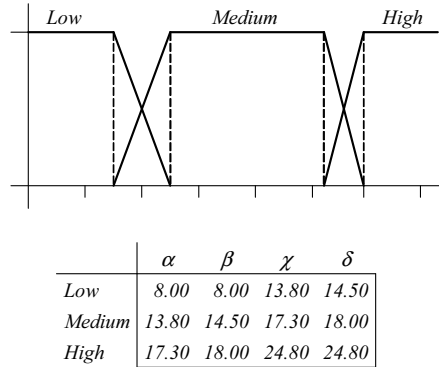


Figure 1: Linguistic sets defined over attribute *Acceleration*

Many dependencies have been found in this database. Table 1 shows a small sample of them. The first one is a pure fuzzy gradual dependence that can be expressed as “The less the Horsepower is High, the less the Displacement is High”. The other three are examples of fuzzy dependencies involving items of different types. They can be interpreted as gradual dependencies that hold for cars with a specific value of a certain attribute. Dependence 2 can be expressed as “For cars with Year=80, the less the Acceleration is Low, the less the MPG consumption is Low”. Dependence 3 means “For cars with Year=70, the less the Acceleration is High, the more the Horsepower is High”. Dependence 4 can be easily interpreted in a similar way.

## 5 Conclusions

We have introduced new types of fuzzy dependencies defined as association rules in a special set of transactions. The approach is based on previous work on the definition of fuzzy approximate dependencies and fuzzy gradual dependencies, in which transactions are asso-

Table 1: Some Fuzzy Dependencies from the Auto MPG dataset

#	Rule	s(%)	CF
1	$[\langle, \text{Horsepower}, \text{'high'}] \rightarrow [\langle, \text{Displacement}, \text{'high'}]$	16	0.94
2	$[\langle, \text{Acceleration}, \text{'low'}][\text{Year}, 80] \rightarrow [\langle, \text{MPG}, \text{'low'}]$	1.34	0.64
3	$[\langle, \text{Acceleration}, \text{'high'}][\text{Year}, 70] \rightarrow [\rangle, \text{Horsepower}, \text{'high'}]$	2	0.57
4	$[\langle, \text{Weight}, \text{'low'}][\text{Origin}, 1] \rightarrow [\langle, \text{Displacement}, \text{'low'}]$	18	0.84

ciated to pairs of tuples in a database, and items are defined according to the semantics of the dependencies. By incorporating different types of items in transactions at the same time, and looking for fuzzy association rules, dependencies with new semantics are obtained, i.e., new knowledge structures that can be extracted from a database.

As future work we plan to incorporate new types of items and to study the semantics of the new dependencies obtained by mixing items of different type. New types of items include items of the form  $[O_X, X]$ , with  $O_X$  a fuzzy order relation, and items of the form  $[X, A]$ , corresponding to fuzzy association rules. Also in the near future we shall study the dependencies obtained by using other measures for fuzzy association rules, like those proposed in [8].

### Acknowledgements

This work has been supported by the Spanish Ministerio de Educación y Ciencia under the project grant TIN2006-15041-C04-01.

### References

- [1] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.
- [2] F. Berzal, I. Blanco, D. Sánchez, J.M. Serrano, and M.A. Vila. A definition for fuzzy approximate dependencies. *Fuzzy Sets and Systems*, 149(1):105–129, 2005.
- [3] F. Berzal, I. Blanco, D. Sánchez, and M.A. Vila. Measuring the accuracy and interest of association rules: A new framework. *Intelligent Data Analysis*, 6:221–235, 2002.
- [4] P. Bosc, L. Lietard, and O. Pivert. Functional dependencies revisited under graduality and imprecision. In *Annual Meeting of NAFIPS*, pages 57–62, 1997.
- [5] M. Delgado, N. Marín, D. Sánchez, and M.A. Vila. Fuzzy association rules: General model and applications. *IEEE Transactions on Fuzzy Systems*, 11(2):214–225, 2003.
- [6] M. Delgado, D. Sánchez, and M.A. Vila. Fuzzy quantified dependencies in relational databases. In *Proc. of EUFIT'99*, 1999.
- [7] M. Delgado, D. Sánchez, and M.A. Vila. Fuzzy cardinality based evaluation of quantified sentences. *International Journal of Approximate Reasoning*, 23:23–66, 2000.
- [8] D. Dubois, E. Hüllermeier, and H. Prade. A systematic approach to the assessment of fuzzy association rules. *Data Mining and Knowledge Discovery*, 13(2):167–192, 2006.
- [9] E. Hüllermeier. Association rules for expressing gradual dependencies. In *Proceedings PKDD 2002 Lecture Notes in Computer Science 2431*, pages 200–211. 2002.
- [10] C. Molina, D. Sánchez, J.M. Serrano, and M.A. Vila. Mining gradual dependencies with variation strength. *Mathware and Soft Computing*, To Appear.
- [11] D. Sánchez, J.M. Serrano, I. Blanco, M.J. Martín-Bautista, and M.A. Vila. Using association rules to mine for strong approximate dependencies. *Data Mining & Knowledge Discovery*, To Appear.