# Interesting Fuzzy Association Rules in Quantitative Databases

Jeannette M. de Graaf, Walter A. Kosters, and Jeroen J.W. Witteman

Leiden Institute of Advanced Computer Science
Universiteit Leiden
P.O. Box 9512, 2300 RA Leiden, The Netherlands
{graaf,kosters,jwittema}@liacs.nl

**Abstract.** In this paper we examine association rules and their interestingness. Usually these rules are discussed in the world of basket analysis. Instead of customer data we now study the situation with data records of a more general but fixed nature, incorporating quantitative (non-boolean) data. We propose a method for finding interesting rules with the help of fuzzy techniques and taxonomies for the items/attributes. Experiments show that the use of the proposed interestingness measure substantially decreases the number of rules.

## 1 Introduction

In this paper we study association rules, i.e., rules such as "if a person buys products $a$ and $b$, then he or she also buys product $c$". Such a rule has a certain *support* (the number of records satisfying the rule, e.g., the number of people buying $a$, $b$ and $c$) and *confidence* (the fraction of records containing the items from the "then part" out of those containing the items from the "if part"). In most practical situations an enormous number of these rules, usually consisting of two or three items, is present. One of the major problems is to decide which of these rules are interesting.

Association rules are of particular interest in the case of basket analysis, but also when more general so-called quantitative or categorical data are considered, cf. [17]. Here one can think of augmented basket data, where information on the customer or time stamps are added, but also on more general fixed format databases. For example, one can examine a car database with information on price, maximum speed, horsepower, number of doors and so on. But also quite different databases can be used, for instance web-log files. So instead of products we shall rather speak of items or attributes, and buying product $a$ should be rephrased as having property $a$. We get rules like "if a car has four doors and is made in Europe, then it is expensive".

If we only consider the support of a rule, there is no emphasis on either "if part" or "then part", and in fact we rather examine the underlying *itemset*, in our first example $\{a, b, c\}$. A $k$-itemset consists of $k$ elements. Such a set is called *frequent* if its support is larger than some threshold, which is given in advance. In this paper we focus on the support rather than the confidence.

In the sequel we shall define a precise notion of interestingness, based on hierarchies with respect to the items. Using both simple real life data and more complicated real life data we illustrate the relevance of this notion. Our goal is to find a moderate number of association rules describing the system at hand, where uninteresting rules that can be derived from others are discarded. Interestingness of itemsets based on a hierarchy for the items is also discussed in [16], where for a one taxonomy situation a different notion of lifting to parents is used. Several other measures of interestingness for the non-fuzzy case not involving taxonomies are mentioned in [2, 3, 6, 10, 15] and references in these papers; for a nice overview see [9].

We would like to thank Jan Niestadt, Daniel Palomo van Es and the referees for their helpful comments.

## 2 Fuzzy approach

If one considers more general items/attributes, one has to deal with non-boolean values. Several approaches have been examined, each having its own merits and peculiarities. Two obvious methods are the usual boolean discretization (see [17]; note that this method suffers from the sharp boundary problem) and the fuzzy method. In this paper we focus on the fuzzy approach: split a non-boolean attribute into a (small) number of possible ranges called *fuzzified attributes*, and provide appropriate membership values (see [7, 12, 13]).

Some attributes naturally split into discrete values, for instance number of doors, giving a small number of crisp values. One can choose to add as many new items/attributes as there are values. It is also possible, in particular for two-valued attributes, to keep the boolean 0/1 notation. One has to keep in mind however that this gives rise to an asymmetry in the following sense: since only non-zero values will contribute, the rules found do not deal with "negative" information. For instance, if an attribute *Doors* has two possible values, 2 and 4, one can either split it into two new attributes *Doors2* and *Doors4* (notice that always exactly one of these will be true, so there is a clear negative dependency), or to keep only one attribute having the value 1 in the case of four doors; in this case rules with "having two doors" cannot easily be found.

An example for a more complex attribute is given in Figure 1, where the attribute *Horsepower* is fuzzified into four regions. We can now say that a record has property $a$ to a certain extent, e.g., a 68 *Horsepower* car has *Hp1* value 0.2 and *Hp2* value 0.8. In many situations the regions are chosen in such a way that for all values in the domain at most two membership values are non-zero. This approach is especially attractive, since it leads to hierarchies in a quite natural way: starting from basic ranges one can combine them into larger and larger ones in several ways, e.g., *Hp12* might be the union of *Hp1* and *Hp2*. Usually the membership values for the fuzzified attributes belonging to the same original attribute of a given record add to 1, as for the crisp case mentioned

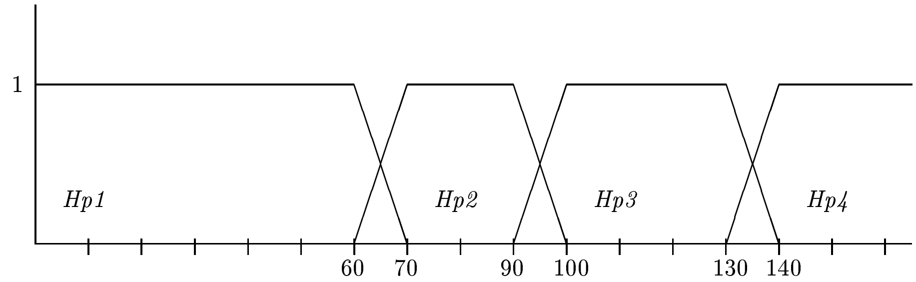


**Fig. 1.** Membership values for attribute *Horsepower*, split into four regions.

above. Note that the choice of the number of regions and the shape of the membership functions may be a difficult one. In this paper we use linear increase and decrease functions for the boundaries of the regions. The fuzzifications are obtained manually; it is also possible to apply clustering algorithms to determine clusters, and then use these as a basis for the regions.

In a fuzzy context the support of an itemset should be understood in the following way: for every record in the database take the *product* of the membership values (that can be crisp) of the attributes under consideration, and sum these numbers. If we have $n$ records, and if $\mu(i,j)$ denotes the membership value of the $j$-th (fuzzified) attribute of the $i$-th record, then the support of an itemset $A = \{a_{j_1}, a_{j_2}, \ldots, a_{j_k}\}$ is defined by

$$\mathit{support}(A) = \sum_{i=1}^{n} \prod_{\ell=1}^{k} \mu(i, j_\ell).$$

Notice that the usual 0/1 version is a special case. Here we mimic the well-known fuzzy AND: $\text{AND}(x,y) = x \cdot y$. Besides taking the product, there are also other possibilities for the fuzzy AND, for instance the often used minimum. The product however has a beneficial property, which is easily demonstrated with an example. Suppose that a car has *Hp1* value 0.2 and *Hp2* value 0.8 (the other *Hp* values being 0), and *Price1* value 0.4 and *Price2* value 0.6. Then it contributes to the combination $\{\mathit{Hp1}, \mathit{Price1}\}$ a value of $0.2 \cdot 0.4 = 0.08$, and similarly to the other three cross combinations values of 0.12, 0.32 and 0.48, respectively, the four of them adding to 1. The minimum would give 0.2, 0.2, 0.4 and 0.6, respectively, yielding a total contribution of $1.4 > 1$. In similar crisp situations every record of the database has a contribution of 1, and therefore we prefer the product.

Some simple example itemsets are $\{\mathit{Milk}, \mathit{Bread}\}$, $\{\mathit{Milk}, \mathit{TimeEarly}\}$ and $\{\mathit{Expensive}, \mathit{Europe}, \mathit{Doors4}\}$. Notice that the first one refers to the number of people buying both milk and bread, the second one measures the number of people buying milk early in the day (where *TimeEarly* is a fuzzified attribute), and the third one deals with the occurrence of expensive European cars having four doors in the current database.

In some situations it may occur that itemsets consisting of different "regions" of one and the same attribute have a somewhat high support, for instance the itemset $\{Hp1, Hp2\}$. This phenomenon indicates that many records lie in the intersection of these regions, and that the attribute needs to be fuzzified in yet another way.

## 3 Taxonomies

Now we suppose that a user defined *taxonomy* for the items is given, i.e., a categorization of the items/attributes is available. In this setting association rules may involve categories of attributes; abstraction from brands gives generalized rules, that are often more informative, intuitive and flexible. As mentioned before, also non-boolean attributes lead to natural hierarchies. Since the number of generated rules increases enormously, a notion of interestingness, cf. [8, 16], is necessary to describe them. It might for instance be informative to know that people often buy milk early in the day; on a more detailed level one might detect that people who buy low fat milk often do so between 11 and 12 o'clock. The more detailed rule is only of interest if it deviates substantially from what is expected from the more general one. It might also be possible to get more grip on the possible splittings of quantitative attributes, cf. [14].

A taxonomy is a hierarchy in the form of a tree, where the original items are the leaves, and the root is the "item" *All*; see [5] for the non-quantitative situation. The (internal) nodes of the taxonomies are sets of original items, these being singleton sets; every parent is the union of his or her children. In the case of fuzzy attributes, the fuzzy value of a parent is the sum of those from its children (assuming that this sum is at most 1), which corresponds to the fuzzy $\textsc{Or}$: $\textsc{Or}(x, y) = \min(1, x + y)$. For example, the *Hp12* value for a 68 *Horsepower* car case is $0.2 + 0.8 = 1.0$. One can also consider the case where several taxonomies are given. In this setting, an itemset is allowed to be any set of nodes from arbitrary levels from the taxonomies. Often we will restrict an itemset to belong to a single taxonomy. The root *All* is the set of all original items, and is the root of all taxonomies at hand.

A simple example of a taxonomy for a car database, with attributes *Price1*, *Price2*, *Doors2*, *Doors4*, *Hp1*, *Hp2*, *Hp3* and *Hp4*, and aggregates *Price12*, *Hp12* and *Hp34*, is presented in Figure 2.

## 4 Interestingness

An itemset (or rule) should be called *interesting* if it is in a way "special" with respect to what it is expected to be in the light of its parents. We first give some definitions concerning the connection between parent itemsets and their children.

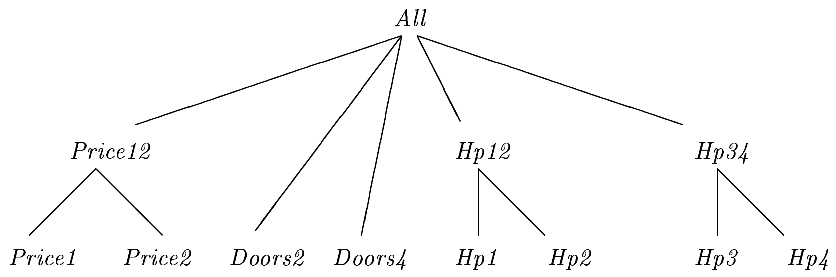


**Fig. 2.** Example—A simple taxonomy for a car database.

### 4.1 Definitions

A *first generation ancestor itemset* of a given itemset is created by replacing one or more of its elements by their immediate parents in the taxonomy. For the moment we choose to stay within one taxonomy, but it is also possible to use several taxonomies simultaneously. The only difference in that case is that elements can have more than one parent. The support of an ancestor itemset gives rise to a prediction of the support of the $k$-itemset $\mathcal{I} = \{a_1, a_2, \ldots, a_k\}$ itself: suppose that the nodes $a_1, a_2, \ldots, a_\ell$ $(1 \leq \ell \leq k)$ are replaced by (*lifted* to) their ancestors $\widehat{a_1}, \widehat{a_2}, \ldots, \widehat{a_\ell}$ (in general not necessarily their parents: an ancestor of $a$ is a node on the path from the root *All* to $a$, somewhere higher in the taxonomy) giving an itemset $\widehat{\mathcal{I}}$. Then the support of $\mathcal{I}$ is estimated by the support of $\widehat{\mathcal{I}}$ times the confidence of the rule "$\widehat{a_1}, \widehat{a_2}, \ldots, \widehat{a_\ell}$ implies $a_1, a_2, \ldots, a_\ell$":

$$\begin{aligned}&\mathit{EstimatedSupport}_{\widehat{\mathcal{I}}}(\{a_1, a_2, \ldots, a_\ell, a_{\ell+1}, \ldots, a_k\}) = \\ &\mathit{Support}(\{\widehat{a_1}, \widehat{a_2}, \ldots, \widehat{a_\ell}, a_{\ell+1} \ldots, a_k\}) \times \frac{\mathit{Support}(\{a_1, a_2, \ldots, a_\ell\})}{\mathit{Support}(\{\widehat{a_1}, \widehat{a_2}, \ldots, \widehat{a_\ell}\})}.\end{aligned}$$

This estimate is based on the assumption that given the occurrence of the lifted items $\{\widehat{a_1}, \widehat{a_2}, \ldots, \widehat{a_\ell}\}$, the occurrences of $\{a_1, a_2, \ldots, a_\ell\}$ and $\{a_{\ell+1}, a_{\ell+2}, \ldots, a_k\}$ are independent events, see [5]. In fact, this is a simple application of conditional probabilities: if

$$\begin{aligned}&P(\mathcal{I} \,|\, \widehat{a_1}, \widehat{a_2}, \ldots, \widehat{a_\ell}) = \\ &\quad P(a_1, a_2, \ldots, a_\ell \,|\, \widehat{a_1}, \widehat{a_2}, \ldots, \widehat{a_\ell}) \times P(a_{\ell+1}, \ldots, a_k \,|\, \widehat{a_1}, \widehat{a_2}, \ldots, \widehat{a_\ell}),\end{aligned}$$

then

$$\begin{aligned}P(\mathcal{I}) &= P(\widehat{a_1}, \widehat{a_2}, \ldots, \widehat{a_\ell}) \times P(\mathcal{I} \,|\, \widehat{a_1}, \widehat{a_2}, \ldots, \widehat{a_\ell}) \\ &= P(\widehat{\mathcal{I}}) \times P(a_1, a_2, \ldots, a_\ell \,|\, \widehat{a_1}, \widehat{a_2}, \ldots, \widehat{a_\ell}),\end{aligned}$$

where

$$P(a_1, a_2, \ldots, a_\ell \,|\, \widehat{a_1}, \widehat{a_2}, \ldots, \widehat{a_\ell}) = \frac{\mathit{Support}(\{a_1, a_2, \ldots, a_\ell\})}{\mathit{Support}(\{\widehat{a_1}, \widehat{a_2}, \ldots, \widehat{a_\ell}\})}.$$

Now an itemset is called *interesting* if and only if the predicted (fuzzy) supports based on all (but one as we shall see soon $(*)$) of its first generation ancestor itemsets deviate substantially from its real (fuzzy) support. If there is at least one parent that predicts the child suitably well, this itemset is not interesting enough. The word "substantially" means that the predicted supports are *all* larger than the real support, or are *all* smaller than the real support, by at least some fixed factor. This factor is called the *interestingness threshold*. If all items from an itemset are lifted, estimated support and real support are exactly the same, so it makes sense to omit this prediction (see $(*)$). Therefore 1-itemsets are always interesting, in particular the itemset $\{All\}$ (which does not have ancestors): there is no way to predict their support. In order to give a complete description of the "rule database" it is sufficient to describe the interesting rules: the behaviour of the others can then be derived—if one remembers which ancestor itemset provided the best prediction.

### 4.2 More Details

The reasons that only first generation ancestor itemsets are used instead of arbitrary ancestors as in [16] (where the number of items in the two itemsets should also be the same) are the following. First, it severely restricts the number of sets that need to be examined. (Note that in a single taxonomy a $k$-itemset has $2^k - 2$ first generation ancestor itemsets in principle, the number of arbitrary ancestors being much higher; $k$ is small in practice.) And second, if a set cannot be understood through any of its parents, but some grandparent does predict its support, in our opinion it still deserves attention.

Some problems arise during the lifting. In [5] the problem of several hierarchies (one item may contribute to several lifted ones at the same time, e.g., *Milk* is both *Dairy* and *Fluid*) is discussed. Another problem mentioned there is this: when lifting sibling attributes one gets itemsets of the form $\{Child, Parent\}$, e.g., $\{Milk, Dairy\}$. In [5] this was interpreted as the set $\{Milk\}$, since—logically speaking—buying milk is enough to satisfy both the milk and dairy requirements:

$$(\mathit{Milk}\ \text{AND}\ \mathit{Dairy}) = (\mathit{Milk}\ \text{AND}\ (\mathit{Milk}\ \text{OR}\ \ldots)) = \mathit{Milk}.$$

In the fuzzy approach the lifting of siblings from one and the same original attribute (which only happens in rare situations) is treated in an analogous manner, using fuzzy AND and OR. For example, suppose that a car has *Hp1* value 0.2 and *Hp2* value 0.6 (this differs from the situation in Figure 2 and in our experiments, where at most two membership values corresponding to one original attribute are non-zero), then its parent *Hp12* has value $0.2 + 0.6 = 0.8$, and its contribution to the itemset $\{Hp2, Hp12\}$ equals $0.6 \cdot (0.2 + 0.6) = 0.48$. Note that this is analogous to the situation for crisp boolean attributes: for boolean $x, y$ we have $x \wedge (y \vee x) = x$, leading to the interpretation mentioned in the beginning of this paragraph.

With respect to the partitioning of the attributes, either fuzzy or discrete, the notion of interestingness has yet another beneficial property. The support of an itemset may depend severely on the chosen partitioning. For instance, if a time period is split into periods of different sizes, the smaller ones will naturally have lower support. In the definition of *EstimatedSupport* the support of an itemset is estimated by the support of its parent multiplied by a factor that accounts for the relative size. The chosen partitioning is therefore of less importance than one would think at first sight, if interestingness is considered. But still it is necessary to carefully make this choice, since the domains should be split into discriminating understandable parts.

During experiments it sometimes occurred that itemsets containing some high supported item appeared to be interesting with respect to their first generation ancestor itemsets. From another point of view however, they might be considered not that interesting. For example, if an itemset $\{a\}$ has very high support, the itemsets $\{a, b, c\}$ and $\{b, c\}$ will probably have (nearly) the same support, and hence we feel that $\{a, b, c\}$ is not interesting. In general this phenomenon can be easily detected by checking whether the support of $\{a, b, c\}$ can be predicted through that of $\{b, c\}$. This corresponds to the situation where in the formula for the estimated support one particular item is lifted to the artificial item *All*. Because this easily computed extra interestingness measure improves the quality of the rules found, we added it in our experiments. This measure for interestingness is analogous to that in [3], where one or more items can be deleted from itemsets in order to check whether or not their support can be predicted from that of the smaller subsets. Finally, note that if in general one lifts to an attribute that is always 1, for instance the common ancestor of the different regions of fuzzified attributes, this corresponds to lifting to *All*.

## 5 Algorithms

The algorithms that find all interesting rules are straightforward. The well-known Apriori algorithm from [1], or any of its refinements, provides a list of all association rules, or rather the underlying itemsets. This algorithm can be easily adapted to generate all rules including nodes from the taxonomy (for more details, see [16]), where special care has to be taken to avoid parent-child problems, and to the fuzzy situation (see [12]). Note that Apriori works under the assumption that the support of a subset is always at least the support of any superset, which also holds in this generalized setting (all fuzzy membership values are at most 1 and we use multiplication as fuzzy And; by the way, the frequently used minimum can also be chosen). In fact, if one augments the list of original items with all non-leaves from the taxonomy, the computations are straightforward. Once the list of all rules and their supports is known, it is easy to generate the interesting ones by just comparing supports for the appropriate rules. The order in which the computations are performed, is of no importance.

For every frequent itemset $\mathcal{I}$ all its first generation ancestor itemsets $\widehat{\mathcal{I}}$ are generated, and expected and real support are compared; we define the *support deviation* of $\mathcal{I}$ to be the smallest *interestingness ratio*

$$\mathit{Support}(\mathcal{I}) \ / \ \mathit{EstimatedSupport}_{\widehat{\mathcal{I}}}(\mathcal{I})$$

that occurs. If this support deviation is higher than the interestingness threshold, the itemset is called interesting. The frequent itemsets can be ordered with respect to this support deviation: the higher this ratio, the more interconnection occurs between the items involved. In fact, the assumption concerning the independence between lifted and non-lifted items clearly does not hold in that case, and an interesting connection is revealed. Of course it is also a possibility to look at overestimated supports—in many cases they are "complementary" to the underestimated ones. If necessary, the confidence can be used to turn the list of interesting itemsets into a list of interesting rules, further decreasing the number of interesting rules. Note that ancestors of frequent itemsets are automatically frequent, unless—as in [8]—different support thresholds are specified at different tree levels (if, e.g., $\{Milk, Bread\}$ is frequent, $\{Dairy, Bread\}$ should be frequent too in order to compute the support deviation).

The run time of the algorithms may—as usual—be long when the number of records is large and the minimum support threshold is low. In order to also get information on the bottom level, and not only on aggregate levels, this minimum support should be small enough. A run time of several hours was quite normal, most of it devoted to the computation of the frequent itemsets using the APRIORI algorithm. Once the rules/itemsets are computed, it is however easy to deal with different interestingness thresholds. This is an advantage over methods that detect interestingness during the computation of the frequent itemsets (cf. [3], where no taxonomies are used).

## 6 Experiments

In order to get a feeling for the ideas, we first present some details for a simple database consisting of descriptions of 205 cars, see [4]. We have clearly dependent attributes like *Price*, *MilesPerGallon*, *EngineSize* and *Horsepower* (an integer between 48 and 288, the median being 96). This last attribute may be fuzzified as in Figure 1, where it is split into four regions, denoted by *Hp1*, *Hp2*, *Hp3* and *Hp4*. One might choose the regions in such a way that they all contain the same number of records—more or less (option **1**). Another option is to split the region simply into four equally large intervals (option **2**). We also examined a random fuzzification (option **3**). Of course there is quite a lot of freedom here, but an advantage of the fuzzy method is that slight changes do not lead to major differences (see, e.g., [12] for a comparison between crisp case and fuzzy case); as mentioned above, the interestingness corrects for different splittings to a certain extent. At aggregate levels we defined *Hp12* and *Hp34* as the "sum" of the first two regions, respectively the last two. In a similar way we also fuzzified the other attributes, all of them having four regions, region 1 corresponding to a low value, and so on. Furthermore we added the attributes *Doors2*, *Doors4* and *Turbo*, the last one being a boolean attribute.

Clear dependencies were easily detected in all cases, such as *Price4* and *Hp4* (more than expected), *Price4* and *MilesPerGallon4* (less than expected, but still enough to meet the support threshold), and *Price4* and *MilesPerGallon1* (more than expected). But also itemsets like $\{Hp1, Price1, Doors2\}$ were found to be interesting for option **1**: all its eight parents (including those obtained by omitting an item) caused an interestingness ratio above 1.3. In Figure 3 some results with respect to the number of rules are presented. The itemset $\{Turbo, Hp34, Price34\}$ had support deviation 1.61, indicating that turbo engine cars occur quite often among cars with high values for *Horsepower* and *Price*; but it also means that among expensive turbo engine cars those with a high value for *Horsepower* occur more than expected. The support threshold was chosen to be 10%. Here the notation 22 / 137 means that 22 out of 137 itemsets are interesting. Note that option **2** leads to only 17 frequent 1-itemsets, due to the irregular distribution of the records over the equally sized intervals for the fuzzified attributes. We may conclude that a substantial reduction in the number of itemsets is obtained.

| option for fuzzification | threshold for support deviation | 1-itemsets | 2-itemsets | 3-itemsets | 4-itemsets |
|---|---|---|---|---|---|
| **1** | 1.3 | 27 / 27 | 34 / 137 | 24 / 185 | 14 / 103 |
| | 1.4 | 27 / 27 | 28 / 137 | 14 / 185 | 5 / 103 |
| | 1.5 | 27 / 27 | 22 / 137 | 11 / 185 | 4 / 103 |
| | 1.6 | 27 / 27 | 15 / 137 | 11 / 185 | 3 / 103 |
| | 1.7 | 27 / 27 | 9 / 137 | 4 / 185 | 1 / 103 |
| | 1.8 | 27 / 27 | 4 / 137 | 1 / 185 | 0 / 103 |
| | 1.9 | 27 / 27 | 4 / 137 | 0 / 185 | 0 / 103 |
| | 2.0 | 27 / 27 | 4 / 137 | 0 / 185 | 0 / 103 |
| **2** | 1.3 | 17 / 17 | 15 / 87 | 7 / 173 | 2 / 148 |
| | 1.4 | 17 / 17 | 13 / 87 | 5 / 173 | 1 / 148 |
| | 1.5 | 17 / 17 | 10 / 87 | 4 / 173 | 1 / 148 |
| | 1.6 | 17 / 17 | 9 / 87 | 4 / 173 | 1 / 148 |
| | 1.7 | 17 / 17 | 8 / 87 | 4 / 173 | 1 / 148 |
| | 1.8 | 17 / 17 | 7 / 87 | 2 / 173 | 1 / 148 |
| | 1.9 | 17 / 17 | 5 / 87 | 2 / 173 | 1 / 148 |
| | 2.0 | 17 / 17 | 4 / 87 | 2 / 173 | 1 / 148 |
| **3** | 1.3 | 24 / 24 | 20 / 109 | 12 / 162 | 4 / 108 |
| | 1.4 | 24 / 24 | 14 / 109 | 5 / 162 | 1 / 108 |
| | 1.5 | 24 / 24 | 13 / 109 | 5 / 162 | 1 / 108 |
| | 1.6 | 24 / 24 | 12 / 109 | 5 / 162 | 1 / 108 |
| | 1.7 | 24 / 24 | 10 / 109 | 5 / 162 | 1 / 108 |
| | 1.8 | 24 / 24 | 8 / 109 | 5 / 162 | 1 / 108 |
| | 1.9 | 24 / 24 | 7 / 109 | 4 / 162 | 1 / 108 |
| | 2.0 | 24 / 24 | 4 / 109 | 1 / 162 | 0 / 108 |

**Fig. 3.** Car database: number of interesting itemsets out of all frequent itemsets, for different fuzzifications and thresholds for the support deviation.

Next we considered a much larger database, obtained from product and sales information from supermarket chains. For every product, and every time period, and for every chain, the number of sales is given—among other things. We restricted ourselves to one chain. The database consisted of 158,301 records, giving sales for 4,059 products over a period of three years (split into 39 periods).

We took minimum support 1%, leading to 163 frequent 1-itemsets, 378 frequent 2-itemsets and 102 frequent 3-itemsets. With a small interestingness threshold of 1.01, 162 2-itemsets were found to be interesting, and 46 3-itemsets. As in the previous example, some obvious itemsets were found quite easily. For example, $\{BrandX, SmallBag\}$ and $\{Mayonnaise, Jar\}$ were above expectation, with support deviations 3.50 and 2.85, respectively. Here *Mayonnaise* denotes a group of mayonnaise-like products, and *BrandX* consists of instant soups and sauces of a certain brand. The package clearly depends on the contents. Some interesting 3-itemsets were also discovered, for example $\{BBQSauce, Bottle, Chili\}$ with support deviation 5.89. Apparently Chili taste in a bottle is even more frequent among other BBQ sauces.

It was much harder to find interesting itemsets containing time information, because the support of itemsets containing for example *Month4* were much smaller by nature than the ones mentioned in the previous paragraph. If one only examines the records corresponding to one category of products, for instance the BBQ sauces (for our database 21,450 records), it is possible to detect small differences in sales throughout the year. It appeared that in the third quarter of the year high sales were more frequent than expected, whereas low sales were more frequent than expected in the first quarter of the year.

Two important problems that arose are the following. Due to the fact that there were very many missing values in the database at hand, for some attributes it was hard to find a proper interpretation for the results. For the moment we chose not to skip the complete record; we ignored the missing fields when generating the frequent itemsets containing these fields. The second problem has to do with the fuzzifying process. In the database the number of products sold during some period in some supermarket chain is given. If one wants to fuzzify this number, one clearly has to take into account that notions like "many" or "few" severely depend on the product and the shop at hand. The usual data mining step that cleans the data has to be augmented with a process that handles this problem. For the current experiment we simply took global values, which seems justified because we deal with only one supermarket chain or even one category.

## 7 Conclusions and Further Research

We have presented a notion of interestingness for frequent itemsets in general fixed format databases with both quantitative, categorical and boolean attributes, using fuzzy techniques. Examples show that the number of itemsets found decreases substantially when restricted to interesting ones. It is in principle also possible to handle important attributes like time.

We would like to study this time dependency of itemsets further, for instance using time windows. It should also be possible to use the notion of interestingness for clustering techniques, cf. [11]. Other research issues are the handling of missing values and different fuzzifications. If for instance both price and sales attributes in a customer database are missing quite often, this might lead to an underestimated value for the support of itemsets containing both price and sales attributes. We would like to handle these kinds of problems, both in theoretical and practical respect.

Another problem is the following: it is sometimes necessary to split one and the same fuzzy attribute (like the number of sales in the second experiment) differently, depending on the record or the group of records. For example a sales of 1,000 may be "many" for *BrandX* but "few" for *BrandY*. It would be interesting to study different possibilities here, especially for practical situations. Finally we would like to get a better understanding of the missing value problem.

## References


1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo. Fast discovery of association rules. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI/MIT Press, 1996.
2. R.J. Bayardo Jr. and R. Agrawal. Mining the most interesting rules. In S. Chaudhuri and D. Madigan, editors, *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 145–154. ACM Press, 1999.
3. R.J. Bayardo Jr., R. Agrawal, and D. Gunopulos. Constraint-based rule mining in large, dense databases. *Data Mining and Knowledge Discovery*, 4:217–240, 2000.
4. 1985 Auto Imports Database. Available at `http://www.ics.uci.edu/~mlearn/`.
5. J.M. de Graaf, W.A. Kosters, and J.J.W. Witteman. Interesting association rules in multiple taxonomies. In A. van den Bosch and H. Weigand, editors, *Proceedings of the Twelfth Belgium-Netherlands Artificial Intelligence Conference (BNAIC'00)*, pages 93–100, 2000.
6. A.A. Freitas. On objective measures of rule surprisingness. In J.M. Żytkov and A. Quafafou, editors, *Principles of Data Mining and Knowledge Discovery, Proceedings of the 2nd European Symposium (PKDD'98)*, Springer Lecture Notes in Computer Science 1510. Springer Verlag, 1998.
7. A. Fu, M. Wong, S. Sze, W. Wong, W. Wong, and W. Yu. Finding fuzzy sets for the mining of fuzzy association rules for numerical attributes. In *Proceedings of the First International Symposium on Intelligent Data Engineering and Learning (IDEAL'98)*, pages 263–268, 1998.
8. J. Han and Y. Fu. Mining multiple-level association rules in large databases. *IEEE Transactions on Knowledge and Data Engineering*, 11:798–804, 1999.
9. R.J. Hilderman and H.J. Hamilton. Heuristic measures of interestingness. In J. Żytkov and J. Rauch, editors, *Proceedings of the 3rd European Conference on the Priciples of Data Mining and Knowledge Discovery (PKDD'99)*, pages 232–241, 1999.
10. M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A.I. Verkamo. Finding interesting rules from large sets of discovered association rules. In *Proceedings*

*of the Third International Conference on Information and Knowledge Management (CIKM'94)*, pages 401–407. ACM Press, 1994.
11. W.A. Kosters, E. Marchiori, and A. Oerlemans. Mining clusters with association rules. In D.J. Hand, J.N. Kok, and M.R. Berthold, editors, *Proceedings of the Third Symposium on Intelligent Data Analysis (IDA99)*, Springer Lecture Notes in Computer Science 1642, pages 39–50. Springer Verlag, 1999.
12. C. Kuok, A. Fu, and M. Wong. Mining fuzzy association rules in databases. *ACM SIGMOD Record*, 27:41–46, 1998.
13. J.-H. Lee and H. Lee-Kwang. An extension of association rules using fuzzy sets. In *The Seventh International Fuzzy Systems Association World Congress (IFSA'97)*, pages 399–402, 1997.
14. M.C. Ludl and G. Widmer. Relative unsupervised discretization for association rule mining. In D.A. Zighed, J. Komorowski, and J. Żytkov, editors, *Principles of Data Mining and Knowledge Discovery, Proceedings of the 4th European Conference (PKDD 2000)*, Springer Lecture Notes in Computer Science 1910, pages 148–158. Springer Verlag, 2000.
15. G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In G. Piatetsky-Shapiro and W.J. Frawley, editors, *Knowledge Discovery and Data Mining*, pages 229–248. MIT Press, 1991.
16. R. Srikant and R. Agrawal. Mining generalized association rules. In U. Dayal, P.M.D. Gray, and S. Nishio, editors, *Proceedings of the 21st VLDB Conference*, pages 407–419. Morgan Kaufmann, 1995.
17. R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In H. V. Jagadish and I.S. Mumick, editors, *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 1–12, Montreal, Quebec, Canada, 1996.