A Fuzzy Approach for Mining Quantitative Association Rules

Attila Gyenesei

Turku Centre for Computer Science

TUCS Technical Reports

No 336, March 2000

A Fuzzy Approach for Mining Quantitative Association Rules

Attila Gyenesei

University of Turku, Department of Computer Science,
Lemminkisenkatu 14, FIN-20520 Turku, Finland

Turku Centre for Computer Science
TUCS Technical Report No 336
March 2000

ISBN 952-12-0634-9
ISSN 1239-1891

Abstract

During the last ten years, data mining, also known as knowledge discovery in databases, has established its position as a prominent and important research area. Mining association rules is one of the important research problems in data mining. Many algorithms have been proposed to find association rules in databases with binary attributes. In this paper, we deal with the problem of mining association rules in databases containing both quantitative and categorical attributes. An example of such an association might be "10% of married people between age 50 and 70 have at least 2 cars".

We introduce a new definition of quantitative association rules based on fuzzy set theory. Using the fuzzy set concept, the discovered rules are more understandable to a human. Moreover, fuzzy sets handle numerical values better than existing methods because fuzzy sets soften the effect of sharp boundaries. The above example could be rephrased eg. 10% of married old people have several cars.

In this paper we present a new algorithm for mining fuzzy quantitative association rules. The algorithm uses new definitions for interesting measures. Experimental results show the efficiency of the algorithm for large databases.

Keywords: data mining, association rules, basket data, support, confidence, linguistic terms, interesting measure

TUCS Research Group
Algorithmics Research Group

1 Introduction

During the last ten years, data mining, also known as knowledge discovery in databases, has established its position as a prominent and important research area. Mining association rules in one of the important research problems in data mining [Piatesky et al.,1991]. The problem of mining boolean association rules over basket data was introduced in [Agrawal et al.,1993]. Given a set of transactions where each transaction is a set of items, an association rule is an expression of the form $X \Rightarrow Y$, where X and Y are sets of items. An example of an association rule is: "40% of transactions that contain beer and potato chips also contain diapers; 5% of all transactions contain all of these items". Here 40% is called the confidence of the rule, and 5% the support of the rule. The problem is to find all association rules that satisfy user-specified minimum support and minimum confidence constraints. There are many known algorithms for mining boolean association rules. For example, [Srikant et al.,1994] has proposed Apriori, AprioriTID and AprioriHybrid algorithms for mining association rule. Moreover, a hash-based algorithm for mining association rule has been introduced in [Park et al.,1995], and a binary tree algorithm in [Gyenesei,1999].

Recently, people have got interested in quantitative attributes. The problem of mining quantitative association rules is introduced in [Srikant et al.,1996]. For illustration, a table with three non-key attributes is shown in Table 1. Age and NumCars are quantitative attributes, whereas Married is a categorical attributes. For example, a quantitative association rule present in this table is: $\langle Age : 30..39\rangle$ and $\langle Married : Yes\rangle \Rightarrow \langle NumCars : 2\rangle$.

RecordID	**Age**	**Married**	**NumCars**
5	23	No	1
8	25	Yes	1
14	27	No	0
18	38	Yes	2
19	39	Yes	2

(min. support = 40%, min. confidence = 50%)

Rules (Sample)	**Support**	**Confidence**
$\langle Age : 30..39\rangle$ and $\langle Married : Yes\rangle \Rightarrow \langle NumCars : 2\rangle$	40%	100%
$\langle NumCars : 0..1\rangle \Rightarrow \langle Married : No\rangle$	40%	66.6%

Table 1: Example of Quantitative Association Rules

In [Srikant et al.,1996], an algorithm has been proposed. The algorithm finds the association rules by partitioning the attribute domain, combining adjacent partitions, and then transforming the problem into binary one. Several efficient algorithms for mining categorical association rules have been published (see [Mannila et al.,1994], [Toivonen,1996] for just a few examples).

Although current quantitative association rule mining algorithms can solve some problems introduced by quantitative attributes, they introduce some other problems. The first problem is caused by the sharp boundary between intervals. The algorithms either ignore or over-emphasize the elements near the boundary of the intervals in the mining process. The use of shape boundary intervals is also not intuitive with respect to human perception. For example, the interval method may classify a person as young if age is less than 35 and old if age is greater than 35. This obviously does not correspond to the human perception of young and old. Furthermore, it is not easy to distinguish the degree of membership for the interval method. For instance, using the interval method, age of 40 and age of 80 will both be classified into old. However, we intuitively know that age of 80 is much older than age of 40.

In this paper, we deal with mining fuzzy association rules of the form, If X is A then Y is B. X, Y are attributes and A, B are fuzzy sets which characterize X and Y respectively. The fuzzy set concept is better than the partition method because fuzzy sets provide a smooth transition between member and non-member of a set. Because of the smooth transition, there are fewer boundary elements being excuded. Moreover, the fuzzy association rule is more understandable because of linguistic terms associated with fuzzy sets.

This paper is organized as follows. In the following section, we will describe different ways to handle quantitative attributes. Then we will give the definition of fuzzy association rules and interest measures of itemsets and rules in section 3. In section 4 we show a new algorithm for fuzzy quantitative association rules. In section 5 the experimental results will be given. We will give a brief conclusion in section 6.

2 Quantitative Attributes

In [Agrawal et al.,1993], [Fayyad et al.,1994], [Sarasere et al.,1995] algorithms to find binary association rules in large databases have been proposed. However, attributes can be quantitative (e.g. age, income) or categirocal (e.g. zip code, make of car). Boolean attributes can be considered a special case of categorical attributes. Since we cannot directly apply the binary algorithms, we either have to transform the quantitative problem into binary one or to find new algorithms.

2.1 Mapping quantitative rules to Boolean rules

If the Quantitative Rules problem can be mapped to the Boolean Rules problem, any algorithm for finding Boolean Association Rules can be used to find quantitative association rules. This mapping can be performed as follows:
Let the relational table contain a boolean field for each attribute value/interval for each quantitative attribute. Then the value of any such Boolean field, which corresponds to $\langle attribute1, value1 \rangle$, would be "1" if the $attribute1$ had $value1$ in the original record, and "0" otherwise. Table 2 shows this mapping for the People table given in Table 1. Age is partitioned into two intervals: 20..29 and 30..39. The categorical attribute, Married, has two boolean attributes: "No" and "Yes".

RID	**Age: 20..29**	**Age: 30..39**	**Married: Yes**	**Married: No**	**NumCars: 0**	**NumCars: 1**	**NumCars: 2**
5	1	0	0	1	0	1	0
8	1	0	1	0	0	1	0
14	1	0	0	1	1	0	0
18	0	1	1	0	0	0	1
19	0	1	1	0	0	0	1

Table 2: Mapping to Boolean Association Rules Problem

Since the number of values for NumCars is small, NumCars is not partitioned into intervals; each value is mapped to a boolean field. Record 14, which had $\langle Age : 29 \rangle$ now has "Age : 20..29" equal to "1", "Age : 30..39" equal to "0", etc.

2.2 Mapping Problems

Unfortunately, a "sharp boundary problem" results from using interval partitions. For example, as shown in Figure 1, suppose $[1, 5]$ and $[6, 10]$ are two intervals created on quantitative attribute. If the minimum support threshold is greater than 40%, the interval $[6, 10]$ will not gain enough support regardless of the large support near its left boundary.

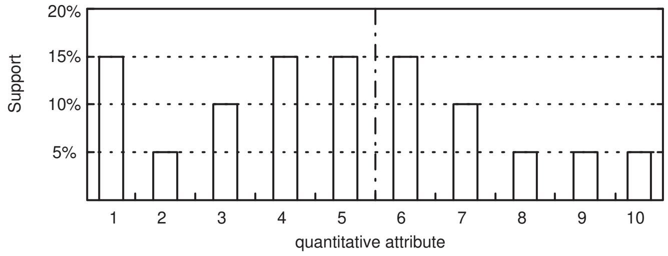

Figure 1: Example of Sharp Boundary problem

Figure 2 also shows the effect of sharp boundary for the database given in Table 3. The graph is the data distribution of *age*. The attribute domain

RID	**Age**	**Salary**	**RID**	**Age**	**Salary**
1	19	1400	11	26	2000
2	22	1600	12	31	2400
3	31	2400	13	19	1400
4	18	1400	14	27	2200
5	23	1600	15	31	2600
6	30	2800	16	15	1000
7	17	1200	17	24	1800
8	25	2000	18	38	2600
9	31	2200	19	17	1200
10	19	1400	20	39	2400

Table 3: An Example - Database

has been partitioned into 3 intervals. Suppose the intervals, 11 to 20, 21 to 30 and 31 to 40, only have 35% – 35% – 30% support and the minimum

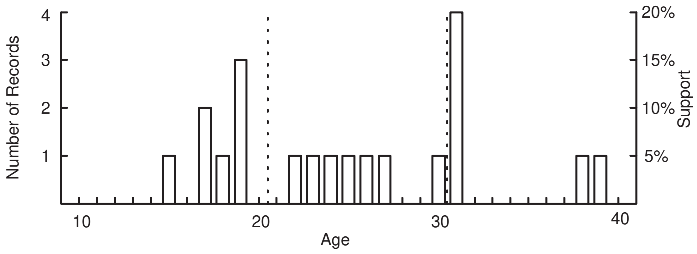

Figure 2: Discrete Intervals

support is 40%. In this case, none of these intervals will have enough support. However, the interval, 20 to 30, should be interesting if we consider the values near both sides.

Another attribute partitioning method [Srikant et al.,1996] is to divide the attribute domain into overlapped regions and is shown in Figure 3. We can see that the boundaries of intervals are overlapped with each other. As a result, the elements located near the boundary will contribute to more than one interval such that some intervals may become interesting in this case. It is, however, not reasonable for an element near the boundaries to contribute the same degree as those located within an interval. This will surely overemphasize the importance of an interval.

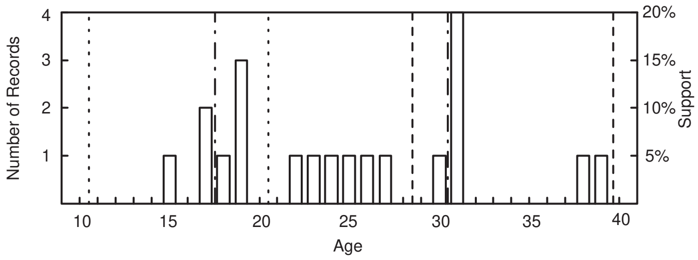

Figure 3: Overlapped Intervals

2.3 Our Approach

The above attribute partitioning methods are subject to the effect of sharp boundaries because of the classical set theory. In the fuzzy set theory, however, an element can belong to a set with set membership value in $[0, 1]$. This value is assigned by the membership function associated with each fuzzy set. For attribute x and its domain dom_x, the mapping of the membership function is $m_{fx}(x) : dom_x \rightarrow [0, 1]$, where f_x is a fuzzy set.

The fuzzy set provides a smooth change over the boundary and the effect is shown in Figure 4. The second graph shows the curve of a traditional fuzzy set and we can see that the values located outside the interval have been considered. Therefore, the sharp boundary problem has been tackled. Moreover, the contribution of a value has been restricted by the membership function as illustrated in Figure 4.

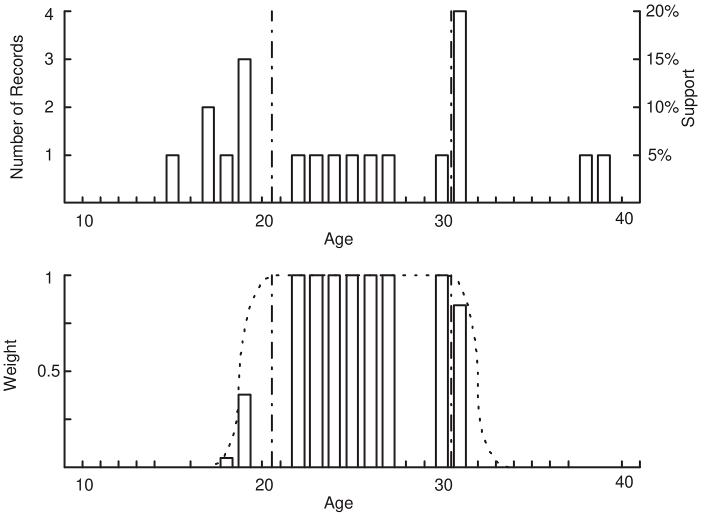

Figure 4: Fuzzy set

In Figure 1, the two intervals will be replaced by two fuzzy sets. Suppose the value 5 has membership degree of 0.9 in the first set and 0.3 in the second set. Then it will contribute 0.9 to the support of the first fuzzy set and 0.3 to the second one. However, this means that the value 5 will be more important than other values since the sum of its contributions to different fuzzy sets has become greater than 1. In the following section we will introduce an additional normalization process for this problem, but we show in the experimental results that this process reduces the number of frequent itemsets.

3 Problem Definition

Mining fuzzy association rules is the discovery of association rules using fuzzy set concepts such that the quantitative attributes can be handled. In this section we will introduce a fuzzy normalization process first. Then we will give the definition of a fuzzy association rule and discuss the interest measures of itemsets and the rules.

3.1 Fuzzy Normalization Process

Let $I = \{i_1, i_2, \ldots, i_m\}$ be the complete item set where each i_j $(1 \leq j \leq m)$ denotes a categorical or quantitative (fuzzy) attribute. Suppose $f(i_j)$ represents the maximum number of categories (if i_j is categorical) or the maximum number of fuzzy sets (if i_j is fuzzy), and $m_{i_j}(l,v)$ represents the membership degree of v in the l^{th} category or fuzzy set of i_j. If i_j is categorical, $m_{i_j}(l,v) = \{0,1\}$. If i_j is fuzzy, $0 \leq m_{i_j}(l,v) \leq 1$.

Let $t = \{t.i_1, t.i_2, \ldots, t.i_m\}$ be a transaction, where $t.i_j$, $(1 \leq j \leq m)$ represents a value of the j^{th} attribute and can be mapped to

$$\left\{ \left(l, m_{i_j}\left(l, t.i_j\right)\right) \mid \text{ for all } l, 1 \leq j \leq f(i_j) \right\}.$$

If i_j is fuzzy, $\sum_{l=1}^{f(i_j)} m_{i_j}(l, t.i_j)$ does not always equal to 1. We have developed a normalization process as follows:

$$m'_{i_j}(l, t.i_j) = \begin{cases} \frac{m_{i_j}(l,t.i_j)}{\sum_{l=1}^{f(i_j)} m_{i_j}(l,t.i_j)} & \text{if } i_j \text{ is fuzzy,} \\ m_{i_j}(l, t.i_j) & \text{if } i_j \text{ is categorical.} \end{cases}$$

Example. Suppose $I = \{level, age\}$ where *level* is a categorical attribute with the domain of {freshman, sophomore, junior, senior, graduate} and *age* is a quantitative attribute with three fuzzy sets {young, middle, old}. Note that it is possible to define other fuzzy set groups for this attribute.
A T = {graduate, 25} will be mapped to {{(graduate, 1)}, {(young, 0.2), (middle, 0.9), (old, 0.1)}}.

$\langle Level, senior\rangle$	$\langle Level, graduate\rangle$	$\langle Age, young\rangle$	$\langle Age, middle\rangle$	$\langle Age, old\rangle$
0	1	0.2	0.9	0.1

Table 4: Without Fuzzy Normalization

Without normalization (Table 4), it would increase the support of itemset $\{level = graduate, age = young\}$ by 0.2, the support of itemset $\{level = graduate, age = middle\}$ by 0.9, and the support of itemset $\{level = graduate, age = old\}$ by 0.1. That is to say, this transaction will be counted 0.2+0.9+0.1=1.2 times for the item *age*. However, it is unreasonable for one transaction to contribute more than others, if the corresponding discrete sets are disjoint.

$\langle Level, senior\rangle$	$\langle Level, graduate\rangle$	$\langle Age, young\rangle$	$\langle Age, middle\rangle$	$\langle Age, old\rangle$
0	1	0.167	0.75	0.083

Table 5: After Fuzzy Normalization

In contrast (Table 5), the normalization process will further transform the transaction T into {{(graduate, 1)}, {(young, 0.167), (middle, 0.75), (old, 0.083)}}, for a total contribution of 1.0 for the item *age*.

3.2 Fuzzy Association Rule

Given a database $D = \{t_1, t_2, \ldots, t_n\}$ with attributes I and the fuzzy sets associated with attributes in I, we want to find out some interesting, potentially useful regularities. We use the following form for fuzzy association rule:

$$\text{If } X = \{x_1, x_2, \ldots, x_p\} \text{ is } A = \{f_1, f_2, \ldots, f_p\} \text{ then } Y = \{y_1, y_2, \ldots, y_q\} \text{ is } B = \{g_1, g_2, \ldots, g_q\},$$

where

$$f_i \in \{\text{fuzzy sets related to attribute } x_i\},$$
$$g_j \in \{\text{fuzzy sets related to attribute } y_j\}$$

and X, Y are itemsets. X and Y are subsets of I and they are disjoint which means that they share no common attributes. A and B contain the fuzzy sets associated with the corresponding attributes in X and Y. As in the binary association rule, "X is A" is called the antecedent of the rule while "Y is B" is called the consequent of the rule.

If a rule is interesting, it should have enough support and high confidence value. We also use a correlation value to determine the satisfiability of rules.

3.2.1 Fuzzy Support Value

To generate fuzzy association rules, we have first to find out all sets of items (itemsets) that have transaction support above a user specified threshold. Itemsets with at least a minimum support are called *frequent itemsets*. The fuzzy support value is calculated by first summing all votes of each record with respect to the specified itemset, then dividing it by the total number of records. Each record contributes a vote which falls in $[0, 1]$. Therefore, a fuzzy support value reflects not only the number of records supporting the itemset, but also their degree of support. We use the following formula to calculate the fuzzy support value of itemset of $\langle X, A\rangle$, that is:

$$FS_{\langle X,A\rangle} = \frac{\sum_{t_i \in T} \pi_{x_j \in X} m_{x_j}(a_j \in A, t_i.x_j)}{|D|}$$

We use an example to illustrate the computation of the fuzzy support value.

Let $X = \{Age, Salary\}$ and $A = \{middle, low\}$ and a part of database shown in Table 6. The fuzzy significance of $\langle X, A\rangle$ is as follows:

$$FS_{\langle X,A\rangle} = \frac{0.35 + 0.06 + 0.1 + 0.12 + 0.12 + 0.32}{6} = 0.178.$$

$\langle Age, middle\rangle$	$\langle Salary, low\rangle$
0.7	0.5
0.2	0.3
0.5	0.2
0.3	0.4
0.6	0.2
0.8	0.4

Table 6: Part of Database Containing Membership

3.2.2 Fuzzy Confidence Value

We use the discovered frequent itemsets to generate all possible rules. If the union of antecedent and consequent has enough support and the rule has high confidence (and correlation value), this rule will be considered as interesting.

When we obtain a frequent itemset $\langle X, A\rangle$, we want to generate fuzzy association rules of the form, "If X is A then Y is B", where $X \subset Z$, $Y = Z - X$, $A \subset C$ and $B = C - A$. Having the frequent itemset, we know its support as well as the fact that all of its subsets will be also frequent. We can calculate the fuzzy confidence value as follows:

$$\begin{aligned}
FC_{\langle\langle X,A\rangle,\langle Y,B\rangle\rangle} &= \frac{FS_{\langle Z,C\rangle}}{FS_{\langle X,A\rangle}} \\
&= \frac{\frac{\sum_{t_i\in T}\pi_{z_j\in Z}m_{z_j}(c_j\in C, t_i.z_j)}{|D|}}{\frac{\sum_{t_i\in T}\pi_{x_j\in X}m_{x_j}(a_j\in A, t_i.x_j)}{|D|}} \\
&= \frac{\sum_{t_i\in T}\pi_{z_j\in Z}m_{z_j}(c_j\in C, t_i.z_j)}{\sum_{t_i\in T}\pi_{x_j\in X}m_{x_j}(a_j\in A, t_i.x_j)},
\end{aligned}$$

where $Z = X \cup Y$, $C = A \cup B$.

Since the fuzzy confidence value is the measure of the degree of support given by records, we use confidence to help us estimate the interestingness of the generated fuzzy association rules. In the above equation, we divide the fuzzy support of $\langle Z, C\rangle$ by fuzzy support of $\langle X, A\rangle$. Given the database in Table 6, we can calculate the fuzzy confidence value of the rule, "If *Age* is *middle* then *Salary* is *low*" as follows:

$$FC_{\langle\langle X,A\rangle,\langle Y,B\rangle\rangle} = \frac{0.35 + 0.06 + 0.1 + 0.12 + 0.12 + 0.32}{0.7 + 0.2 + 0.5 + 0.3 + 0.6 + 0.8} = 0.345.$$

3.2.3 Fuzzy Correlation Value

In data mining, an association rule $X \Rightarrow Y$ usually means X implies Y and we cannot assume Y also implies X because of the data distribution of X and Y. Therefore, we change the calculation of expectation such that we can accommodate the meaning of fuzzy association rules. The following equation is used for computing the fuzzy correlation value:

$$FCORR_{\langle\langle X,A\rangle,\langle Y,B\rangle\rangle} = \frac{Cov(X,Y)}{\sqrt{Var(X)\cdot Var(Y)}}$$

where

$$\begin{aligned}
Cov(X,Y) &= E\left[\langle Z,C\rangle\right] - E\left[\langle X,A\rangle\right]\cdot E^{'}\left[\langle Y,B\rangle\right] \\
Z &= X\cup Y, C = A\cup B \\
Var(X) &= E\left[\langle X,A\rangle^2\right] - E[\langle X,A\rangle]^2 \\
Var(Y) &= E^{'}\left[\langle Y,B\rangle^2\right] - E^{'}[\langle Y,B\rangle]^2 \\
E\left[\langle X,A\rangle\right] &= \frac{\sum_{t_i\in T}\pi_{x_j\in X} m_{x_j}(a_j\in A, t_i.x_j)}{|D|} \\
E^{'}\left[\langle Y,B\rangle\right] &= \frac{\sum_{t_i\in T}\pi_{y_k\in Y} m_{y_k}(b_k\in B, t_i.y_k)}{|D|}
\end{aligned}$$

In the above equations, we can see that the calculation of $E\left[\langle X,A\rangle\right]$ is similar to an ordinary expectation. $E^{'}\left[\langle Y,B\rangle\right]$ calculates the expectation of the consequent.

The value of the fuzzy correlation ranges from -1 to 1. Only a positive value tells that the antecedent and consequent are related. The higher the value is, the more related they are. Therefore, if the rule "If X is A then Y is B" holds, the fuzzy correlation of this rule should be at least greater than zero.

We will use the information in Table 6 to illustrate the calculation of the fuzzy correlation value of a rule. Given the rule, "If *Age* is *middle* then *Salary* is *low*", the fuzzy correlation value of the rule is as follows:

$$FCORR_{\langle\langle X,A\rangle,\langle Y,B\rangle\rangle} = \frac{0.178-0.172}{\sqrt{0.045\cdot 0.012}} = 0.261.$$

4 Algorithm for Mining Fuzzy Quantitative Association Rules

An efficient algorithm for mining quantitative association rules has been proposed in [Srikant et al.,1996]. However, a new algorithm is needed to solve the mining of fuzzy quantitative association rules. The problem of discovering all fuzzy quantitative association rules can be decomposed into two subproblems:

1. Find all itemsets that have fuzzy support $\left(FS_{\langle X,A\rangle}\right)$ above the user specified minimum support (see in section 3.2.1). These itemsets are called *frequent itemsets.*

2. Use the frequent itemsets to generate the desired rules. The general idea is that if, say, X and Y are frequent itemsets, then we can determine if the rule $X \Rightarrow Y$ holds by computing the ratio $\left(FC_{\langle\langle X,A\rangle,\langle Y,B\rangle\rangle}\right)$ (see in section 3.2.2). If this value is larger than the user specified minimum confidence value, then the rule will be interesting. We also can be use the fuzzy correlation value $\left(FCORR_{\langle\langle X,A\rangle,\langle Y,B\rangle\rangle}\right)$ for this problem (see in section 3.2.3).

An algorithm for mining quantitative association rules has the following inputs an outputs.
Inputs: A database D, three threshold values *minsup*, *minconf* and *mincorr.*
Output: A list of interesting rules.
Notations:

D	the database
D_T	the transformed database
F_k	set of *frequent k-itemset* (have k items)
C_k	set of *candidate k-itemset* (have k items)
I	complete item set
$minsup$	support threshold
$minconf$	confidence threshold
$mincorr$	correlation threshold

Main Algorithm $(minsup, minconf, mincorr, D)$

```
I = Search(D);
(C_1, D_T) = Transform(D, I);
k = 1;
(C_k, F_k) = Checking(C_k, D_T, minsup);
while (|C_k| ≠ ∅) do
begin
    inc(k);
    if k == 2 then
        C_k = Join1(C_{k-1});
    else C_k = Join2(C_{k-1});
    C_k =Prune(C_k);
    (C_k, F_k) = Checking(C_k, D_T, minsup);
    F = F ∪ F_k;
end
Rules(F, minconf, mincorr);
```

The subroutines are outlined as follows:

1. **Search**(D): The subroutine accepts the database, find out and returns the $I = \{i_1, i_2, \ldots i_m\}$ complete item set. For example, $I = \{Age, Salary\}$ for the database given in Table 3.

2. **Transform**(D, I): This step generates a new transformed (fuzzy) database D_T from the original database by user specified fuzzy sets. At the same time, the *candidate 1-itemset* C_1 will be generated from the transformed database. For example, $C_1 = \{\langle Age, young\rangle, \langle Age, middle\rangle, \langle Age, old\rangle, \langle Salary, low\rangle, \langle Salary, medium\rangle, \langle Salary, high\rangle\}$ is the complete set of *candidate 1-itemset.*

3. **Checking**$(C_k, D_T, minsup)$: In this subroutine, the transformed (fuzzy) database is scanned and the fuzzy support $\left(FS_{\langle X,A\rangle}\right)$ of candidates in C_k is counted. A *k-itemset* in C_k is deleted if its fuzzy support is less then minsup. The remaining candidate itemsets will be kept in C_k. At the same time, the frequent itemsets F_k will be generated from C_k.

4. **Join1**(C_{k-1}): This Join step generates C_2 from C_1 as follows:
   insert into C_2
   select $\langle X, A\rangle, \langle Y, B\rangle$
   from $\langle X, A\rangle, \langle Y, B\rangle$ in C_1
   where $X \neq Y$

For example, after this Join step C_2 will be $C_2 = \{\langle\langle Age, young\rangle, \langle Salary, high\rangle\rangle, \langle\langle Age, middle\rangle, \langle Salary, low\rangle\rangle, \cdots\}$, but $C_2 \neq \{\cdots, \langle\langle Age, young\rangle, \langle Age, middle\rangle\rangle, \cdots\}$.

5. **Join2**(C_{k-1}): This Join step generates C_k from C_{k-1} as in [Agrawal et al.,1993]. For example, if we have $\{\langle\langle Age, young\rangle, \langle Salary, low\rangle\rangle, \langle\langle Age, young\rangle, \langle Balance, low\rangle\rangle\}$ in C_{k-1}, $\{\langle\langle Age, young\rangle, \langle Salary, low\rangle, \langle Balance, low\rangle\rangle\}$ will be generated in C_k.

6. **Prune**C_k: During the prune step, the itemset will be pruned if a subset of the candidate itemset in C_k does not exist in C_{k-1}.

7. **Rules**(F): Find the rules from the *frequent itemsets* F. For example, if $\langle Age, young\rangle$ and $\langle Salary, low\rangle$ are frequent itemsets, then we get the $\langle Age, young\rangle \Rightarrow \langle Salary, low\rangle$ rule, if its fuzzy confidence value (and fuzzy correlation value) is larger then the user specified minimum value.

5 Experimental Results

In this section, we will examine the accuracy and efficiency of our approach by experimenting with a real-life dataset. We will describe the parameter settings and the results of different methods. We applied our approach to a database called FAM95. This database contains data for the 63756 families that were interviewed in the March 1995 Current Population Survey (CPS), conducted by the Bureau of the Census for the Bureau of Labor Statistics. The data had 23 attributes: 7 quantitative and 16 categorical.

5.1 Interest Measure

In this experiment, we use six quantitative attributes to illustrate how the fuzzy concept gives more interesting rules than the discrete. The quantitative attributes were age of head in years ("head" is the reference person if in a family), number of persons, children in family, education level of head, head's personal income and family income. Each quantitative attribute has three intervals/fuzzy sets.

Figure 5 shows the number of frequent itemsets for different minimum support. As expected, the number of frequent itemsets decreases as the minimum support increases from 10% to 45%. Fuzzy1 denotes the fuzzy method without normalization and Fuzzy2 denotes the fuzzy method with normalization. We can see that fuzzy method with normalization gives less

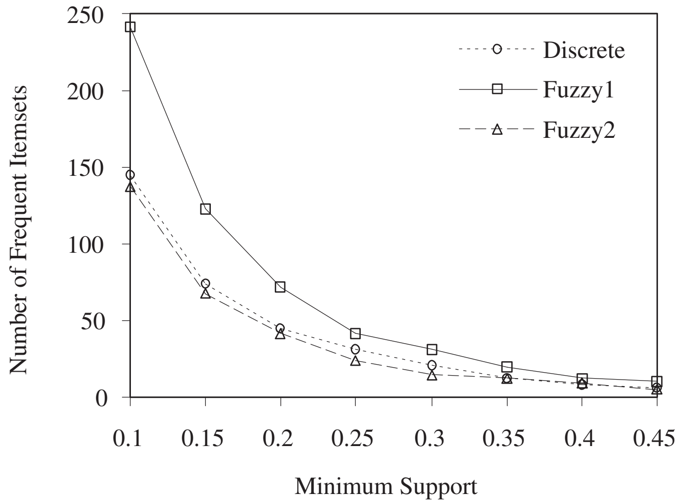

Figure 5: Number of Frequent Itemsets

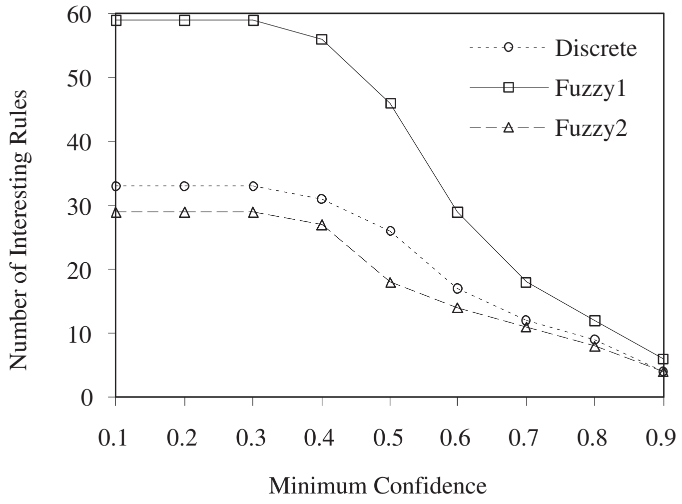

Figure 6: Number of Interesting Rules

frequent itemsets than fuzzy method without normalization. Although in the pursuance of normalization we were reducing its fuzzy values. This method and discrete interval method give similar frequent itemsets.

Figure 6 and Figure 7 show the number of interesting rules for different minimum confidence and correlation values. In both cases the minimum support was set to 20%. The results are quite similar to those of Figure 5. We can see that the fuzzy method without normalization using confidence to calculate interest measure gives the highest number of expected interesting rules. However, in the correlation case the fuzzy method with normalization gives more rules than the others if the minimum correlation value was 0.6.

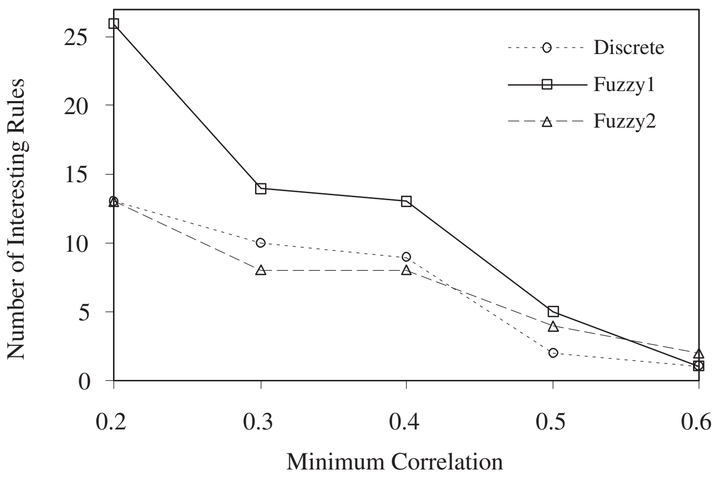

Figure 7: Number of Interesting Rules

The following shows some interesting rules. The minimum support was set to 20%, minimum confidence to 50%, and minimum correlation to 0.5.

IF *IncHead* is *low* THEN *IncFam* is *low*
IF *IncHead* is *medium* THEN *IncFam* is *medium*
IF *FamPers* is *low* AND *IncHead* is *low* THEN *IncFam* is *low*
IF *NumKids* is *low* AND *IncHead* is *low* THEN *IncFam* is *low*
IF *FamPers* is *low* AND *NumKids* is *low* AND *IncHead* is *low* THEN *IncFam* is *low*

5.2 Scale-Up Experiment

In this experiment, we will give the experimental results on the performance of the algorithm using the confidence and correlation interest measures. The

running time for the algorithm can be split into two parts:
- *Candidate generation.* The time for this is independent of the number of records.
- *Counting support, confidence and correlation.* The time for this is directly proportional to the number of records. When the number of records is large, this time will dominate the total time.

Thus we would expect the algorithm to have near-linear scaleup. This is confirmed by Figure 8, which shows the execution time as we increase the number of input records from 10000 to 64000. Note that we use five quantitative attributes in the database and each attribute has three fuzzy sets. We have set the user specified parameters such that both methods will give same number of rules. The graph shows that the methods scale quite linearly for this dataset.

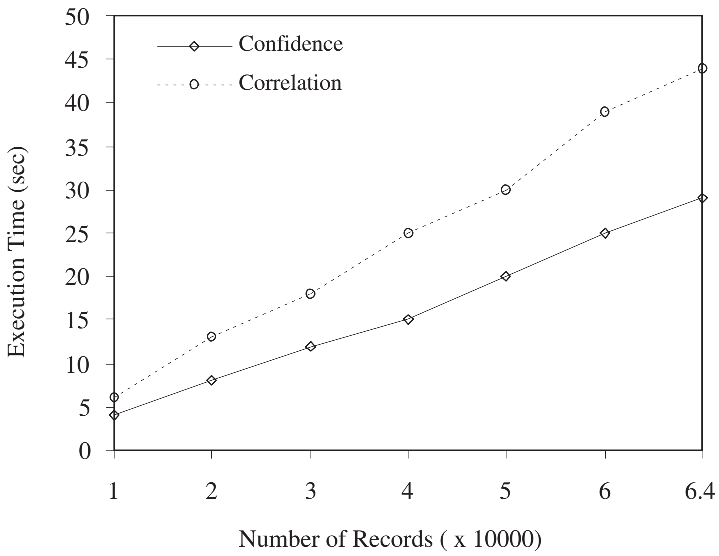

Figure 8: Scale-up: Number of Records

6 Conclusion

In this paper, we have shown a method to handle quantitative attributes.We assign each attribute with several fuzzy sets which characterize the quantitative attribute. Fuzzy sets provide a smooth transition between member and non-member of a set. The fuzzy association rule is also easily understandable to a human because of the linguistic terms associated with the fuzzy sets. We showed two different definition for interest measure: fuzzy confidence, and fuzzy correlation.

We proposed two different methods of mining fuzzy quantitative association rules: without normalization, and with normalization.

We gave an new algorithm for mining such quantitative association rules. Our experiments on a real-life dataset indicate that the algorithm scales linearly with the number of records. It also showed that the confidence interest measure gives better performance than the correlation interest measure.

Acknowledgment

I wish to thank Jukka Teuhola for his insightful comments and suggestions.

References

[Srikant et al.,1996] R. Srikant and R. Agrawal (1996), *Mining quantitative association rules in large relation tables*, Proceedings of ACM SIGMOD, pp. 1–12.

[Park et al.,1995] J.S. Park, M-S. Chen, P.S. Yu (1995), *An effective hash-based algorithm for mining association rules*, Proceedings of ACM SIGMOD, pp. 175-186.

[Agrawal et al.,1993] R. Agrawal, T. Imielinski, A. Swami (1993), *Mining association rules between sets of items in large databases*, ACM SIGMOD, Washington, DC, USA, pp. 207–216.

[Srikant et al.,1994] R. Srikant and R. Agrawal (1994), *Fast algorithms for mining association rules*, Proceedings of the 20th VLDB Conference, pp. 487–499.

[Gyenesei,1999] A. Gyenesei (1999), *Data mining approach for solving decision support problems of warehouse networks*, POLVAX (in hungarian), pp. 69–93.

[Toivonen,1996] H. Toivonen (1996), *Sampling large databases for association rules*, Proceedings of the 20th VLDB Conference.

[Piatesky et al.,1991] G. Piatesky-Sshapiro, W.J. Frawley (1991), *Knowledge Discovery in Databases*, AAAI Press/The MIT Press, Menlo Park, California.

[Mannila et al.,1994] H. Mannila, H. Toivonen, A.I. Verkamo (1994), *Efficient Algorithms for discovering association rules*,KDD-94: AAAI Workshop on Knowledge Discovery in Databases, pp. 181–192.

[Sarasere et al.,1995] A. Sarasere, E. Omiecinsky, S. Navathe (1995), *An efficient algorithm for mining association rules in large databases*, 21st Int'l Conf. on VLDB, Zrich, Switzerland.

[Fayyad et al.,1994] U.M. Fayyad, R. Uthurusamy (1994), *Efficient algorithms for discovering association rules*, AAAI Workshop on KDD, Seattle, Washington, pp. 181–192.

Turku Centre for Computer Science
Lemminkäisenkatu 14
FIN-20520 Turku
Finland

http://www.tucs.abo.fi

University of Turku
- **Department of Mathematical Sciences**

Åbo Akademi University
- **Department of Computer Science**
- **Institute for Advanced Management Systems Research**

Turku School of Economics and Business Administration
- **Institute of Information Systems Science**