Mining Fuzzy Quantitative Association Rules

Weining Zhang*
Division of Computer Science
University of Texas at San Antonio
wzhang@cs.utsa.edu

Abstract

Given a relational database and a set of fuzzy terms defined for some attributes, we consider the problem of mining fuzzy quantitative association rules that may contain crisp values, intervals, and fuzzy terms in both antecedent and consequent. We present an algorithm extended from the equi–depth partition (EDP) algorithm for solving this problem. Our approach combines interval partition with pre–defined fuzzy terms and is more general.

1. Introduction

An association rule [1], expressed as $Y \rightarrow X$, indicates a pattern in a transaction database that transactions containing items Y also contain items X. The significance of such a rule is measured by its *support* and *confidence*. The support is the proportion of transactions that contain both X and Y, and the confidence is the proportion of transactions containing Y that also contain X. The problem is to find all association rules that satisfy user–specified minimum support and confidence. Many algorithms [1, 2, 5, 3] have been proposed for solving the problem. Association rules are useful for marketing, document clustering, Web management, etc..

The quantitative association rule [4, 6] mining views records in a relational database as transactions that contain attribute–value pairs (or attribute–interval pairs, if the attribute is numerical) as items. A quantitative association rule may have attribute–value/interval pairs in both antecedent and consequent. For example, a quantitative association rule may be $Age \in [40, 45]$ and $FastTrack = no \rightarrow Married = yes$. The main reason for quantitative association rule mining is that numerical attributes typically contains many distinct values. The support for any particular value is likely to be low, while that for intervals are much higher.

*This research was partially supported by a research grant of the Natural Science and Engineering Research Council of Canada.

For many applications, an association rule may be more interesting if it reveals relationship among some useful concepts, such as "high income", "new car", and "frequent customer". These concepts are often imprecise or uncertain. In this paper, we consider the problem of mining fuzzy quantitative association rules in relational databases. Interesting concepts are defined using fuzzy terms and interpreted based on fuzzy set [7]. We refer association rules involving fuzzy terms as *fuzzy quantitative association rules*. For example, $Age = young$ and $Income = high \rightarrow RiskLevel = mediumHigh$ is a fuzzy quantitative association rule, where "young", "high" and "mediumHigh" are fuzzy terms. We define the problem of mining fuzzy quantitative association rules and present an algorithm extend from the EDP algorithm [4] to solve the problem. Our approach is able to discover association rules that have crisp values, intervals, and fuzzy terms in both the antecedent and the consequent, thus, is more general than previous approaches. To the best of our knowledge, this problem has not been studied elsewhere.

The remainder of this paper is organized as the following. In Section 2, we present some background on how fuzzy terms are interpreted. In Section 3, we define the problem of mining fuzzy quantitative association rules. In Section 4, we present an algorithm for solving the problem. Section 5 concludes the paper.

2. Interpretation of Fuzzy Terms

A fuzzy data has an uncertain or imprecise value. We associate each fuzzy data v with a fuzzy term and a membership function (of a fuzzy set [7]). A fuzzy term is a linguistic term, such as, "young" and "high income". The membership function, denoted by μ_v, maps each crisp value x in the universe of v to a membership degree $\mu_v(x)$ in $[0, 1]$ to indicate the possibility of $v = x$.

A membership function can be defined in a number of ways. Over a numerical universe, a membership functions is typically convex (with a convex curve) and normal (at least one member has degree 1). As in [9, 8] we con-

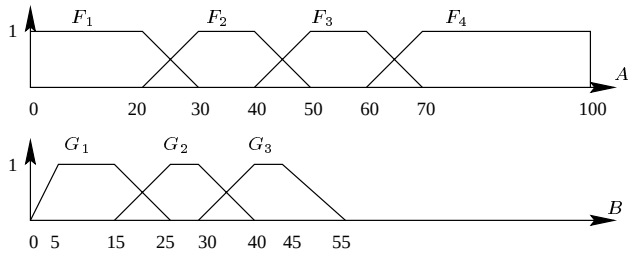

Figure 1. Fuzzy Terms

sider membership functions of a trapezoidal shape, and denote them by $MF(a, b, c, d)$, where the parameters mark the endpoints of the shape. For example, the membership function defining fuzzy term F_2 in Figure 1 is denoted by $MF(20, 30, 40, 50)$. If a value v has a membership function defined by $MF(a, b, c, d)$, the interval $[a, d]$ is called the *supporting interval* of v. As special cases, $MF(l, l, u, u)$ defines an interval $[l, u]$, $MF(v, v, v, v)$ defines a crisp value v, and $MF(a, b, b, d)$ is a triangular function.

Over a categorical universe, membership function is defined by $\mu_v = x_1/m_1 + x_2/m_2 + \cdots + x_k/m_k$, where x_i is a value in the universe and m_i is the membership degree of x_i. The membership function of a single crisp value v is $\mu_v = v/1$.

3. Problem Statement

Let $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$ be a set of attributes, $U(A_i)$ be the universe of A_i, that is, the set of crisp values in A_i, $FT(A_i)$ be a set of fuzzy terms (pre–)defined over $U(A_i)$, $In(A_i) = \{[a, b] \mid a, b \in U(A_i), a < b\}$ be the set of all intervals over $U(A_i)$ ($In(A_i) = \emptyset$ if A_i is categorical), $Dom(A_i) = U(A_i) \cup In(A_i) \cup FT(A_i)$ be the domain of A_i, and $\mathcal{I} = \{< a, v > \mid a \in \mathcal{A}, v \in Dom(a)\}$. We refer $< a, v > \in \mathcal{I}$ as an *item*. An item is fuzzy if v is a fuzzy term. Thus, an item may represent a crisp value, an interval (if the attribute is numerical), or a fuzzy term. For each $X \subseteq \mathcal{I}$, the set of attributes in X is denoted by $attr(X) = \{a \mid < a, v > \in X\}$.

A data record is a set of attribute–value pairs, where all attributes are distinct and all values are crisp. A data record r supports an item $< a, v > \in \mathcal{I}$ with a degree d, if there is an item $< a, v' > \in r$, such that, $d = \mu_v(v')$. A data record supports an itemset $X \subseteq \mathcal{I}$ with a degree d if it supports every item in X with degree d or higher, and at least one of them with degree d. Notice that, the degree with which a data record supports an itemset is in the range of $[0, 1]$. If the database contains N records, and the total degree of support to an itemset is M, M/N gives the support of the itemset.

A *fuzzy quantitative association rule* is am implication $Y \rightarrow X$, where $X \subset \mathcal{I}$, $Y \subset \mathcal{I}$, and $attr(X) \cap attr(Y) = \emptyset$. The support of the association rule is the support of $X \cup Y$. The confidence is the ratio of the support of $X \cup Y$ and the support of Y.

Given a set of data records with a set of attributes, and a set of fuzzy terms defined for the attributes, the problem is to find all fuzzy quantitative association rules that satisfy user–specified minimum support and confidence.

Example 3.1 With fuzzy terms defined on attributes A and B in Figure 1, data records in Figure 2, a minimum support 20% and a minimum confidence 70%, Figure 2 shows some of the intervals, frequent itemsets, and fuzzy quantitative association rules that may be discovered. □

4. The Algorithm

We use an EDPFT (equal–depth partition with fuzzy terms) algorithm to discover fuzzy quantitative association rules. The algorithm consists of the following steps.

1. Use equal–depth partition to determine intervals for numerical attributes.
2. Map crisp values and fuzzy terms of each categorical attribute into consecutive integers.
3. Use an extended Apriori Algorithm to discover frequent itemsets.
4. Generate all association rules from the frequent itemsets, and keep only those satisfying the minimum confidence.
5. Remove association rules that are less interesting.

Except for Steps 1 and 4, this algorithm extends the EDP algorithm to account for the inclusion of fuzzy terms in items. The extensions are discussed in following subsections.

4.1. Candidate Itemset Generation

We extend the Apriori algorithm, by incorporating an order of intervals, for the join operation in candidate generation.

Items in an itemset are ordered first by the lexicographical order of their attributes. For items that have an identical numerical attribute, we associate them with the supporting intervals (see Section 2) of their values, and order them based on these intervals. We adopt the partial order of intervals in [8]. Basically, the order of two intervals is determined by first comparing their left ends, and then, if necessary, their right ends. For items that have identical categorical attribute, the order is given by the integers that

Data Records			
Rid	A	B	C
101	18	20	4
102	27	37	2
103	32	43	2
104	32	46	2
105	38	43	2
106	38	46	2
107	45	6	1
108	45	25	3
109	55	28	3
110	65	28	3

Base Intervals	
A	B
$[0,4]$	$[0,2]$
$[5,9]$	$[3,5]$
$[10,14]$	$[6,8]$
$[15,19]$	$[9,11]$
...	...
$[95,100]$	$[57,60]$

Frequent Item	Support
$< A,[0,29] >$	0.2
$< A,[30,34] >$	0.2
$< A,[35,39] >$	0.2
$< A,[45,49] >$	0.2
$< A,F_2 >$	0.57
...	...
$< B,[0,20] >$	0.2
$< B,[9,26] >$	0.2
$< B,[27,29] >$	0.2
$< B,[42,44] >$	0.2
$< B,G_2 >$	0.38
...	...

Fuzzy Quantitative Association Rules	support	confidence
$\{< A,F_2 >,< C,2 >\} \rightarrow < B,G_3 >$	45%	95.7%
$\{< A,F_3 >,< C,3 >\} \rightarrow < B,G_2 >$	20%	100%
$\{< A,F_3 >,< C,3 >\} \rightarrow < B,[27,29] >$	20%	75%
$\{< A,[55,100] >,< C,3 >\} \rightarrow < B,[27,29] >$	20%	100%
...		

Figure 2. Example

crisp values and fuzzy terms of the attribute are mapped to (fuzzy terms are mapped to large integers).

With such an ordering, the (k)–itemsets can be obtained from the frequent ($k-1$)–itemsets with a join process similar to that of Apriori.

Example 4.1 If $\{< A,[0,29] >,< B,G_1 >\}$ and $\{< A,[0,29] >,< B,G_2 >\}$ are both in the frequent (2)–itemsets, the join will create a (3)–itemset $\{< A,[0,29] >,< B,G_1 >,< B,G_2 >\}$, because the supporting interval of $< B,G_1 >$ is $[0,25]$ and that of $< B,G_2 >$ is $[15,40]$. Thus $< B,G_1 >$ will precede $< B,G_2 >$. Base on the join condition, this (3)–itemset will be generated only once. □

4.2. Counting Support

The degree with which a data record t supports an itemset X is calculated as follows. For each attribute $A \in attr(X)$, let $< A,v >\in t$ and $< A,u >\in X$. The degree with which t supports $< A,u >$ is given by $ds_A(t,X) = \mu_u(v)$, that is, the possibility for v to be the value represented by u. Since crisp values, intervals, and fuzzy terms are all interpreted by membership functions, the degree of support can be calculated in the same way for all items. The degree with which t supports X is given by $min_{A \in attr(X)}(ds_A(t,X))$, which is consistent with the semantics of fuzzy logic AND. It is obvious that the support counted in the Apriori is a special case.

Example 4.2 In Example 3.1, data record $\{< A,32 >,< B,46 >,< C,2 >\}$ supports itemset $\{< A,F_2 >,< B,G_3 >,< C,2 >\}$ with degree $min(1,0.9,1) = 0.9$. □

When counting supports, we read each data record, find each candidate itemset that is supported by the data record with a non–zero degree, and increment the counter of that itemset by the degree of support.

To find all candidate itemsets (partially) supported by a data record, We extend the "super–candidate" technique of the EDP algorithm to account for items that have an identical attribute and overlapping intervals. Basically, candidate itemsets are partitioned into groups, called "super–candidate", so that, itemsets of the same "super–candidate" have an identical set of items with categorical attributes, and an identical number of items with each distinct numerical attribute. The items with categorical attributes are referred to as the categorical items of the "super–candidate".

To find a "super–candidate" supported by a data record, we use a *hash–tree* to store the categorical items of "super–candidates". The leave nodes of the hash–tree contain categorical items of "super–candidates" and pointers to multi–dimensional data structures containing the remaining items of the itemsets in "super–candidates". Assume that each itemset of a "super–candidate" has n numerical items. Each of these numerical items corresponds to an interval (defined by the membership function of its attribute value) on a dimension in an n–dimensional space. Thus, each itemset in the "super–candidate" represents an n–dimensional rect-

angle. We can map a data record into a point in this n–dimensional space, thus reduce the problem of finding all itemsets supported by the data record to that of finding all rectangles that contain the point. We store numerical items of itemsets of a "super–candidate" in a multi–dimensional data structure, such as a $R*$–tree or an n–dimensional array.

It is possible for a data record to support itemsets in multiple "super–candidates". We use a linked list structure to determine which "super–candidate" needs to be searched. The detail is omitted due to the space limit.

Once a "super–candidate" is found, we use the multi–dimensional data structure of the "super–candidate" to search for itemsets. To do so, we map the remaining items of the data record into a point in the multi–dimensional space of the "super–candidate". Let the "super–candidate" have n numerical items. If each of the n items has a distinct attribute, the mapping is straightforward. If, however, some items have an identical attribute, say A, then the item with A in the data record must be duplicated before the mapping is done.

Once an itemset is identified, the counting of support is done as described previously.

4.3. Interestingness Measures

In Steps 3 and 5 of the EDPFT algorithm, we prune (k)–itemsets and association rules using both the R–interesting measure of EDP algorithm and a new measure for certainty.

Among association rules that describe the same association, we prefer the one that is more precise, accurate, and certain. Since items with numerical attributes are associated with intervals defined by their membership functions, we can talk about the overlap and containment of items in terms of their associated intervals.

Example 4.3 Consider items $< A, F2 >$, $< A, F3 >$, $< A, [45, 49] >$, where F_2 and F_3 are fuzzy terms shown in Figure 1. The three items are pairwise overlapping. The intersection of $< A, F2 >$ and $< A, F3 >$ is $[40, 50]$. Item $< A, [45, 49] >$ is contained in $< A, F2 >$, $< A, F3 >$, and $\{< A, F2 >, < A, F3 >\}$. □

Definition 4.1 Let X and $\hat{X}$ be itemsets. We say that X is *more certain* than $\hat{X}$ if they have the same support, $attr(X) = attr(\hat{X})$, and for every attribute $A \in attr(X)$, the intersection of items with A in X is contained in the intersection of items with A in $\hat{X}$.

Let $Y \rightarrow X$ and $\hat{Y} \rightarrow \hat{X}$ be two fuzzy quantitative association rules. We say that $Y \rightarrow X$ is *more certain* than $\hat{Y} \rightarrow \hat{X}$ if they have the same confidence, and $X \cup Y$ is more certain than $\hat{X} \cup \hat{Y}$. □

Example 4.4 Consider itemsets $\{< A, F_2 >, < A, F_3 >\}$ and $\{< A, [45, 47] >\}$, where F_2 and F_3 are fuzzy terms shown in Figure 1. If they have the identical support, then $\{< A, [45, 47] >\}$ is more certain than $\{< A, F_2 >, < A, F_3 >\}$.

Consider following fuzzy quantitative association rules.

$$\{< A, F_3 >, < A, [55, 100] >\} \rightarrow < B, [42, 44] >$$
$$\{< A, F_3 >, < A, [55, 100] >\} \rightarrow < B, G_3 >$$
$$\{< A, F_3 >\} \rightarrow < B, [42, 44] >$$

If they have the same support and confidence, the first rule is more certain than others. □

5. Conclusion

In this paper, we defined the problem of mining fuzzy quantitative association rules in relational databases. We present an EDPFT algorithm for solving this problem. Our approach is able to discover association rules that have crisp values, intervals, and fuzzy terms in both the antecedent and the consequent, thus, is more general.

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.

[2] J. Han and Y. Fu. Discovery of multi–level association rules from large databases. In *Proceedings of the 21th International Conference on Very Large Data Bases*, pages 420–431, 1995.

[3] J. S. Park, M. S. Chen, and P. S. Yu. An efficient hash–based algorithm for mining association rules. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 175–186, 1995.

[4] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1–12, 1996.

[5] H. Toivonen. Sampling large databases for association rules. In *Proceedings of the 22nd International Conference on Very Large Data Bases*, pages 134–145, 1996.

[6] K. Wang, S. Hock, W. Tay, and B. Liu. Interesting–based interval merger for numeric association rules. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 121–127, 1998.

[7] L. Zadeh. Fuzzy set. *Information and Control*, 8:338–353, 1965.

[8] W. Zhang and K. Wang. An efficient evaluation of a fuzzy equi–join using fuzzy equality indicators. *IEEE Transactions on Knowledge and Data Engineering*, 1999. to appear.

[9] W. Zhang, C. Yu, B. Reagan, and H. Nakajima. Context dependent interpretations for linguistic terms in fuzzy relational databases. In *IEEE International Conference on Data Engineering*, pages 139–146, 1995.