Modifying Fuzzy Association Rules with Linguistic Hedges*

Qiang Wei[1], Guoqing Chen
School of Economics & Management, Tsinghua University, Beijing 100084, China
{weiq, chengq}@em.tsinghua.edu.cn

Geert Wets
Faculty of Applied Economic Sciences, Limburg University, 3590 Diepenbeek, Belgium
geert.wets@luc.ac.be

Abstract

This paper introduces linguistic hedges in data mining, especially in association rules mining, based on our recent work on fuzzy taxonomic structures where fuzzy rules like "Expensive cosmetics ⇒ Tropical fruits" can be dealt with. In order to express the decision knowledge more naturally, the notion of fuzzy association rules with linguistic hedges is presented, such as "very expensive goods ⇒ sort of fruit". Furthermore, a method of mining all possible fuzzy association rules with a particular pool of linguistic hedges is discussed.

1. Introduction

Data mining is one of the emerging fields of information processing. Recently, a lot of attention has been paid to the discovery of association rules [1, 2, 4-7]. An example of such rules is "Apple⇒Pork" (e.g., customers who bought apples turned to buy pork). In particular, Srikant and Agrawal [6] introduced the notion of generalized association rules so as to discover the association upon all levels of presumed exact taxonomic structures (e.g., Fruit⇒Meat).

In many real-world applications, however, the taxonomic structures may not be crisp. This issue is deemed to be of particular interest to high-level decision-makers. Thus, in [7], we presented the notion of fuzzy taxonomic structures and extended the approach to discovering generalized association rules, in which each child-node may belong to its parent-node in a partial degree (between 0 and 1). Generally speaking, each interior node (non-leaf node) can be regarded as a fuzzy set (or linguistic term) represented by its child nodes. In this way, certain fuzzy association rules can be represented and discovered, such as "Tropical Fruit⇒Fresh Meat", where tropical fruit and fresh meat are both fuzzy terms. This allows managers or decision-makers to express higher-level (e.g., abstract, conceptual) concepts and knowledge, as well as in a more natural manner. Here, *Dsupport* and *Dconfidence* are used as measures in the mining process. *Dsupport* of a fuzzy rule X⇒Y means the "number" (e.g., Σcount) of the transactions containing X∪Y, and *Dconfidence* of X⇒Y = *Dsupport*(X⇒Y) / *Dsupport*(X) [7]. A set that contains k items is called a k-candidate itemset, and if Dsupport exceeding the threshold min-support, the itemset is called a k-frequent itemset. Only the rules with Dsupport and Dconfidence exceeding the threshold min-support and min-confidence are regarded useful.

In this paper, we further consider using linguistic

* Partly supported by the National Science Foundation of China (No. 79925001).
[1] Corresponding author

0-7803-6274-8/00/$10.00 © 2000 IEEE

hedges (modifiers) to modify the fuzzy items in the fuzzy taxonomic structures, such as "*very* expensive goods", "*almost* young people". There are two reasons that linguistic hedges are worth being considered in mining fuzzy association rules: First, with the operation of linguistic hedges on the items, the discovered knowledge is more understandable and closer to human language. For the decision-makers, especially the top managers, this type of knowledge may be more often used and meaningful. Second, it can enrich the semantics of association rules and make the discovered rules more granular. For example, we can get the rules like "Apple ⇒ Jeans", "Expensive Apple ⇒ Cool Jeans", and "Very Expensive Apple ⇒ More-or-less Cool Jeans".

2.Main ideas and Problem Statement

Linguistic hedges, such as very, more-or-less, sort of, are not themselves modeled by fuzzy sets as the primary terms are, but rather are modeled as operators acting on the fuzzy sets representing the primary terms [3].

Consider a hedge operator H_λ, which can be used to deal with a number of linguistic hedges. Let F(U) be the set of all fuzzy set on U, and H_λ be a hedge operator with $\lambda \in [0, \infty]$, then H_λ is a mapping from F(U) to F(U) such that $\forall A \in F(U)$,

$$H_\lambda(A) = A^\lambda \in F(U) \; or \; \forall a \in U, \mu_{H_\lambda(A)}(a) = [\mu_A(a)]^\lambda \in [0,1].$$

When $\lambda > 1$, H_λ reduces the membership degrees for the elements of the fuzzy set being modified, which are called concentration operators, while $\lambda < 1$, H_λ increases the membership degrees for the elements of the fuzzy set being modified, which are called dilation operators. For example, H_2 is referred to as a concentration operator for hedge "very" semantically. Given a linguistic hedge h = *very* and a fuzzy term (node) w = expensive electronics = {1/air-conditioner, 0.6/mobile, 0.5/television, ...}, then hw = *very* expensive electronics = {1/air-conditioner, 0.36/mobile, 0.25/television, ...}.

Since in the fuzzy taxonomic structures, an interior node could be expressed as a fuzzy set on its child-nodes [7], the interior node could be modified in forms of hedges with the same child-nodes. Then if we apply each $h_i \in H$ onto the items (nodes) in the taxonomic structures, we can derive all the fuzzy sets of the modified items with linguistic hedges (i.e., h_iw's). In so doing, the modified items could be added into the original structures and form the new fuzzy taxonomic structures. Apparently, the methods proposed in [7] could be applied in a straightforward fashion. However, it is worth noting that such methods can be considerably improved, since a great number of itemsets can be pruned in the mining process due to certain properties, which will be discussed in the next section.

Now mining fuzzy association rules with linguistic hedges is to discover all the possible rules with any possible linguistic hedges in the transaction set. Three basic inputs are as follows:

A. A transaction set T as an original data source.

B. Pre-specified fuzzy taxonomies (FG) associated with transaction set T. Concretely [6, 7], let FG be a directed graph on the items, $I = \{i_1, i_2, \ldots, i_m\}$. An edge in FG represents a fuzzy *is-a* relationship, which means along with each edge, there exists a partial degree μ with which the child-node on this edge belongs to its parent-node on this edge, where $0 \le \mu \le 1$. Figure 1 shows an example.

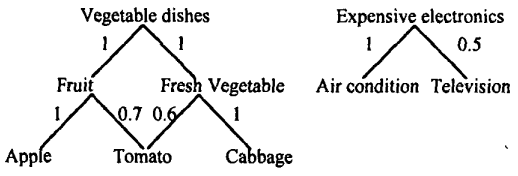

Figure 1. An example of fuzzy taxonomic structures

C. A set (or pool) of linguistic hedges (modifiers), H = {h | h is a hedge operator represented by H_λ}. H

could be specified by users or decision-makers, or obtained from their past querying records, which reflects either their interests or their behaviors. An example is web mining to trace customers' interests and behaviors in relation to relevant product web-sites. This may be particularly meaningful for companies to gain/sustain their competitive advantages in e-business environments.

3.Building New Fuzzy Taxonomies

This process consists of constructing a match table and the corresponding taxonomic structures.

A. Match table

When applying the hedges in H onto the items in FG, not all the linguistic hedges can be applied on every node. First, semantically, none of the leaf-nodes can be operated on any linguistic hedges. Each leaf-node is basic item of the taxonomic structures, and is not regarded as a fuzzy set on other items.

Second, some linguistic hedges cannot be applied onto certain items. Usually, "very" can be applied onto any nouns with adjective, but cannot be applied onto nouns directly. However, "sort of" can be applied onto both nouns and nouns with adjective. Typically, the pool of linguistic hedges, H, may contain adjectives, which can be used to modify nouns, and/or adverbs, which can be used to modify adjectives.

To facilitate the meaningful linguistic modification, a match table will be constructed based on FG and H. Four steps are identified: 1. Divide the items in I into two sets, nouns and nouns with adjective. 2. Divide the hedges in H into two sets, adjectives and adverbs. Since some hedges can be used as both adjectives and adverbs, such as "sort of", they may appear in both sets. 3. Match each item with corresponding hedges according to the above principle. 4. Filter the match table, say, by users or experts. This is because that some combinations of specific hedges and specific items are not correct linguistically, such as "higher than fresh fruits". An example of a match table is as follows:

Items	Corresponding hedges
Fruit	Sort of
Fresh Vegetable	Very, Sort of
Vegetable dishes	Sort of
Expensive electronics	Very, Sort of

Table 1. An example of a match set

B. New fuzzy taxonomic structures

With the match table and H, we can construct the fuzzy sets of all the new modified items, by which we can add all the new items into original structures FG and form new structures called FG'. For example, given figure 1 and table 1, we can obtain new fuzzy taxonomic structures as follows (partly).

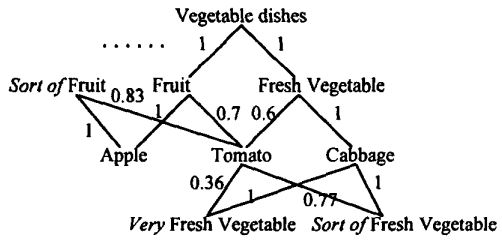

Figure 2. An example of FG'

For each item (node w) to be modified, a new node (hw) such as "*very* expensive electronics" is labeled, and then constructed as a fuzzy set by connecting it to the corresponding elements where the respective degrees are modified according to H_λ.

4.Mining and computational complexity

In this section, some key steps of mining will be discussed. Based on the method in [7], the membership degree that each leaf-node belongs to its ancestor-node

is first computed. Then, the candidate itemsets and frequent itemsets will be generated. Next, for all the frequent itemsets, the Dconfidence of association rules will be computed, by which the final rules are derived. Notably, generating the frequent itemsets is the most crucial and time-consuming step. In this regard, there are two ways to optimize this step.

A. Optimization I

Straightforward mining without any optimization may produce correct results but inefficiently, due to the fact that the number of items in FG' is much bigger than the number of items in FG. Generally, each interior-node will be expanded to $m+1$ nodes if there are on average m linguistic hedges in H which are operated on each node. Then, there will be $n\times(m+1)$ nodes in FG' if there are n nodes in FG (regardless the leaf-nodes).

In order to simplify the analysis, we assume min-support = 0, which means all possible combination of items are frequent itemsets and should be counted. And we also do not eliminate the combination that the child-node and its parent-node are in the same itemset.

Thus, theoretically, we will generate $2^{n\times(m+1)}$ frequent itemsets and should scan the database for $2^{n\times(m+1)}$ times, if using the straightforward mining strategy directly on FG'. Semantically, however, in FG', many items are usually modified items with different linguistic hedges based on a same original item in FG, which we call both them and the original item to be in a **class of items**. For example, "very expensive electronics", "relatively expensive electronics" and "expensive electronics" are in one class of items. Then when combining different items into an itemset, we cannot combine any two items of the same class into one itemset, because it is meaningless.

Therefore, when we generate itemsets on FG', we can only select one item from specific class of items (New Algorithm). Then the number of frequent itemsets that we can generate is $(m+2)^n$. Under the same assumption, the scan of database in mining fuzzy generalized association rules on FG without linguistic hedges is 2^n. Then we compare these three situations in figure 3, where **A** presents using the old algorithm on FG', **B** presents using new algorithm on FG', **C** presents using old algorithm on FG.

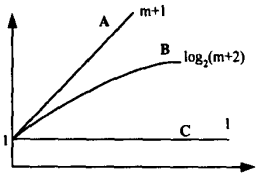

Figure 3. Comparison of the three situation

For $(m+2)^n = 2^{\log_2(m+2)^n} = 2^{n\log_2(m+2)}$ and m ≥ 0.

$\Rightarrow 2^n \le 2^{n\log_2(m+2)} \le 2^{n(m+1)}$, it appears that, with the enlargement of m, the space between **A** and **B** and space between **B** and **C** are bigger and bigger, but the speed of increment of the space between **B** and **C** is slower and slower, which means that using New Algorithm can reduce the computational complexity compared with **A**. Especially, when m = 0 (representing that no linguistic hedges are applied), the three situations are the same.

B. Optimization II

Further optimization may be made for the algorithm in the mining process.

First, in a class of items, only one is the basic item (node) and others result from operating onto the basic item by H_λ. The basic item in the class can also be regarded as operated by H_1. According to the definition of H_λ, given two items i_1 and i_2 in one class, which are operated by $H_{\lambda 1}$, $H_{\lambda 2}$ respectively, and $\lambda_1 > \lambda_2$, then the degree that each child-node belongs to i_1 is less than the degree that the same child-node belongs to i_2.

Given two itemsets $A=\{i_1, i_2, \ldots, i_p\}$ and $B=\{i'_1, i'_2, \ldots, i'_p\}$, where i_j and i'_j are in the same class of items, which are operated by $H_{\lambda j}$, $H_{\lambda' j}$ respectively, and

$\lambda_j \leq \lambda'_j$, $1 \leq j \leq p$, then it is clear that the degree that t $\in$ T supports A is greater than or equal to the degree that t supports B according to the definition of Dsupport in [7]. Then Dsupport of A is greater than or equal to that of B.

Next, with the New Algorithm, we could sort the class of items lexicographically, and for each class we sort the items in the ascending order of λ. This can guarantee that, in generating candidate itemsets, the itemset with smaller Dsupport appears after the itemset with larger Dsupport. Consequently, the 1-frequent itemsets and k+1-frequent itemsets (for $k > 0$) could be generated as follows:

1. When we generate 1-frequent itemsets, we can find if an itemset composed of one item, i, is not a frequent itemset, then any itemset composed of other items in the same class to which i belongs is not a frequent itemsets either. For example, if "sort of fresh fruit' is not a frequent itemset, clearly "very fresh fruit" is not a frequent itemset either. In this way, we can filter such itemsets to reduce the time in scanning of the database.

2. After generating k+1 candidate itemsets based on k-frequent itemsets, we should compute Dsupport of each candidate itemset to eliminate the non-frequent itemsets. If we find an itemset such as A that is not a frequent itemset, then we can scan the rest of the set of k+1-candidate itemsets and eliminate all the itemsets such as B with respect to A, for they are not frequent itemsets either. This results in the set of k+1-frequent itemsets.

Ongoing studies for the algorithm optimization include further theoretical explorations of itemsets properties and related pruning strategies, as well as more detailed analysis of experiments with both synthetic and real data.

5. Conclusions

In this paper, linguistic hedges have been incorporated into fuzzy association rules to facilitate the representation and discovery of more meaningful knowledge in a natural manner. This is also deemed to be of particular interest for web mining in e-business environments. In dealing with the problem, new fuzzy taxonomic structures could be constructed from the original one by directly applying the linguistic hedges onto the appropriate basic nodes. Based on the straightforward mining strategy, further optimizations have been discussed to reduce the number of candidate/frequent itemsets for the classes of items as well as for certain itemsets with smaller Dsupport.

References

1 R. Agrawal, T. Imielinski, A. Swami, *Mining Association Rules between Sets of Items in Large Databases*, Proc. of ACM SIGMOD Conf. Washington DC, USA, May 1993.

2 R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. Inkeri Verkamo, *Fast Discovery of Association Rules* in Advances in Knowledge Discovery and Data Mining, AAAI Press/The MIT Press, 1996.

3 Guoqing Chen, Fuzzy Logic in Data Modeling: semantics, constraints and database design, Kluwer Academic Publishers, Boston, 1998.

4 J, Han, Y. Fu, *Discovery of Multiple-level Association Rules from Large Databases*, Proceedings of the 21st International Conference on Very Large Databases, Zurich, Switzerland, September 1995.

5 Savasere, E. Omiecinski, S. Navathe. *An Efficient Algorithm for Mining Association Rules in Large Databases*, Proceedings of the VLDB Conference, Zurich, Switzerland, September 1995.

6 R. Srikant, R. Agrawal, *Mining Generalized Association Rules*, Proc. of the 21st VLDB Conf. Zurich, Swizerland, 1995.

7 Qiang Wei, Guoqing Chen, Mining Generalized Association Rules with Fuzzy Taxonomic Structures, NAFIPS 99, New York, 1999.