Discovery of Generalized Association Rules with Multiple Minimum Supports¹

Chung-Leung Lui and Fu-Lai Chung

Department of Computing, Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong.
{csl1ui, cskchung}@comp.polyu.edu.hk

Abstract. Mining association rules at multiple concept levels leads to the discovery of more concrete knowledge. Nevertheless, setting an appropriate *minsup* for cross-level itemsets is still a non-trivial task. Besides, the mining process is computationally expensive and produces many redundant rules. In this work, we propose an efficient algorithm for mining generalized association rules with multiple *minsup*. The method scans appropriately $k+1$ times of the number of original transactions and once of the extended transactions where k is the largest itemset size. Encouraging simulation results were reported.

1. Introduction

Generalized association rules (GAR) [9] and multiple level association rules (MLAR) [2,3] were proposed to find rules at multiple concept levels. Unfortunately, a dilemma exists when setting *minsup* in GAR. First, if it is set too high, low-level rules may not have *minsup*. Second, if it is set too low, this will produce too many rules [7]. On the other hand, MLAR can set different *minsup* at each level, but extension to cross-level itemsets seems to be difficult. Furthermore, both methods generate many redundant rules. Pruning algorithms [3,9] were then proposed, but they always remove descendant rules, retaining rules at the most general concept that may often lead to rules corresponding to prior knowledge [4]. Efficiency is another major concern. If the largest itemset discovered is k -itemset, the number of scanning of database in GAR is roughly $k+1$ times of the number of extended transactions. MLAR scans roughly $k+1$ times of the number of original transactions for each level of taxonomy.

In this work, a new framework based on concept generalization [5-6,8] for mining generalized association rules with multiple *minsup* is proposed. It scans roughly $k+1$ times of the number of original transactions and once of the extended transactions, if the largest size of discovered itemsets is k . The *minsup* of each itemset can be different. Hence, low-level rules can have *minsup* while high-level rules are prevented from combinatorial explosion. The paper is organized into five sections. The new framework and major mining steps are described in section 2 and 3 respectively. Simulation results are reported in section 4 and the final section concludes the paper.

¹ Ack: supported by the H.K. Polytechnic University Research Studentship, project no. G-V722

2. A New Framework for Mining Generalized Association Rules

In Fig.1, the basic framework for mining *multiple-level frequent itemsets* is shown. It is efficient by running once the Apriori algorithm on the original transactions and scanning once the extended transactions. As the support at low concept level is implicitly smaller, finding rules with *minsup* at higher level requires first generating a list of itemsets with their support < *minsup*. We call them as *less frequent itemsets* and the support parameter as *sub-minimum support*, γ . Note here that the smaller the value of γ , the larger the potential to discover hidden high-level rules, but the slower the mining process. The concept generalization process, it is divided into two parts, namely, inter-itemset and intra-itemset generalizations. Firstly, the *less frequent itemsets* are divided into two groups, i.e., the *frequent itemsets* and the itemsets with $\gamma < minsup$. The latter group will then undergo the inter-itemset generalization. The results (*inter-generalized itemsets*) will be merged with the *frequent itemsets* and they will then be passed to the intra-itemset generalization. After itemset generalization, all subsets of the generalized itemsets are generated and any duplication is deleted because any subset of a large itemset must be large [1]. Those generalized itemsets and its subsets are *multiple-level less frequent itemsets*. They are finally filtered by multiple *minsup* to produce the *multiple-level frequent itemsets*. In this step, we have to count the support of each candidate itemset by scanning the extended transactions only once. As proposed in [9], we can pre-compute the ancestors for each item and add those ancestors that are in *multiple-level less frequent itemsets*. Following the same idea, we propose to pre-compute the generality of each item. This helps to speed up concept generalization and multiple *minsup*. With the final *multiple-level frequent itemsets*, it is a simple step to apply rule generation algorithm [1] to produce generalized association rules.

Fig. 1. Framework for mining multiple-level frequent itemsets

Fig. 2. Taxonomy of Drink

3. Major Mining Steps

3.1 Concept Generalization of Itemsets

Concept generalization of itemsets is a process of forming a general description of a class of objects (leaf-level itemsets) given a set of taxonomies. With reference to Fig.2, let us consider the following three itemsets $t1:(drink\ apple_juice)$, $t2:(drink\ milk)$, and $t3:(drink\ orange_juice)$. Since all itemsets are characterized by the same predicate *drink*, the conceptual distance [5] between them is exclusively determined by the distance between their arguments. Intuitively, $distance(t1, t3) < distance(t1, t2)$. Let us also consider the following generalizations $G(t_i, t_j)$ of two itemsets t_i and t_j : $G(t1, t3):(drink\ juice)$. While *orange_juice* is an instance, *juice* is a generalization with 2 instances. Thus, we define the degree of generality of an argument $g(a)$ as:

$$g(a) = \frac{\text{number of instances of "a"}}{\text{number of instances of the domain of "a"}} \quad (1)$$

e.g., $g(juice) = 2/7$. This definition applies also to the so-called linear (quantitative) variables [8]. To sum up, Let P_i be a predicate; a_i , b_i and c_i be the arguments of P_i :

$$\begin{aligned} t1: & (P_1 a_1) \& (P_2 a_2) \& \dots (P_i a_i) \dots \& \dots (P_n a_n) \\ t2: & (P_1 b_1) \& (P_2 b_2) \& \dots (P_i b_i) \dots \& \dots (P_n b_n) \\ G(t1, t2): & (P_1 c_1) \& (P_2 c_2) \& \dots (P_i c_i) \dots \& \dots (P_n c_n) \end{aligned}$$

Let $g(c_i)$ be the degree of generality of c_i . The estimation of the contribution of P_i to the dissimilarity D and similarity S between $t1$ and $t2$ are by the following functions:

$$D(t1, t2, P_i) = 0.5(\sum g(c_i))/n \quad (2)$$

$$S(t1, t2, P_i) = 1 - D(t1, t2, P_i) \quad (3)$$

We distribute the total contribution of P_i between similarity and dissimilarity in accordance with the generality degree of its arguments. The range of similarity and dissimilarity are 0.5 to 1 and 0 to 0.5 respectively, but their total must be 1. The estimation of conceptual distance between $t1$ and $t2$ (corresponding to G), as a function of D and S is $f(t1, t2, G) = D(t1, t2, G)/S(t1, t2, G)$. We take the minimum of $f(t1, t2, G)$ over all possible generalizations of $t1$ and $t2$ as the conceptual distance:

$$\text{concept_dist}(t1, t2) = \min_{G(t1, t2)} \{f(t1, t2, G(t1, t2))\} \quad (4)$$

3.1.1 Inter-itemset Generalization

When same predicate appears in two itemsets whose size and number of common items are equal, they can be inter-generalized. The common items should be extracted out first, and they will be added back after generalization. For instance, *Food D* is a common item and it is extracted. *Drink X* and *drink Y* are generalized to *drink Z*. *Food D* is being added back lastly. Thus, we have $t1:(drink\ X)(food\ D)$, $t2:(drink\ Y)(food\ D) \rightarrow t3:(drink\ Z)(food\ D)$. Unfortunately, the problem becomes complicated if there is

more than one item in each itemset. For example, we have two combinations in the following case:

Combination 1:

t1:(*drink apple_juice*)(*drink beer*)
t2:(*drink orange_juice*)(*drink VSOP*)
→ t3:(*drink juice*)(*drink alcoholic*)

Combination 2:

t1:(*drink apple_juice*)(*drink beer*)
t2:(*drink VSOP*)(*drink orange_juice*)
→ t3:(*drink any_drink*)(*drink any_drink*)

In this case, only one generalization is chosen as new itemset where its conceptual distance is minimum. For example, combination 1 is chosen for inter-generalization.

Combination 1:

$D(t1, t2, drink) = 0.5 * [2/7 + 4/7] / 2 = 0.21$
 $S(t1, t2, drink) = 1 - D(t1, t2, drink) = 0.79$
Conceptual distance = $0.21 / 0.79 = 0.27$

Combination 2:

$D(t1, t2, drink) = 0.5 * [7/7 + 7/7] / 2 = 0.5$
 $S(t1, t2, drink) = 1 - D(t1, t2, drink) = 0.5$
Conceptual distance = $0.5 / 0.5 = 1$

3.1.2 Intra-itemset Generalization

The problem of intra-itemset generalization is much simpler. For those items with common predicate, they are generalized. For instance, t1:(*drink X*)(*drink Y*)(*food D*) → t1a:(*drink Z*)(*food D*). The intra-generalized itemset does not replace the original one, i.e., itemsets t1 & t1a co-exist. The newly generated itemsets are then undergone intra-itemset generalization recursively. The following is an example:

<Case 1> t1:(*drink beer*)(*drink FOV*)(*drink XO*) → t1a:(*drink alcoholic*)

<Case 2> t1:(*drink beer*)(*drink FOV*)(*drink XO*) → t1b:(*drink alcoholic*)(*drink XO*)

<Case 3> t1:(*drink beer*)(*drink FOV*)(*drink XO*) → t1c:(*drink FOV*)(*drink alcoholic*)

<Case 4> t1:(*drink beer*)(*drink FOV*)(*drink XO*) → t1d:(*drink beer*)(*drink brandy*)

If a generalized itemset consists of an item and its ancestor, the itemset should be deleted. Since the support for an itemset t that contains both an item x and its ancestor x' will be the same as the support for the itemset $t-x'$ [9]. So, case 2 and 3 are deleted.

3.2 Multiple Minimum Supports

Now, we propose a method to calculate *minsup* for each itemset dynamically from its generality. Let t be an itemset, and x_i be an item in t . The generality of t is:

$$G(t) = \min[g(x_1), g(x_2), \dots, g(x_n)] \quad (5)$$

where the predicate of x_i can be different. Here, we define *lower-minimum support* and *upper-minimum support* for lowest level and highest level itemsets respectively.

Definition 1: The *lower-minimum support* β is a user-defined parameter. This is the *minsup* for the lowest concept level itemsets, and $\beta > \gamma$.

Definition 2: The *upper-minimum support* α is a user-defined parameter. This is the *minsup* for the highest concept level itemsets, and $\alpha > \beta$.

Based upon α and β , the *minsup*(t) for each itemset t is computed independently as:

$$minsup(t) = \beta + \{(\alpha - \beta) * g(t)\} \quad (6)$$

Here, $minsup(t)$ varies from β to α , depending on the generality $g(t)$. For example, if $\alpha = 20$, $\beta = 10$, and $g(t) = 0.2$, then $minsup(t) = 12$. Subsequently, each *multiple-level less frequent itemset* can be filtered by its *minsup* dynamically.

4. Simulation Results

We simulated two sets of experiment. First, the performance of the proposed algorithm with single *minsup* against the GAR algorithm [9] was compared. Second, the effectiveness of the proposed mining algorithm with multiple *minsup* was demonstrated. The synthetic data generation program was obtained from the IBM Almaden Research Center [http://www.almaden.ibm.com/cs/quest]. Details can be found in [1,9]. The default parameters for both experiments are shown in Table 1 & 2.

Table 1. Parameters for data generation

Parameter	Value
Number of transactions	100K
Average size of the transactions	5
Number of items	1,000
Number of roots	10
Number of levels	5

Table 2. Parameters for the experiments

Parameter	Value
Minimum support	10
Minimum confidence	80
Sub-minimum support (Concept Generalization only)	0.3
r-interest (Cumulate only)	1.1

In the first simulation, the performance of the GAR (Cumulate) algorithm with redundant rules pruned [9] and the proposed method (Concept Generalization) with single *minsup* was compared and shown in Fig.3. We changed the number of transactions from 100K to 400K. The number of rules found by both algorithms shows little changes, but the proposed method scales up better than the Cumulate algorithm in execution time. In the second simulation, the performance of the proposed method with multiple *minsup* was evaluated. We first ran the algorithm with single *minsup* at 10% and 20%. Then, we fixed the β at 10% and varied the α from 10% to 20%. The results are plotted in Figs.4&5. At $\alpha = 10\%$, the number of itemsets/rules found is exactly the same as that of single *minsup* fixed at 10%. At $\alpha = 20\%$, the number of itemsets/rules is smaller than single *minsup* with 20%, because not all itemsets' *minsup* are 20%. In fact, their *minsup* vary from 10% to 20%.

Fig. 3. Simulation Results I

Fig. 4. Number of itemsets discovered with different *upper-minimum supports*

Fig. 5. Number of rules discovered with different *upper-minimum supports*

5. Conclusions

We have introduced a new approach to the discovery of generalized association rules with multiple minimum supports. It eliminates some of the limitations of existing generalized or multiple level association rule mining algorithms, e.g., using single *minsup* or setting it at each level manually. More importantly, the number of scanning of database required by the proposed algorithm is approximately $k+1$ times of the number of original transactions plus one pass of the extended transactions, if the largest itemset being discovered is k -itemset; while other methods have to scan $k+1$ times of the number of extended transactions or more.

Reference

1. R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," *Proc. VLDB Conference*, Santiago, Chile, pp.487-499, Sept. 1994.
2. J. Han and Y. Fu, "Discovery of multiple-level association rules from large databases," *Proc. 21st VLDB Conference*, Zurich, Switzerland, pp.420-431, Sept. 1995.
3. J. Han and Y. Fu, "Mining multiple-level association rules in large databases," *IEEE Trans. on Knowledge and Data Engineering*, Vol.11, No.5, pp.798-805, Sept. 1999.
4. M. Klemettinen, et. al., "Finding interesting rules from large sets of discovered association rules," *Proc. 3rd Int. Conf. on Information and Knowledge Management*, pp.401-408, 1994.
5. Y. Kodratoff, and J. G. Ganascia, "Improving the generalization step in learning," *Machine Learning: An Artificial Intelligence Approach*, vol.2, LA, pp. 215-244, 1986.
6. Y. Kodratoff, and G. Tecuci, "Learning based on conceptual distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.10, No.6, pp.897-909, Nov. 1988.
7. B. Liu, W. Hsu, and Y. Ma, "Mining association rules with multiple minimum supports," *Proc. ACM SIGKDD Conf. on KDD*, San Diego, CA, pp. 337-341, August, 1999.
8. R. S. Michalski, and R. Stepp, "Learning by observation," *Machine Learning: An Artificial Intelligence Approach*, pp.163-190, 1983.
9. R. Srikant and R. Agrawal, "Mining generalized association rules," *Proc. of the 21st VLDB Conf.*, Zurich, Switzerland, pp.407-419, Sept. 1995.