Mining Two-Dimensional Optimized Association Rules and Hidden Patterns for Numeric Attributes

Zhi He, Shengfeng Tian, Houkuan Huang

School of Computer and Information Technology, Beijing Jiaotong University, Beijing, 100044, China

Abstract

Association rules can find the relation between attributes. If we focus on some certain attributes, optimized association rules can help us discover interesting characteristics of them. Optimized association rules contain uninstantiated attributes, and the optimization procedure is to determine the instantiations such that support, confidence or gain of the rules are maximized. In this paper, we generalize the optimized rules to including uninstantiated attributes in both antecedents and consequents of rules. Also, we replace confidence by interest, since confidence has some disadvantages as a measure. Consequently, we propose algorithms for finding optimized interest rules and optimized support rules. We test the performance of our algorithms on synthetic data sets and visualize the results by gray scale images. During this procedure, it can be noticed that optimized association rules can disclose different hidden patterns while the parameters are varying. Thus, we apply Fuzzy C-Means(FCM) to approximately find hidden patterns.

Keywords: optimized association rules, interest, dynamic programming, clustering

1. Introduction

Association rule was first presented in [1], and it was used to find correlation between attributes. The general form of association rules is as: $C_1 \rightarrow C_2$, where C_1 and C_2 are conjunctions over conditions, each of which is like $A_j=v_j$ or $A_j \in [l_j,u_j]$(v_j, l_j, u_j are the values in the domain of the attribute A_j). Every rule has its confidence and support value. The support of C_1 is the ratio of the number of tuples satisfying C_1 and the number of the whole tuples. The support of the rule, *sup*, is the support of condition $C_1 \wedge C_2$. The confidence of the rule, *conf*, is the ratio of the support of $C_1 \wedge C_2$ and the support of C_1. Mining association rules is to find all the rules satisfying the minimum support and confidence threshold.

1.1 Optimized Association Rules

The problem of optimized association rules is presented first by [2]. The optimized association rule, R, has the form $(A_1 \in [l_1,u_1]) \wedge C1 \rightarrow C2$, where A_1 is a numeric attribute, l_1 and u_1 are uninstantiated variables, C_1 and C_2 are conjunctions over instantiated conditions(that is, there are no uninstantiated variables in C_1 and C_2). The authors presented algorithms to determine l_1 and u_1 for such cases as: (a)maximize the confidence of R when $sup(A_1 \in [l_1,u_1])$ is more than the given minimum support threshold; (b)maximize the support of $A_1 \in [l_1,u_1]$ when $conf(R)$ is more than the given minimum confidence threshold. Optimized association rules is very useful for finding such intervals of attributes that form strong correlations with other conditions. Unfortunately, [2] has the disadvantage of not permitting uninstantiated variables in the consequence.

1.2 Correlation Measure

Although the correlation is more interesting for us, the association rules found with the support-confidence framework often reflect the coocurrence instead of correlation. Therefore, people tried to define other better measures. For example, [3] introduced a symmetric interestingness measure, *interest*, by applying chi-square test. It was defined as: $P(A,B)/(P(A)P(B))$.

1.3 Our Contributions

In this paper, we generalize the optimized association rule to allowing one uninstantiated numeric attribute in consequence. Moreover, we replace confidence by interest for finding more interesting optimized association rules. To achieve this goal, we first split the ranges of two numeric attributes into equal-depth intervals. As a result, the intervals of attributes intersect each other to form a grid. Based on the grid, we propose two algorithms with dynamic

programming to find optimized interest rules and optimized support rules.

We notice that the instantiated range of an optimized association rule is usually concerned with some hidden pattern. However, when there are more than one hidden patterns, the rule is not able to represent all of them, because there is always one optimized rule. We also notice that the optimized association rule will suggest different hidden patterns when the parameter changes continuously. That is, we can get the rules representing all hidden patterns. Thus, a simple clustering method can be used to find multi-hidden patterns based on the midterm experiment results.

2. Related Work

The object of [4] was to mine the optimized association rule, the antecedence of which contained two numeric attributes. The authors developed algorithms for computing the rectangular and admissible regions that maximize the gain, support or confidence, respectively. [5] developed an algorithm for finding optimized support rule that contains disjunctions over intervals of the same attribute in the antecedence. [6] presented more efficient algorithms to find optimized gain rules. For the rule containing a single numeric attribute, the algorithm had the time complexity O(nk), where n is the number of values in the domain of the uninstantiated attribute. For the rule containing two numeric attributes, the authors presented an approximation algorithm based on dynamic programming. [7] generalized the optimized association rules problem in three ways.

The remaining sections of this paper are arranged as follows: the optimized support rules and optimized interest rules are defined in detail in section 3. The algorithms for discovering optimized interest rules and optimized support rules are presented in section 4. Before that, the data must be pre-processed. In section 5, the algorithms are performed on synthetic data sets, and the results are visualized by gray scale images. We conclude the paper and clarify the future work in section 6.

3. Problem Formulation

In this paper, the optimized association rule is permitted containing one numeric attribute in antecedence and consequence, respectively. The general form is:

$$A_i \in [l_i, u_i] \wedge C_1 \rightarrow A_j \in [l_j, u_j] \wedge C_2 \qquad \text{(R)}$$

A_i and A_j are numeric attributes, l_i, u_i and l_j, u_j are the values in the domains of them, respectively. C_1 and C_2 are conjunctions over conditions, and they are not allowed to contain uninstantiated variables. For simplicity, R can be written as:

$$A_i \in [l_i, u_i] \rightarrow A_j \in [l_j, u_j] \qquad \text{(R')}$$

In fact, it is very simple for our algorithms to generalize from R' to R. The support of R' is sup($A_i \in [l_i, u_i] \wedge A_j \in [l_j, u_j]$). In fact, a region represents a rule. As a result, the support of R' can be written as: sup($[l_i, u_i], [l_j, u_j]$). For convenience, let sup(C) be the number of tuples instead of the ratio. Thus, the interest of R is ins($[l_i, u_i], [l_j, u_j]$)= sup($[l_i, u_i], [l_j, u_j]$)N/(sup($A_i \in [l_i, u_i]$)sup($A_j \in [l_j, u_j]$)), where N is the number of whole tuples. Let $0<\eta<1$ be the user-specified minimum support threshold and $\lambda>1$ the interest threshold. If sup($[l_i, u_i], [l_j, u_j]$)$\geq\eta$N, we call R' the ample rule. Among ample rules, an optimized interest rule maximizes ins($[l_i, u_i], [l_j, u_j]$). Similarly, R' is interesting if ins($[l_i, u_i], [l_j, u_j]$) $\geq \lambda$. Among interesting rules, an optimized support rule maximizes sup($[l_i, u_i], [l_j, u_j]$). Our goal is to instantiate A_i and A_j so that either interest or support of R' is maximized.

4. Optimized Association Rules

4.1 Data Preparation

As the authors in [2] did, we split the ranges of the two numeric attributes into equal-depth intervals. The number of the intervals was pre-specified by the user. In general, the support of each interval is far less than η. The two reasons why we adopt equal-depth split instead of equal-width split are that: (a) the former leads to less information loss[2] than the latter; (b) the intervals formed by the former are more appropriate for our algorithms than the latter. Let the sequence, B_1, B_2, …, B_n, be the intervals formed by equal-depth split without overlap, where $B_i=[x_i,y_i]$, $x_i \leq y_i < x_{i+1}$(i=1, …, n). Thus, $A_i \in [l_i, u_i]$ can be approximately written as: $A_i \in [x_s^{(i)}, y_t^{(i)}]$, if $l_i \in B_s = [x_s, y_s]$ and $u_i \in B_t = [x_t, y_t]$. The data have to be ordered before split, and this operation is very much time-consuming in the case that the volume of data is very large. However, the sampling and parallel bucketing techniques in [2] can be adopted to alleviate this difficulty.

After the ranges are split, the boundaries of intervals intersect to form a two-dimensional grid, H. The size of H is K=mn, where n and m is the number of intervals of A_i and A_j, respectively. Suppose ($[x_s^{(i)}, y_s^{(i)}], [x_t^{(j)}, y_t^{(j)}]$) be the region of grid cell (s,t). The next step is to scan the data set to count $G_{(s,t)}$, the number of tuples that fall into the cell (s,t). We have sup(s,t)= $G_{(s,t)}$ and ins(s,t)= $G_{(s,t)}$N/

(sup($A_j \in [x_t^{(j)}, y_t^{(j)}]$)sup($A_i \in [x_s^{(i)}, y_s^{(i)}]$)). Due to the equal-depth split, sup($A_j \in [x_t^{(j)}, y_t^{(j)}]$)=N/m, (t=1, … ,m) and sup($A_i \in [x_s^{(i)}, y_s^{(i)}]$)=N/n □ (s=1, … ,n). Then we have ins(s,t)= $G_{(s,t)}$nm/N. Let ($[x_u^{(i)}, y_{u+p-1}^{(i)}], [x_v^{(j)}, y_{v+q-1}^{(j)}]$)(1≤u≤u+p-1≤n□1≤v≤ v+q-1≤m) be an arbitrary region S of H. We have

$$\sup(S)=G_S, \text{ins}(S)=G_S mn/(Npq) \quad (1)$$

where $G_S = \sum_{t=v}^{v+q-1} \sum_{s=u}^{u+p-1} G(s,t)$. To find the optimized interest rule, we have to find such S that S□ $\arg\max_I(ins(I))$, and sup(I)≥Nη. Similarly, to find the optimized support rule, we have to find such S that S□ $\arg\max_I(\sup(I))$, and ins(I)≥λ.

4.2 Optimized Support Rules

We can find the optimized support rule in two steps. Firstly, we find the optimized support rule in the region between row r_1 and r_2($1 \le r_1 \le r_2 \le m$). Secondly, we select the maximum support rule among the optimized support rules obtained from all possible pairs of r_1 and r_2. Based on the results of the first step, the second step is straightforward. Let **x** be a n-dimensional vector, each element of which is the sum of the support of a column of cells between r_1 and r_2. Actually, let **r** be a subvector of **x**, then the region with the maximum support corresponds to the **r** with the maximum sum. In our algorithm, such **r** is found by dynamic programming[4][8]. In detail, let maxsum(**x**) be the subvector with the maximum sum and ins(maxsum(**x**)) <λ. If ins(**x**)<λ, we have (in c language) maxsum(**x** = sum(maxsum(**x**(1:*l*-1)))> sum(maxsum(**x**(2:*l*)))? **x**(1:*l*-1) : **x**(2:*l*)(*l* is the length of **x**). Otherwise, we have maxsum(**x**)= **x**. Obviously, the primary part of the algorithm is the procedure named "maxsum", which is called recursively. The time complexity of the algorithm is O(n), and that of computing **x** is O(mn). Finally, we get the optimized support rule in $O(m^3n)$. We should notice that N is not involved in the time complexity of the algorithm, owing to the data preparation. It allows the algorithm performing efficiently on a large magnitude of data set.

4.3 Optimized Interest Rules

In this section, the idea that we use to find the optimized interest rule is similar to that in the previous section. Of course, there is a little difference. According to the formula (1), maximizing ins(S) implies maximizing G_S/p because m, n, q are constant. That is, we are going to find such subvector of **x** that maximizes the mean. Let maxmean(*l*) denote the subvector with the maximum mean among all the subvectors (**x**(k),…,**x**(*l*))(k≤*l*), and maxmean(≤*l*) the subvector with the maximum mean among all the subvectors (**x**(k),…,**x**(t))(k≤t≤*l*). Then, we have (in c language) maxmean(*l*) = mean([maxmean(*l*-1),x(*l*)])≥ x(*l*)? [maxmean(*l*-1),x(*l*)]:x(*l*); maxmean(≤ *l*) = mean(maxmean(≤ *l*-1)) ≥ mean(maxmean(*l*))? maxmean(≤*l*-1): maxmean(*l*). For saving the space, we will not list the algorithm in detail.

5. Experiments

5.1 Generation of Data

We will test the performance of our algorithms on two data sets generated randomly. Each data set comprises of two numeric attributes, A_1 and A_2. Both of their ranges are between 0 and 1. The values of A_1 are generated by uniform distribution, and the values of A_2 are generated according to given rules and the value of A_1. If the value of A_1 falls into the region of some rule, the value of A_2 will fall into that region with a probability, *p*. Otherwise, the value is generated by uniform distribution. We pre-specified three rules, from which one data set, D, is produced.

5.2 Visualization

After the data preparation discussed in section 4.1, we get a grid formed by intersected boundaries of intervals. We can easily map the grid with counts $G_{(s,t)}$ into a gray scale image: the horizontal axis is mapped from the intervals of A_2, the vertical axis from the intervals of A1, and gray scale values from the counts of cells. The numbers in the fig. 1 represent the regions covered by the corresponding pre-specified rules(hidden patterns).

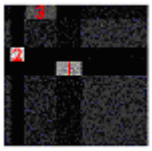
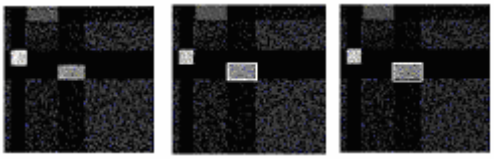


Fig. 1 The image corresponding to hidden rules

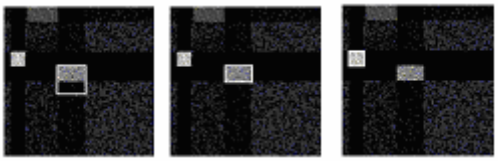
5.3 Optimized Region

The algorithm in section 4.3 is performed three times on D withη= 0.05, 0.08, 0.1, respectively. And the algorithm in section 4.2 is performed three times on D withλ=3, 5, 8. As a result, the optimized interest rules are represented by white rectangles in fig. 2. In fig. 3, the rectangles represent those optimized support rules.



(a)η=0.05 (8.6) (b)η=0.08 (5.6) (c)η=0.1 (4.4)

Fig. 2 The optimized interest region of Fig. 1 (the numbers in brackets are the interest values)

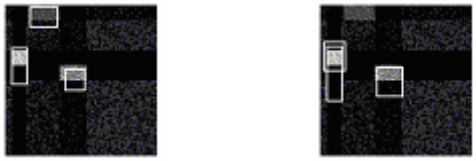


(a)λ=3 (11%) (b)λ=5(9.87%) (c)λ=8(7.86%)

Fig. 3 The optimized support region of Fig. 1 (the numbers in brackets are the support values)

5.4 Discovery of Hidden Pattern

Actually, one optimized rule can only discover the region of one hidden pattern. Sometimes, people are desired to find all the hidden patterns. To achieve this object, we apply the Fuzzy C-Means(FCM)[9] algorithm to cluster the midterm results of experiment, which are the optimized region for all possible pairs of A_1 and A_2. Fig. 4 shows the three centers of clusters (we set the number of clusters 3).



(a)η=0.05 (8.6) (b)η=0.08(5.6)

Fig.4 The centers of clusters

In fig. 4, (a) approximately shows the outlines of three hidden patterns, while (b) doesn't include the region 3 in fig.1. The reason is that the support of the region 3 is below 0.08, and the regions expanded from it have such a low interest that they are filtered out from the midterm results. Therefore, it is natural that region 3 is not included in clusters.

6. Conclusions

In this paper, we generalized the optimized rule to allow one uninstantiated numeric attribute in antecedence and consequence. Moreover, for finding more interesting optimized rules, we replaced the confidence by interest. Based on the idea of Dynamic Programming, we presented two efficient algorithms to mining optimized interest and support rules. To test the performance of our algorithms, we performed them on synthetic data set, and visualized the results by gray scale images. We then noticed that the optimized rules could disclose different hidden patterns when we tuned the parameters. To find all hidden patterns, we applied FCM algorithm on the midterm results of experiment. As a result, the centers of clusters represent well the hidden patterns.

In this paper, the instantiated regions of all optimized rules are rectangular. However, in practice, the admissible regions often bring more information to users[4]. In future, we will further work on how to find optimized rules with admissible regions.

7. References

[1] Agrawal R., Imielinski T., and Swami A., "Mining Association Rules between Sets of Items in Large Databases," *Proc. ACM SIGMOD Conf. Management of Data*, pp. 207-216, May 1993.

[2] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama, "Mining Optimized Association Rules for Numeric Attributes," *Proc. ACM SIGACT-SIGMOD-SIGART Symp. Principles of Database Systems*, June 1996.

[3] S. Brin, R. Motwani, and C. Silverstein. "Beyond market baskets: Generalizing association rules to correlations," In *Proc. Of the ACM SIGMOD*, pages 265–276, 1997.

[4] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama, "Data Mining Using Two-Dimensional Optimized Association Rules:Scheme, Algorithms, and Visualization," *Proc. ACM SIGMOD Conf. Management of Data*, June 1996.

[5]R. Rastogi, and K. Shim, "Mining Optimized Support Rules for numeric Attributes," *Proc. Int'l Conf. Data Eng*., 1999.

[6] S. Brin, R. Rastogi, and K. Shim, "Mining optimized gain rules for numeric attributes," *Knowledge and Data Engineering, IEEE Transactions on* , Volume: 15, Issue: 2, March-April, Pages:324 – 338 2003

[7] R. Rastogi, and K. Shim, "Mining optimized association rules with categorical and numeric attributes," *Knowledge and Data Engineering, IEEE Transactions on* , Volume: 14, Issue: 1, Jan.-Feb. Pages:29 – 50 2002

[8]J. Bentely, "Programming pearls," *Communications of the ACM*, 27(27):865:871, Septermber, 1984

[9] J. C. BEZDEK, "Pattern Recognition With Fuzzy Objective Function Algorithms," Plenum Press, New York, NY. 1981.