ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.10, October 2007, pp.2528−2537
DOI: 10.1360/jos182528

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种挖掘数值属性的二维优化关联规则方法*

贺　志+，田盛丰，黄厚宽

(北京交通大学 计算机与信息技术学院 人工智能实验室,北京　100044)

An Approach to Mining Two-Dimensional Optimized Association Rules for Numeric Attributes

HE Zhi+,　TIAN Sheng-Feng,　HUANG Hou-Kuan

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

+ Corresponding author: Phn: +86-10-51683602, Fax: +86-10-51840526, E-mail: hezhidragon@126.com

He Z, Tian SF, Huang HK. An approach to mining two-dimensional optimized association rules for numeric attributes. *Journal of Software*, 2007,18(10):2528−2537. http://www.jos.org.cn/1000-9825/18/2528.htm

Abstract: Optimized association rules are permitted to contain uninstantiated attributes. The optimization procedure is to determine the instantiations such that some measures of the rules are maximized. This paper tries to maximize interest to find more interesting rules. On the other hand, the approach permits the optimized association rule to contain uninstantiated numeric attributes in both the antecedence and the consequence. A naive algorithm of finding such optimized rules can be got by a straightforward extension of the algorithm for only one numeric attribute. Unfortunately, that results in a poor performance. A heuristic algorithm that finds the approximate optimal rules is proposed to improve the performance. The experiments with the synthetic data sets show the advantages of interest over confidence on finding interesting rules with two attributes. The experiments with real data set show the approximate linear scalability and good accuracy of the algorithm.

Key words: optimized rule; association rule; interest; heuristic; approximation

摘　要: 优化关联规则允许在规则中包含未初始化的属性.优化过程就是确定对这些属性进行初始化,使得某些度量最大化.最大化兴趣度因子用来发现更加有趣的规则;另一方面,允许优化规则在前提和结果中各包含一个未初始化的数值属性.对那些处理一个数值属性的算法进行直接的扩展,可以得到一个发现这种优化规则的简单算法.然而这种方法的性能很差,因此,为了改善性能,提出一种启发式方法,它发现的是近似最优的规则.在人造数据集上的实验结果表明,当优化规则包含两个数值属性时,优化兴趣度因子得到的规则比优化可信度得到的规则更有趣.在真实数据集上的实验结果表明,该算法具有近似线性的可扩展性和较好的精度.

关键词: 优化规则;关联规则;兴趣度;启发式方法;近似

中图法分类号: TP311　　　**文献标识码**: A

Association rule was first presented in Ref.[1] and used to find relationships between attributes. The general form of association rules is like: $C_1 \rightarrow C_2$, where C_1 and C_2 are called antecedence and consequence, respectively.

* Supported by the National Natural Science Foundation of China under Grant Nos.60442002, 60443003 (国家自然科学基金)

Received 2005-07-24; Accepted 2006-06-06

Both of them are conjunctions over conditions. The form of a condition is like $A_j=v_j$ (for categorial or numeric attribute)or $A_j\in[l_j,u_j]$ (for numeric attribute), where A_j is an attribute and, v_j, l_j and u_j are the values in the domain of A_j. The quality of a rule is often described by *support* and *confidence*. In this paper, we denote them with *sup* and *conf*, respectively. The support of the rule equals to the support of condition $C_1\wedge C_2$. The support of a condition is the ratio of the number of tuples satisfying to the number of the whole tuples. The confidence of the rule is the ratio of the support of $C_1\wedge C_2$ to the support of C_1. Mining association rules is to find all the rules satisfying the minimum support and confidence thresholds.

The problem of finding optimized association rules was presented first by Ref.[2]. An association rule to be optimized usually has the form $(A_1\in[l_1,u_1])\wedge C_1\rightarrow C_2$, where A_1 is a numeric attribute, l_1 and u_1 are uninstantiated variables (that is, they are not assigned with any values). C_1 and C_2 contain only instantiated conditions (that is, there are no uninstantiated variables). The authors proposed the algorithms for determining values for l_1 and u_1 to maximize confidence, support or gain with some thresholds satisfied at the same time. The optimized association rule is very useful for finding such intervals of attribute that form strong correlations with other conditions. For example, it is known that there are some correlations between income and education level. But we are not certain about what range of income makes strong correlation with college experience. In this case, the optimized confidence rule *income*$\in[l_1,u_1]\rightarrow$*college* will help us to find the answer.

In this research, we generalize the optimized association rule into allowing uninstantiated numeric attributes in both antecedent and consequence. Besides, we replace confidence with interest for finding more interesting rules. A naive algorithm for two numeric attributes can be got by a straightforward extension of methods for rules with one numeric attribute. However, this results in poor performance with respect to the number of buckets. Therefore, we present a heuristic method to get approximate results to improve the performance.

The remaining sections of this paper are arranged as follows: in Section 1, we discuss the related work, and in Section 2, the optimized support rules and the optimized interest rules are defined. The heuristic algorithm, HFOIR, is presented in Section 3. In Section 4, the performance of the algorithm is evaluated with synthetic and real data sets. Finally, we conclude the paper and clarify the future work in Section 5.

1 Related Work

The objective of Ref.[3] was to mine the optimized association rule, the antecedence of which contains two numeric attributes. The authors developed the algorithms for computing the rectangular and admissible regions that maximize the gain, support or confidence, respectively. In Ref.[4], R. Rastogi and K. Shim developed an algorithm for finding the optimized support rule that contains disjunctions over intervals of the same attribute in the antecedence. In Ref.[5], they generalized the optimized association rule problem in three ways by permitting the antecedence: (a) to contain disjunctions over uninstantiated attributes; (b) to contain an arbitrary number of uninstantiated attributes; (c) to contain uninstantiated attributes that can be either categorical or numeric. The authors in Ref.[6] presented more efficient algorithms to find the optimized gain rules. They generalized the optimized gain association rule problem by permitting the antecedences of rules to contain upto k disjunctions over one or two uninstantiated numeric attributes. For one attribute, the algorithm has 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. Reference [7] also dealt with the problem of optimizing disjunction association rules. The authors presented the first polynomial time algorithm for the problem of finding such a region maximizing support and meeting a minimum cumulative confidence threshold. Running the algorithm on a small random sample was proposed as a means of obtaining near optimal results with a high probability. Theoretical bounds on sufficient sample size to achieve a given performance level were proved, and rapid convergence on synthetic and real-world data was validated experimentally.

2 Problem Formulation

In our research, association rules are permitted to contain uninstantiated attributes in antecedence and consequence. It is different from the problem defined in Refs.[5,6], where the optimized rule was permitted to contain two numeric attributes only in antecedence. The general form is as R: $A_1\in[l_1,u_1]\wedge C_1\rightarrow A_2\in[l_2,u_2]\wedge C_2$, where A_1 and A_2 are uninstantiated numeric attributes and, l_1, u_1 and l_2, u_2 are unistantiated variables. C_1 and C_2 are conjunctions over conditions and they are not allowed to contain uninstantiated variables. For simplicity, R can be written without loss of generality as R': $A_1\in[l_1,u_1]\rightarrow A_2\in[l_2,u_2]$.

Different from confidence or gain measure used in Refs.[2−5], we optimize the *rule interest* and *support* in this paper. Brin *et al* first proposed *rule interest* in Ref.[8]. This metric was defined to be the ratio between the joint probability of two variables with respect to their expected probabilities under the independence assumption. The *interest* of R' is defined as $ins(R')=\sup(R')/(\sup([l_1,u_1])\sup([l_2,u_2]))$. The authors argued that the *interest* measure was preferable as it directly captured dependence, as opposed to confidence which considered directional implication. As a result of measuring significance of dependence via the chi-squared test for independence, they reduced the mining problem to the search for a border between dependent and independent itemsets in the lattice. In contrast, we try to find the rule maximizing interest or support instead in this paper.

Definition 1. Let $0<\eta<1$ be the user-specified minimum support threshold. If $\sup(R')\geq\eta$, we call R' the ample rule. Among all ample rules, the optimized interest rule (OIR) maximizes $ins(R')$.

Definition 2. Let $\lambda>1$ be the user-specified minimum interest threshold. R' is interesting if $ins(R')\geq\lambda$. Among all interesting rules, the optimized support rule (OSR) maximizes $\sup(R')$.

In short, our goal is to find OIR and OSR. That is, we aim at the instantiation of $[l_1,u_1]$ and $[l_2,u_2]$ so that either interest or support of R' is maximized.

3 FOIR: Finding Optimized Interest Rules

In this section, we first transform the problem of finding OSR in a two-dimensional space into that of finding a particular region in a two-dimensional grid. Then, the problem is simplified into permitting the uninstantiated variables to appear only in the antecedence, that is, we instantiate the variables in consequence with some constant values. By doing this, the grid is simplified into a vector. In this case, we adapt the algorithm that was proposed in Ref.[2] to optimize the confidence rule to find such a subvector from a vector that maximizes its weighted average and has its sum satisfying some threshold. Finally, the original problem is resumed by a straightforward extension of the method for rules with one numeric attribute. But it results in a poor performance. A heuristic method, with which the approximate results are got, is proposed to improve the performance. We discuss the details of the three parts in the following subsections.

3.1 Transformation: Bucketing

As a preparation of our algorithm, we split the ranges of two numeric attributes into equal-width intervals (that is, the size of every interval is the same). We don’t split them into equal-depth intervals (that is, the number of tuples falling into every interval is the same) because it needs sorting operation and it is very time consumable. The numbers of the intervals on two attributes, v_1 and v_2, are pre-specified by user. Then the width of every interval on A_1 is $w_1=(U_1-L_1)/v_1$, where U_1 and L_1 are the upper and lower bounds of the domain of A_1. Similarly, the width of each interval on A_2 is $w_2=(U_2-L_2)/v_2$. Let $I_i^{(1)}$ denote the i-th interval on A_1 and its range is $[L_1+(i-1)w_1,L_1+iw_1,]$. Using these intervals, the antecedence in R' can be approximately represented as: $A_1\in[s_1,s_2]$ $(1\le s_1\le s_2\le v_1)$, if $l_1\in I_{s_1}^{(1)}$ and $u_1\in I_{s_2}^{(1)}$ hold. In a similar way, R' is rewritten as $[s_1,s_2]\rightarrow[t_1,t_2]$ if the consequence can be written as $A_2\in[t_1,t_2]$ $(1\le t_1\le t_2\le v_2)$.

The intervals on two attributes intersect to form a two-dimensional grid denoted with H. The size of H is $v_1\times v_2$. A cell of H, c, can be represented by a pair of integers, (s,t) $(1\le s\le v_1$ and $1\le t\le v_2)$, where s is the horizontal coordinate of c on H and t is the vertical coordinate. The value of c is the number of tuples falling into c. Further, a rectangular region, C, can be represented by two pairs of integers, $([s_1,s_2],[t_1,t_2])$, where $1\le s_1\le s_2\le v_1$ and $1\le t_1\le t_2\le v_2$. In fact, (s_1,t_1) denotes the leftmost bottom cell of C, and (s_2,t_2) denotes the rightmost top one. The boundary of C is $([L_1+(s_1-1)w_1,L_1+s_2w_1,],[L_2+(t_1-1)w_2,L_2+t_2w_2,])$. In the same way, the cell c can also be denoted by $([s,s],[t,t])$.

Now, R' can be approximately denoted by a region C of H, $([s_1,s_2],[t_1,t_2])$, if $l_1\in I_{s_1}^{(1)}$, $u_1\in I_{s_2}^{(1)}$, $l_2\in I_{t_1}^{(2)}$, and $u_2\in I_{t_2}^{(2)}$. The *interest* of R' can be approximately rewritten as:

$$ins(R')=ins(C)=N\sum_{j=t_1}^{t_2}\sum_{i=s_1}^{s_2}n_{i,j}\Bigg/\sum_{j=1}^{v_2}\sum_{i=s_1}^{s_2}n_{i,j}\sum_{i=1}^{v_1}\sum_{j=t_1}^{t_2}n_{i,j} \tag{1}$$

Where $n_{i,j}$ is the value of cell (i,j). N is the number of the whole tuples. Thus, to find OIR in Definition 1 is to find $\arg\max\limits_{\sup(C)\ge\eta}(ins(C))$. To find OSR in Definition 2 is to find $\arg\max\limits_{ins(C)\ge\lambda}(\sup(C))$.

3.2 Simplification: Finding OIRs with one numeric attribute

In this subsection, we simplify the problem into that allowing the rule to contain uninstantiated variables only in antecedence. Formally, R' is rewritten as R^*: $[s_1,s_2]\rightarrow[T_1,T_2]$, where T_1 and T_2 are constant integers $(1\le T_1\le T_2\le v_2)$. As a result, formula 1 is simplified as $N\sum_{j=T_1}^{T_2}\sum_{i=s_1}^{s_2}n_{i,j}\Big/\sum_{j=1}^{v_2}\sum_{i=s_1}^{s_2}n_{i,j}\sum_{i=1}^{v_1}\sum_{j=T_1}^{T_2}n_{i,j}$. After removing the constant items from it, we get

$$\sum_{j=T_1}^{T_2}\sum_{i=s_1}^{s_2}n_{i,j}\Bigg/\sum_{j=1}^{v_2}\sum_{i=s_1}^{s_2}n_{i,j} \tag{2}$$

Next we give the following theorem:

Theorem 1. Let vector $\boldsymbol{p}=(p_1,\dots,p_{v_1})$, where $p_i=\sum_{j=T_1}^{T_2}n_{i,j}$. Let vector $\boldsymbol{r}=(r_1,\dots,r_{v_1})$ is the weight of $\boldsymbol{p}$, where $r_i=m_i/M$, $m_i=\sum_{j=1}^{v_2}n_{i,j}$ and $M=\min(m_i)$. To find OIR is to find such a subvector from $\boldsymbol{p}$ that maximizes its weighted average and has its sum satisfying a threshold.

Proof: As discussed in Subsection 3.1, to find OIR is to maximize Eq.(2) and satisfy $\sum_{j=T_1}^{T_2}\sum_{i=s_1}^{s_2}n_{i,j}\ge\eta N$. Substituting the items in Eq.(2) with the element of $\boldsymbol{p}$ and $\boldsymbol{r}$, we get $\frac{1}{M}\sum_{i=s_1}^{s_2}p_i\Big/\sum_{i=s_1}^{s_2}r_i$. Since M is fixed for a particular H, finding OIR is equal to maximizing

$$\sum_{i=s_1}^{s_2}p_i\Bigg/\sum_{i=s_1}^{s_2}r_i \tag{3}$$

and satisfying

$$\sum_{i=s_1}^{s_2}p_i\ge\eta N \tag{4}$$

Eq.(3) is the weighted average of the subvector, $(p_{s_1},\dots,p_{s_2})$ and the left item of Eq.(4) is its sum.

In Ref.[2], the authors presented an algorithm to find the optimized confidence rule with linear time complexity. They defined the support of R' as sup(A_1). And they viewed the sup(A_1) and sup(A_1A_2) of R' as the horizontal and vertical axes of a two-dimensional space, respectively. Then the initiation of the optimized confidence rule corresponds to finding a pair of points with the slope maximized and the difference of their horizontal coordinates satisfying the threshold. For our problem, we adapt the algorithm to finding OIRs by redefining the meanings of the points. For a vector $\boldsymbol{p}=(p_1,\ldots,p_{v_1})$, it corresponds to a set of points, $\{q_0,\ldots,q_{v_1}\}$. If the horizontal and vertical coordinates of q_i are denoted by x_i and y_i, respectively, they satisfy $y_i=\begin{cases}0, & i=0\\ \sum_{j=1}^{i}p_j, & \text{else}\end{cases}$ and $x_i=\begin{cases}0, & i=0\\ \sum_{j=1}^{i}r_j, & \text{else}\end{cases}$. The slope between q_i and q_j ($i<j$) is $l=(y_j-y_i)/(x_j-x_i)=\sum_{k=i+1}^{j}p_k\Big/\sum_{k=i+1}^{j}r_k$. It is identical with Eq.(3). Thus, we get an algorithm of finding OIRs for one uninstantiated numeric attribute with the linear time complexity of $O(v_1)$.

3.3 Resume: Finding OIRs with two numeric attributes

We return to the original problem with two numeric attributes. For the rule with two uninstantiated numeric attributes, a naive algorithm to find OIRs, named NFOIR, can be got by a straightforward extension of the algorithm discussed in the last subsection. The variables, t_1 and t_2, in formula 1 are substituted with a possible pair of row. Since there are v_2^2 pairs of rows, NFOIR has the time complexity, $O(v_1 v_2^2)$.

A heuristic method is proposed to improve this poor performance. The heuristic idea is based on our observation on experiments. For instance, it is assumed that a region d of H is denser (the density of a region is the ratio of the sum of all the cells to the number of the cells) than any other region with the same size. A region G contains d. Another region G' does not contains d. It is noticed that the range with the maximum average in G is often smaller than that in G'. We give the definition of *valid region* to describe the regions containing dense sub-regions.

Definition 3. A region in H, denoted with G, is bounded between the t_1-th row and the t_2-th row ($t_1\le t_2$). The super region of G in H, G', is bounded between the t_1-th row and the (t_2+1)-th row. The ranges of OIRs found in G and G' are ($[s_1,s_2],[t_1,t_2]$) and ($[s_1',s_2'],[t_1,t_2+1]$), respectively. If either of the following two conditions

$$[s_1,s_2]\cap[s_1',s_2']>\delta v_1 \tag{5}$$

$$s_2-s_1\ge s_2'-s_1' \tag{6}$$

doesn't hold, G is a valid region for G'.

Based the valid regions, the heuristic algorithm (HFOIR) is shown in Fig.1. The sentences between the lines 5 and line 15 are used to find all valid regions. Its time complexity is $O(v_1v_2)$. The time complexity of line 16 is $v_1\sum_{i=1}^{vrn}(vr[i,2]-vr[i,1]+1)^2$. Generally, $\sum_{i=1}^{vrn}(vr[i,2]-vr[i,1]+1)^2=v_2^2$, since $\sum_{i=1}^{vrn}(vr[i,2]-vr[i,1]+1)=v_2$ and $vrn=v_2$ hold. Therefore, HFOIR gets a much better time complexity than NFOIR.

The idea that we use to find OSRs is similar to that used in finding OIRs. We can also adapt the algorithms 4.3 and 4.4 in Ref.[2] to find such a subvector that maximizes its sum and having its weighted average satisfying a threshold. For saving space, we will not discuss the detail.

```
Input: $H$, $v_1$, $v_2$;
Output: the region that corresponds to the OIR.
Function HFOIR
1. $vr$=[]; %the set of the bounds of valid regions
2. $i$=1;
3. $j$=1;
4. $vrn$=0; % the number of valid regions
5. while $i \le v_2$
6.     Check ([1, $v_1$], [$i$, $j$]) to be valid region or not;
7.     if yes
8.         $vrn$++;
9.         $vr[vrn]$=[$i$, $j$];
10.        $i$=$j$+1;
11.        $j$=$j$+1;
12.    else
13.        $j$=$j$+1;
14.    endif
15. endwhile
16. Apply NFOIR in every valid region;
17. Get the region with maximum weighted average from the results of Step 16
```

Fig.1 The pseudocode of HFOIR

4 Experiments

4.1 Experiments with synthetic data sets

In this subsection, we run HFOIR with synthetic data sets to show the advantages of *interest* as an evaluation criteria over *confidence* on finding the interesting rules with two numeric attributes.

4.1.1 Generation of data

First, we prespecify some rules, by which two synthetic data sets are generated. Both of the data sets include 50000 tuples and comprise of two numeric attributes, A_1 and A_2. The ranges of both of attributes are between 0 and 1. For a tuple, the value of A_1 is first generated by a uniform distribution. Then the value of A_2 is generated according to the value of A_1 and the corresponding rule. That is, if the value of A_1 falls into a region represented by some rule, the value of A_2 will fall into the same region with a probability, p. Otherwise, the value is generated by a uniform distribution. In this experiment, the prespecified rules (the supports, confidences, and interests are computed after the data have been generated) are as follows:

$$A_1 \in [0.4,0.5] \rightarrow A_2 \in [0.3,0.4] \ (p=100\%, \text{sup}=10\%, conf=100\%, ins=5.71) \qquad \text{(Rule 1)}$$

$$A_1 \in [0.3,0.4] \rightarrow A_2 \in [0.06,0.08] \ (p=80\%, \text{sup}=7.6\%, conf=81.2\%, ins=8.83) \qquad \text{(Rule 2)}$$

$$A_1 \in [0,0.1] \rightarrow A_2 \in [0.1,0.3] \ (p=50\%, \text{sup}=6.3\%, conf=61.6\%, ins=2.94) \qquad \text{(Rule 3)}$$

$$A_1 \in [0,0.2] \rightarrow A_2 \in [0.1,0.3] \ (p=50\%, \text{sup}=14.8\%, conf=75.3\%, ins=1.98) \qquad \text{(Rule 4)}$$

$$A_1 \in [0.1,0.3] \rightarrow A_2 \in [0.1,0.2] \ (p=80\%, \text{sup}=16.6\%, conf=83.6\%, ins=3.09) \qquad \text{(Rule 5)}$$

The first data set, *DS*1, is generated by Rules 1~3. The second data set, *DS*2, is generated by Rules 4 and 5 that overlap each other on both attributes. During the procedure of producing the data of *DS*2, if the value of A_1 falls into the intersection, the rule by which the value of A_2 is generated depends on the ratio of p. That is, if the value of A_1 falls into [0.1 0.2], the ratio of the probability of applying Rule 4 to that of applying Rule 5 is 5/8. After the data have been produced, we get the rule intersected by Rules 4 and 5,

$$A_1 \in [0.1\ 0.2] \rightarrow A_2 \in [0.1\ 0.2] \ (\text{sup}=8.1\%, conf=84.4\%, ins=3.12) \qquad \text{(Rule 6)}$$

and the rule united by Rules 4 and 5,

$$A_1 \in [0\ 0.3] \rightarrow A_2 \in [0.1\ 0.3] \ (\text{sup}=23.6\%, conf=78.8\%, ins=2.07) \qquad \text{(Rule 7)}$$

4.1.2 Experiment results of HFOIR and FOCR

HFOIR is run with different values of η on DS1. We also run the algorithm to find the optimized confidence rule (FOCR), which is got by extending the algorithm in Ref.[2] in a naive way used in NFOIR. The found rules are compared in Table 1. For this experiment, the ranges of A_1 and A_2 are both split into 100 equal-width intervals. All the rules got by FOCR have the maximum confidence 1. However, they can not (exactly) reveal the pre-specified rules. For example, Rules 8, 10 and 12 reach the maximum confidence by enlarging the consequence of Rule 1. Rule 14 even enlarges the consequence to the whole range of A_2. Compared with those rules got by FOCR, Rules 9 and 11 got by HFOIR almost find Rule 2. When the value of η approaches the support value of Rule 1, Rule 13 almost reveals the antecedence of Rule 1. When the value of η is larger than the support value of Rule 1, Rule 15 is got by enlarging a little Rule 1.

Table 1 Experimental results with *DS*1

Rule #	Algorithm	Antecedence (Range on A_1)	Consequence (Range on A_2)	Confidence	Interest	η
8	FOCR	(0.4 0.42)	[0.0 0.4]	1	1.87	0.02
9	HFOIR	(0.31 0.37)	(0.06 0.07]	0.41	9.3	0.02
10	FOCR	(0.4 0.46)	[0.0 0.4]	1	1.87	0.05
11	HFOIR	(0.3 0.4)	(0.06 0.08]	0.81	8.85	0.05
12	FOCR	(0.4 0.48)	[0.0 0.4]	1	1.87	0.08
13	HFOIR	(0.4 0.5)	(0.3 0.38]	0.8	5.73	0.08
14	FOCR	[0 0.13]	[0.0 1]	1	1	0.12
15	HFOIR	(0.4 0.62)	(0.3 0.45]	0.22	2.52	0.12

We do almost the same experiments with *DS*2. The results are show in Table 2. Again, those rules got by FOCR enlarge the consequences to reach the maximum confidence. In contrast, those rules got by HFOIR come from the prespecified rules having their supports close to the thresholds and having relative high interests. Rule 17 comes from Rule 6 because it has the maximum value of interest among all rules. Rule 19 and Rule 21 come from Rule 5 because it has the maximum interest among all prespecified rules having their supports larger than η. The corresponding OIR (Rule 23) comes from Rule 7 when the value of η is approaching its support (0.23).

Table 2 Experimental results with *DS*2

Rule #	Algorithm	Antecedence (Range on A_1)	Consequence (Range on A_2)	Confidence	Interest	η
16	FOCR	(0.11,0.17]	[0.0,0.95]	1	1.03	0.05
17	HFOIR	(0.11,0.23)	(0.14,0.19]	0.45	3.36	0.05
18	FOCR	(0.18,0.28)	[0.0,0.99]	1	1	0.1
19	HFOIR	(0.11,0.26)	(0.11,0.19]	0.68	3.15	0.1
20	FOCR	(0.0,0.16)	[0.0,1.0]	1	1	0.15
21	HFOIR	(0.1,0.29)	[0.1,0.2]	0.84	3.1	0.15
22	FOCR	[0.0,0.21]	[0.0,1.0]	1	1	0.2
23	HFOIR	[0.0,0.3]	(0.1,0.22]	0.68	2.33	0.2

4.2 Experiment with real-life data set

In this subsection, we run HFOIR and FOCR on IPUMS[9] to show the feasibility of HFOIR in the real world. As discussed in Section 4, HFOIR is a heuristic algorithm and aims at improving the performance. As a tradeoff, it gets the approximate results. Therefore, the scalability and accuracy of HFOIR are evaluated with this data set. This data set is available on http://www.ipums.umn.edu/. In the following experiments, we only select three numeric attributes, *age*, *edurec* (educational attainment recode) and *inctot* (total personal income) from IPUMS. We remove the tuples with *age*≤15 and *inctot*≤100 from the data set to find some more interesting rules. The ranges of *age*, *edurec* and *inctot* are split into 76, 9 and 1 000 equal-width intervals, respectively. The threshold in Definition 3, δ, is set to be 0.04, experientially. We run HFOIR on two pairs of attributes, {*age*,*inctot*} and {*edurec*,*inctot*}, in turn. OIRs are compared with the optimized confidence rules(OCRs) in Tables 3 and 4.

Compared with OCRs with various η, the corresponding OIRs have smaller ranges on the consequences. They are more useful to reveal some matters unknown to us although their confidences are smaller than OCRs'. For *inctot*, OCRs have their lower bounds at 100 and upper bounds at least 100 235, under which most of the people (97.8%) have their incomes. Therefore, such rules are trivial for us. In contrast, OIRs have their upper bounds of at most 11 226 as well as lower bounds of 100. We can find some interesting information from those rules. For example, Rules 25, 27 and 29 tell us that those people at the age of between 15 and 23 are more likely to have incomes below 11 236 than those at other ages. Similarly, Rules 31 and 33 tell us that those at the age of beyond 24 have incomes larger than those at the age of below 24. In short, income increases with the increment of age. There is a sharp increment at age 24.

Table 3 Experimental results with *age* and *inctot*

Rule #	Algorithm	Antecedence (Range on *age*)	Consequence (Range on *inctot*)	Confidence	Interest	η
24	FOCR	[23,23]	[100,100235]	1	1.	0.02
25	HFOIR	[15,18]	[100,3552]	0.65	5.9	0.02
26	FOCR	[15,19]	[100,195767]	1	1	0.05
27	HFOIR	[15,21]	[100,5471]	0.55	3.2	0.05
28	FOCR	[28,31]	[100,207661]	1	1	0.1
29	HFOIR	[15,23]	[100,11226]	0.74	1.94	0.1
30	FOCR	[27,34]	[100,241039]	1	1	0.2
31	HFOIR	[32,62]	(26956,125941]	0.41	1.42	0.2
32	FOCR	[15,38]	[100,342326]	1	1	0.5
33	HFOIR	[24,64]	(10842,91027]	0.7	1.15	0.5

Table 4 Experimental results with *edurec* and *inctot*

Rule #	Algorithm	Antecedence (Range on *age*)	Consequence (Range on *inctot*)	Confidence	Interest	η
34	FOCR	[2,2]	[100 89493]	1	1.	0.02
35	HFOIR	[9,9]	(85656 361893]	0.1	3.24	0.02
36	FOCR	[1,2]	[100 197685]	1	1	0.05
37	HFOIR	[9,9]	(52661 332734]	0.24	2.83	0.05
38	FOCR	[1,3]	[100 319306]	1	1	0.1
39	HFOIR	[9,9]	(33862 361893]	0.48	2.27	0.1
40	FOCR	[1,6]	[100 333885]	1	1	0.2
41	HFOIR	[8,9]	(29641 383762]	0.41	1.54	0.2
42	FOCR	[1,7]	[100 378007]	1	1	0.5
43	HFOIR	[1,8]	[100 18515]	0.63	1.14	0.5

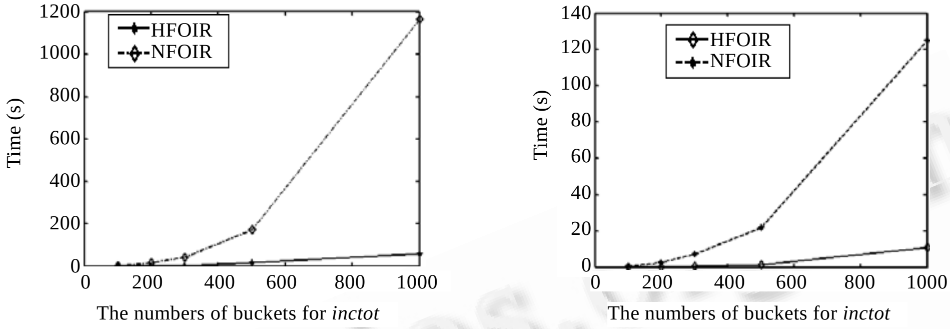
In IPUMS, the values of *edurec* are encoded with positive integers. The larger integer denotes higher education level. Zero denotes missing value. The codes of *educrec* and their meanings are shown in Table 5. OIRs and OCRs for {*edurec*,*inctot*} are listed in Table 4. Again, those OIRs are more useful than OCRs in showing the relationship between *edurec* and *inctot*. In especial, Rules 35, 37 and 39 recover the strong interreaction of high education level and high income. Compared with the above rules, Rule 43 shows the relationship between income and relative low education level. In sum, those having higher education levels are more likely to get higher incomes. There is a sharp increment of income for those having more than 4 years of college experience.

Table 5 Codes of *educrec* and their meanings

Code	Meaning	Code	Meaning
0	*N/A*	5	Grade 10
1	None or preschool	6	Grade 11
2	Grade 1, 2, 3, or 4	7	Grade 12
3	Grade 5, 6, 7, or 8	8	1 to 3 years of college
4	Grade 9	9	4+ years of college

4.3 Scalability and accuracy

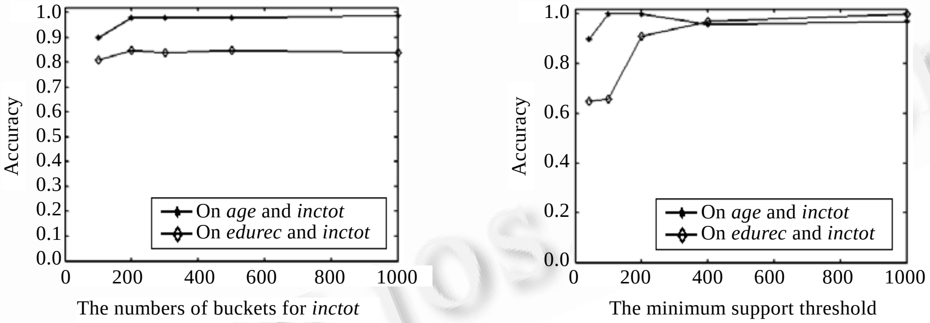
The graphs in Fig.2 plot the execution time of HFOIR and NFOIR on two pairs of attributes. The numbers of buckets on *age* and *edurec* are 76 and 9, and they are fixed during the experiments. The time collapsed for various numbers of buckets on *inctot* is the average of the time for the five various support thresholds appeared in Table 3 or Table 4. It can be noticed that the execution time of HFOIR is significantly less than that of NFOIR. HFOIR shows an approximately linear scalability with the number of buckets.



(a) With *age* and *inctot* (b) With *edurec* and *inctot*

Fig.2 The scalability of HFOIR and NFOIR

Figure 3 depicts the sensitivity of accuracy of HFOIR to the number of buckets on *inctot* and to the minimum support threshold. The vertical axis represents the accuracy, which is the ratio of the interest value got by HFOIR to that got by NFOIR. Actually, the vertical coordinates of points in Fig.3(a) are the averages of accuracies for the five various thresholds. We get most of the accuracies beyond 80% with *edurec* and *inctot*, and beyond 90% with *age* and *inctot*, respectively. Similarly, those in Fig.3(b) are the averages of accuracies for the various numbers of buckets. We notice that the accuracies are satisfying except for low supports (0.02 and 0.05).



(a) To the number of buckets (b) To the minimum support threshold

Fig.3 The sensitivity of accuracy of HFOIR

5 Conclusion

In this paper, we generalize the optimized association rule problem by allowing two uninstantiated numeric attributes in antecedence and consequence. Besides, we replace *confidence* with *interest* in order to find more interesting optimized rules. We first simplify the problem into that of finding the optimized *interest* rules with one numeric attribute. Then an efficient linear algorithm is adapted for it. Finally, we tackle the original problem by a heuristic extension of the efficient algorithm. The experiment results with two synthetic data sets show the advantages of *interest* over *confidence* on finding the interesting rules. The experiment results with a real-life data set demonstrate the approximately linear scalability of HFOIR with the number of buckets. Based on the experiments, we also draw the conclusion that the accuracy of HFOIR is satisfying in most cases.

In our research, the instantiated regions of all optimized rules are rectangular. However, in practice, the admissible regions often bring more information to users[3]. In future, we will further work on how to find OIRs and OSRs with admissible regions.

References:

[1] Agarwal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. In: Peter B, Sushil J, eds. Proc. of the ’93 ACM SIGMOD Int’l Conf. on Management of Data. Washington: ACM Press, 1993. 207−216.

[2] Fukuda T, Morimoto Y, Morishita S, Tokuyama T. Mining optimized association rules for numeric attributes. In: Egenhofer Max J, ed. Proc. of the 15th ACM SIGACTSIGMOD-SIGART Symp. on Principles of Database Systems. Montreal: ACM Press, 1996. 182−191.

[3] Fukuda T, Morimoto Y, Morishita S, Tokuyama T. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. In: Jagadish HV, Mumick IS, eds. Proc. of the ACMSIGMOD Conf. On Management of Data. Montreal: ACM Press, 1996. 13−23.

[4] Rastogi R, Shim K. Mining optimized support rules for numeric attributes. In: Dayal U, Ramamritham K, Vijayaraman TM, eds. Proc. of the 15th Int’l Conf. on Data Engineering. Sydney: IEEE Computer Society Press, 1999. 126−135.

[5] Rastogi R, Shim K. Mining optimized association rules with categorical and numeric attributes. IEEE Trans. on Knowledge and Data Engineering, 2002,14(1):29−50.

[6] Rastogi SBR, Shim K. Mining optimized gain rules for numeric attributes. IEEE Trans. on Knowledge and Data Engineering, 2003, 15(2):324−338.

[7] Elble J, Heeren C, Pitt L. Optimized disjunctive association rules via sampling. In: Wu XD, Alex T, eds. Proc. of the 3rd IEEE Int’l Conf. on Data Mining. Melbourne: IEEE Computer Society Press, 2003. 43−50.

[8] Brin S, Motwani R, Silverstein C. Beyond market baskets: Generalizing association rules to correlations. In: Joan P, ed. Proc. of the ACM SIGMOD Int’l Conf. on Management of Data. Tucson: ACM Press, 1997. 265−276.

[9] Ruggles S, Sobek M, Alexander T, Fitch CA, Goeken R, Hall PK, King M, Ronnander C. Integrated public use microdata series: Version 3.0 [Machine-readable database]. Minneapolis: Minnesota Population Center [producer and distributor], 2004.

HE Zhi was born in 1976. He is a Ph.D. candidate at the School of Computer and Information Technology, Beijing Jiaotong University and a CCF student member. His current research areas are data mining and evolution algorithm.

HUANG Hou-Kuan was born in 1940. He is a professor and doctoral supervisor at the School of Computer and Information Technology, Beijing Jiaotong University and a CCF senior member. His research areas are artificial intelligence and KDD.

TIAN Sheng-Feng was born in 1944. He is a professor and doctoral supervisor at the School of Computer and Information Technology, Beijing Jiaotong University. His research areas are pattern recognition and artificial intelligence.