SARM — Succinct Association Rule Mining: An Approach to Enhance Association Mining*

Jitender Deogun and Liying Jiang

Department of Computer Science and Engineering,
University of Nebraska - Lincoln, Lincoln, NE, 68588-0115, USA
{deogun, ljiang}@cse.unl.edu

Abstract. The performance of association rule mining in terms of computation time and number of redundant rules generated deteriorates as the size of database increases and/or support threshold used is smaller. In this paper, we present a new approach called *SARM — succinct association rule mining*, to enhance the association mining. Our approach is based on our understanding of the mining process that items become less useful as mining proceeds, and that such items can be eliminated to accelerate the mining and to reduce the number of redundant rules generated. We propose a new paradigm that an item becomes less useful when the most interesting rules involving the item have been discovered and deleting it from the mining process will not result in any significant loss of information. SARM generates a compact set of rules called *succinct association rule* (SAR) set that is largely free of redundant rules. SARM is efficient in association mining, especially when support threshold used is small. Experiments are conducted on both synthetic and real-life databases. SARM approach is especially suitable for applications where rules with small support may be of significant interest. We show that for such applications SAR set can be mined efficiently.

1 Introduction

Association rule mining is one of the most important data mining techniques. Association rules are of the form $X \Rightarrow Y$, where X is the *antecedent*, and Y the *consequent* of the rule. *Support* of $X \Rightarrow Y$ indicates the percentage of transactions in dataset that contain both X and Y. *Confidence* of $X \Rightarrow Y$ denotes the probability of a transaction containing Y given that it contains X. Association rule mining is to find rules that have support and confidence greater than user-specified *minimum support* (s_{min}) and *confidence* (c_{min}) threshold values.

Apriori algorithm [4] is the well known standard method for association mining, and most of the later algorithms follow the framework of Apriori. Real world studies, however, show that association rule mining is still faced with the

* This research was supported in part by NSF Grant No. EIA-0091530, USDA RMA Grant NO. 02IE08310228, and NSF EPSCOR, Grant No. EPS-0346476.

M.-S. Hacid et al. (Eds.): ISMIS 2005, LNAI 3488, pp. 121–130, 2005.

problems of time-inefficiency, especially for applications on large databases when minimum support threshold used is small [8]. While in many cases, small s_{min} is desirable, and it is important to improve the mining efficiency when s_{min} used is small. Another problem in association mining is that often too many rules are generated, many of which are redundant [6]. In this paper, to solve the problem of time inefficiency and rule redundancy in association mining, we propose approach called *SARM— succinct association rule mining*. The SARM approach generates a set of *succinct association rules* (SAR) that contains most of the interesting and useful rules and can be mined efficiently.

A key factor that causes the inefficiency and redundancy in association rule mining is the large amount of items in a database. It maybe noted that the mining complexity is exponentially proportional to the dimension of the database [7]. We claim that some items might lose their usefulness from the point of view of information as the mining proceeds. That is, at a certain point in the mining process, if there are interesting rules involving the item that have not been discovered yet, then the item is useful and important to the mining process at this point. Therefore, an item loses its usefulness or importance as the mining progresses and the interesting rules involving it are discovered. We define the SARM paradigm as follows: *if the most interesting rules involving an item have been discovered, then such item becomes less useful, and deleting it from the mining accelerates the mining and reduces the number of redundant rules generated as well*. Finding the point at which most of the interesting rules involving the item have been discovered during the mining process is based the model of *maximal potentially useful* (MaxPUF) association rules [2], which are a set of rules that are most informational and interesting. If the MaxPUF rules of an item have been mined, we show that such an item becomes less useful and deleting it from the mining process will not result in any significant loss of information.

2 Related Work

The SARM approach is based on the *maximal potentially useful* (MaxPUF) pattern model. We give a brief review for background knowledge of our work in [2, 3]. To facilitate interesting pattern discovery, we develop a logic model of data mining that integrates *formal concept analysis* (FCA) [5] and *probability logic* [1] in [3]. Probability logic extends first-order logic with probability expression capability. It defines a language that includes predicate symbols, object variables, statistical probability operation "[ ]". Predicate symbols represent conjunctions of attributes, and Object variables define the domain of a set of objects of interest. Operation "[ ]" computes the probability of a proposition represented by predicate symbols over a set of objects.

Definition 1. *Concept. If P is a predicate symbol and x an object variable, P_x is a concept, or simply denoted as P if the domain is clear from the context. If P_1,...,P_j are concepts of the same domain, $P = P_1 ... \wedge P_j$ is a concept.*

If $P = \bigwedge_{i=1}^{n} P_i$, we say P is a *superconcept* of P_i and P_i is a *subconcept* of P, $1 \leq i \leq n$. The set of all concepts for given context form a lattice under superconcept/subconcept relation [2, 5]. Moreover, we define all superconcepts and subconcepts of a concept Q as the *relative concepts* of Q.

Definition 2. Elementary and Conditional Pattern. *If P is a concept, then $[P] = r$ $(0 \leq r \leq 1)$ is an elementary pattern. If P, Q are concepts and $P \neq \emptyset$, then $[Q|P] = r$ is a conditional pattern. Q is called conSequent Concept (SC), and P conDition Concept (DC) of the pattern. Probability r is called confidence of the pattern, and $r = \frac{[QP]}{[P]}$. A pattern is an elementary or a conditional pattern.*

We can logically formulate an association rule as a conditional pattern. *DC* of the conditional pattern represents the *antecedent* of the rule and *SC* represents the *consequent* of the rule. *Probability* of the pattern denotes the *confidence* of the rule. *Support* of the antecedent, the consequent and the rule is equal to the *probability* of the corresponding elementary patterns. For example, if we have an rule $A \Rightarrow B$ with confidence c and support s, then the corresponding conditional pattern is $[B|A] = c$, and $sup(A) = [A]$, $sup(B) = [B]$, $s = sup(AB) = [AB]$.

Definition 3. MaxPUF patterns and Valid DC. *Let c_{min} be the user-defined minimum confidence threshold, a pattern $[B|A] = r$, if $r \geq c_{min}$, and there is no patterns in the form of $[B|A'] = r'$ where $r' \geq c_{min}$ and $A' \subset A$, then $[B|A] = r$ is a MaxPUF pattern of consequent concept B, and A is called a* valid DC *of B.*

The interestingness of a pattern is tightly related to the confidence of the pattern. No doubt, high confidence patterns are interesting. However, MaxPUF patterns are the most informational and potentially useful patterns among all the high confidence patterns. Among all high-confidence patterns of a certain *SC*, MaxPUF patterns are the patterns *DC* of which has smallest number of items. The *DC*s of a MaxPUF pattern is the point of articulation in the concept lattice [2]. Below this point, no high-confidence pattern can be constructed with the *relative concepts* of the *Valid DC*. A *Valid DC* is the most informational condition concept, because it is the minimal condition concept such that the *SC* occurs at high enough frequency. A further narrower condition concept (*DC*), is not as interesting as *Valid DC*, because the additional items are very likely minor condition factors or trivial factors. In this sense, *Valid DC* can be seen as the set of main conditional factors to assure the occurrence of the consequent concept. As a *Valid DC* is the most informational concept among all of its relative concepts, we state that a MaxPUF pattern is most informational pattern among all the patterns of a consequent concept (*SC*).

3 SARM — Succinct Association Rule Mining

In association mining, an attempt to obtain more complete information results in much higher cost in terms of computation time and in addition a larger number

of redundant rules generated. To strike a balance between these two extremes, we do not mine the entire set of useful rules, rather, we want to efficiently discover a subset of rules that contains most of the informational rules while only as small number of redundant rules as possible are generated.

As the mining process is to first generate candidate frequent itemsets and then validate the candidates, to accelerate the procedure, our thought process is to reduce the number of candidates that are not useful. The number of candidates is largely determined by the number of total items in the dataset. For example, if one more items is added to the dataset, at each pass, the number of candidates is at least doubled. In this sense, if we can efficiently reduce the number of items, we can greatly reduce the number of candidates generated and the related computation of support. Therefore, to accelerate the rule mining, we consider how to effectively reduce the number of unimportant items in the mining process. The importance of an item is related to the notion of redundant rules.

Assertion. Rules of the following types are redundant: *1) rules with low confidence; 2) rules that is not MaxPUF rule and has consequent identical to a MaxPUF rule; 3) the rules information of which can be deducted or implied from the rules already generated.*

At the beginning of association mining, all the items are important. However, as mining progresses, some items become less important or less informational, because useful rules involving these items have already been discovered. Thus we could delete such items from the mining process and not consider them in further computations. Now the problem is which are these items?

Assume after generating frequent k-itemsets in the mining, instead of continuing to generate $(k+1)$-itemsets, we first generate high-confidence patterns using k-itemsets. Suppose that we discover a pattern $[B|A] = r$, where $r \geq c_e > c_{min}$. Here c_e is a user-defined parameter that we call *elimination confidence threshold*. c_e is a relatively high value, we usually define c_e higher than 0.7. What will happen if we delete itemset B from the dataset? There are six types of patterns that will be affected by the deletion of itemset B, which are 1)$[B|AX] = r$, 2)$[BY|AX] = r$, 3) $[Y|BA] = r$, 4) $[B|Y] = r$, 5) $[Y|B] = r$, 6) $[BY|A] = r$, where A, X and Y represent arbitrary itemsets.

Discussion on Missing Patterns. First of all, if these patterns have confidence lower than c_{min}, then they are redundant and thus these patterns should not be mined. So in the following discussions, we assume that these patterns, if mined, will have confidence higher than c_{min}.

Case 1. For patterns of the form $[B|AX]$, we argue they are not as useful as pattern $[B|A]$ because $[B|A]$ is the MaxPUF pattern of consequent concept B. Based on the Assertion in Section 3.1, pattern $[B|AX]$ is redundant compared to the MaxPUF pattern $[B|A]$. It is desirable that such patterns are not mined.

Case 2. After B is deleted, no patterns of the form $[Y|BA]$ will be discovered. We argue that if we could discover the pattern $[Y|A]$, then $[Y|A]$ is the MaxPUF pattern of $[Y|BA]$ and thus $[Y|BA]$ is redundant in presence of $[Y|A]$.

Lemma 1. Assume a pattern $[B|A] \geq c_e$ and $[Y|BA] \geq c_{min}$, then $[Y|A] \geq c_{min} * c_e$, and $[[Y|A] \geq c_{min}] \geq \frac{1-c_{min}}{1-c_{min} \cdot c_e}$, that is, the probability that $[Y|A] \geq c_{min}$ is greater than $\frac{1-c_{min}}{1-c_{min} \cdot c_e}$.

Proof: *Since* $[B|A] = \frac{supp(AB)}{supp(A)} = c \Rightarrow supp(AB) = supp(A) \cdot r$, *and*$[Y|AB] = \frac{supp(YAB)}{supp(AB)} = \frac{supp(YAB)}{supp(A) \cdot r} \geq c_{min} \Rightarrow \frac{supp(YAB)}{supp(A)} \geq r \cdot c_{min}$, *then* $[Y|A] = \frac{supp(YA)}{supp(A)} \geq \frac{supp(YAB)}{supp(A)} \geq r \cdot c_{min} \geq c_e \cdot c_{min}$. As $[c_{min} \cdot c_e \geq c_{min}] \geq \frac{1-c_{min}}{1-c_{min} \cdot c_e}$, therefore, $[[Y|A] \geq c_{min}] \geq \frac{1-c_{min}}{1-c_{min} \cdot c_e}$. ∎

Case 3. $[BY|AX]$ implies two pieces of information: $[B|AX]$ and $[Y|AX]$. The mining of $[Y|AX]$ is not affected by deletion of B. $[B|AX]$ is less useful compared to its MaxPUF pattern $[B|A]$. Because we use high c_e value, $[B|A]$ suggest that A may always imply B, then AX also imply B. Therefore, $[B|AX]$ and $[Y|AX]$ together imply the information of $[BY|AX]$ and make $[BY|AX]$ redundant.

Case 4. The discovery of $[B|Y]$ can be formulated as discovering the MaxPUF patterns, SC of which is B. If B is an interesting consequent concept (SC), we can start a special process to find all of its MaxPUF patterns.

Case 5. For patterns $[Y|B]$, there are two cases. First if Y is a 1-itemset, the mining $[Y|B]$ is not affected by deletion of B. On the other hand, if $Y = \{y_1, y_2, ..., y_k\}$ $(k > 1)$ and $[Y|B] \geq c_{min}$, it is easy to see that $[y_i|B] \geq c_{min}$ $(1 \leq i \leq k)$. As y_i is 1-itemset, mining of $[y_i|B]$ is not affected by the deletion of B. Then we only need a complementary process to discover patterns with k-itemset SC $(k > 1)$ from the patterns that have 1-itemset SC and an identical DC, which is B in this case. We introduce this process in Section 4.

Case 6. Similar strategy in Case 5 is used for patterns of the form $[BY|A]$. That is, can we discover this pattern based on $[B|A]$ and $[Y|A]$, which are patterns with 1-itemset SC and an identical DC, which is A in this case.

Succinct Association Rule Set. From the above discussions, we can see if we delete the SCs of high-confidence patterns during the mining process, we are still able to discover most of the useful patterns/rules. As it is not necessary to discover all the rules, which results in many redundant ones, we propose SARM, a novel approach for improving the efficiency of association mining and at the same time discovering most of the useful rules. In general, SARM approach is a mining process involves dynamic elimination of items during the mining process. That is, after discovering the frequent k-itemsets, if we can construct a pattern $[B|A] \geq c_e$ from a k-itemsets, we prohibit the SC of the pattern, which is B in this case, from generating candidate $(k+1)$-itemsets. SARM generates a set of association rules we call *succinct association rule* (SAR) set. As association rules can be represented as patterns, the formal definition of SAR set is given as follows using the format of patterns.

Definition 4. Succinct Pattern Set. *Let* $I = \{i_1, i_2, ..., i_m\}$ *represent the set of items in a database,* c_{min} *and* c_e *respectively be minimum and elimination confi-*

dence thresholds. Succinct pattern set is a set of patterns of the form $[X|Y] = r$ *where* $0 \leq < c_{min} \leq r \leq 1$*, and meet the condition that in the set, if there is a rule of the form* $[B|A] \geq c_e$*, then there is no rule of the form* $[B_1|A_1] = r_1$ *such that* $B \subseteq B_1 \cup A_1$ *and* $|A| + |B| > |A_1| + |B_1|$.

Lemma 2. *1-itemset deletion property.* In SARM, if pattern $[B|A] = r$ is generated from a k-itemset and $r \geq c_e$, then B is a 1-itemset and A is a $(k-1)$-itemset.

This Lemma can be proven by contradiction. Assume k-itemsets are generate in the k^{th} pass. If there is a pattern $[B|A] \geq c_e$, where $B = \{b_1, ..., b_m\}$ $(k > m > 1)$, then in the $(k - m + 1)^{th}$ pass, we must have discovered the pattern $[b_1|A] \geq c_e$, which results in the deletion of item b_1. And similar case is for other b_i. As b_i has been deleted before k^th pass, the pattern $[B|A]$ will not exist. It is a contradiction.

Some patterns of the form $[BY|A]$ and $[Y|B]$ are not included in the SAR set, to prevent information loss, SARM includes a special complementary process to discover such patterns whenever deemed necessary. SARM approach explores only the most informational patterns. In SARM, after an item has been explored with adequate information, it is eliminated and does not take part in any future mining process. The deletion of some items will greatly reduce the number of candidate patterns, and at the same time, it is safe from loss of information. SARM is a good model for the objective to accelerate association rule mining and have an overview of the most useful patterns. The SAR set may not contain some special patterns, but it retains most of the informational association rules and is largely free of redundant rules.

4 The SARM Algorithm

The SARM approach has two parts, the main part, mining the SAR set, and the other is a complementary process to mine patterns not included in SAR set.

Mining the SAR Set. SARM algorithm combines itemset discovery and rule mining, and use rule mining results to help eliminate less important item, which is shown in Figure 1. SARM algorithm commits several passes to discover frequent itemsets from 1-itemsets to l-itemsets. Each pass includes three main steps. The first step is to generate candidate k-itemsets (CS_k) and check the support of them to find the set of frequent k-itemsets (FS_k). This is similar to Apriori. But different from Apriori, before continue to generate candidate $(k+1)$-itemsets, in the second step, we use the discovered frequent k-itemsets to build up association rules. If a rule has confidence higher than c_e, we delete the consequent item, i.e., we prune the FS_k to generate FSC_k, so that none of itemsets in FSC_k includes the eliminated items. In general, FSC_k is the set of FS_k except those itemsets that include the items in EC. In the third step, use FSC_k to generate candidate frequent $(k + 1)$-itemsets (CS_{k+1}). The process is repeated until no candidate itemsets can be generated.

Algorithm: $SARM$ ($\mathcal{D}$, I, s_{min}, c_{min}, c_e)
Input: 1)Database $\mathcal{D}$, 2)s_{min}, 3)c_{min} and 4)elimination confidence (c_e).
Output: SAR set satisfying s_{min}, c_{min}, and c_e.
1)Discover all frequent 1-itemsets, store into FS_1;
2)$FSC_1 = FS_1$; $k = 2$;
3)$CS_2 = \{c \mid c = f1 \cap f2, |c| = 2, \forall f1, f2 \in FSC_1\}$;
4)FS_2=Gen-FS(CS_2);
5)EC=Gen-rule(FS_2);
6)$FSC_2 = FS_2 - \{f \mid f \in FS_2, f \cap EC \neq \emptyset\}$;
7)while ($FSC_k \neq \emptyset$)
8) $k++$;
9) $CS_k = \{c \mid c = f1 \cap f2, |c| = k, \forall f1, f2 \in FSC_{k-1}\}$;
10) FS_k=Gen-FS(CS_k);
11) EC=Gen-rule(FS_k);
12) $FSC_k = FS_k - \{f \mid f \in FS_k, f \cap EC \neq \emptyset\}$;
13)end while

Fig. 1. SARM: Succinct Association Rules Mining Algorithm

The Complementary Process. The complementary process is to generate patterns with k-itemset SC ($k > 1$) from the patterns with 1-itemset SC and an identical DC.

Lemma 3. It is possible that pattern $[X_1X_2...X_i|A] \geq c_{min}$ only if for $\forall X_j$, $[X_j|A] \geq c_{min}, 1 \leq j \leq i$.

Based on Lemma 4, to the discover the patterns SC of which is a k-itemset ($k \geq 2$), it depends on the patterns SC of which is 1-itemset. The idea is, check the discovered patterns, only if two or more 1-itemset SCs have a common DC, we can construct a candidate pattern, DC of which is the common DC and SC is a combination of the 1-itemset SCs. For example, assume there are two discovered patterns [1|3] and [2|3], where the common DC=(3), then [1, 2|3] should be a candidate pattern because it is possible $[1, 2|3] \geq c_{min}$.

5 Experiments and Analysis

The experiments are designed to test the efficiency of the proposed algorithm for mining SAR set, compare the SAR set with the set of rules discovered by Apriori-like algorithms and evaluate the set of SAR. Comprehensive performance studies are conducted on both synthetic and real-life datasets. The programs are coded in C++, and experiments are conducted on a 950 MHz Intel Pentium-3 PC with 512 MB main memory.

Figure 2 and 3 show the experimental results on the synthetic database. For T40 databases, SARM is 5 to 20 times faster. The improvement increases almost exponentially as s_{min} decreases. This demonstrates that SARM model is efficient and suitable for applications requiring small s_{min}. The decrease of running time is due to reduction in the number of candidates generated in SARM

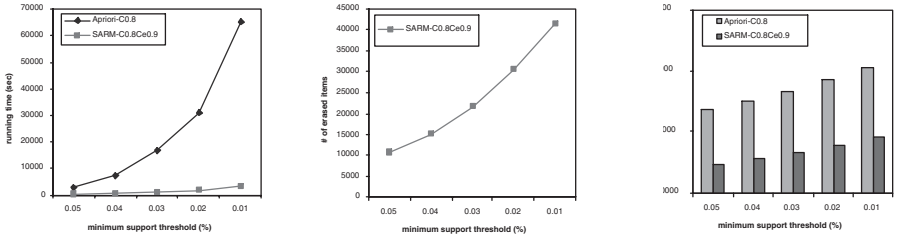

Fig. 2. Varying Support for Database T40I10D200K

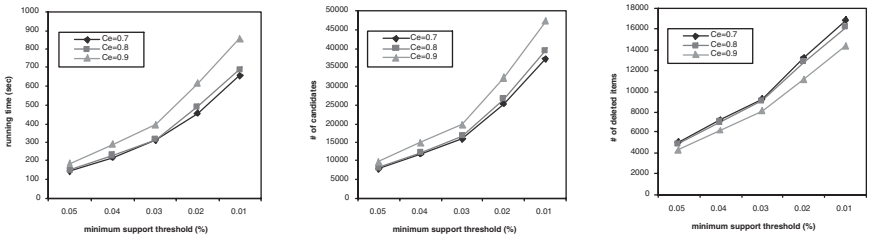

Fig. 3. Performance Study for Different c_e Value on Database T25I6D100K

and correspondingly less supporting computations are needed as some items are deleted during the mining process. Following Lemma 1, it is noted that a huge enough value for c_e must be chosen so that patterns of the form $[Y|A] \geq c_{min}$ can be effectively discovered. In the experiments, we choose $c_e = 0.9, 0.8, 0.7$, and $c_{min} = 0.6$, then based on Lemma 1, it is estimated that $[[Y|A] \geq c_{min}]$ is respectively greater than $82\%, 77\%, 69\%$. From Figure 3, we see that as c_e decreases, the execution time decreases, so do the number of candidates and rules generated (see Figure 3). These are natural results since smaller c_e will result in more items satisfying the elimination threshold earlier and thus being deleted earlier, and then smaller number of candidates and rules are generated. The SAR set is much smaller than general association rule set, for T25 data, the number of rules in SAR set is 25 times smaller than that of Apriori rules on average, and 70 to 190 times smaller for T40 data.

Experiments on Real-life Databases. We further evaluate the SAR set that is generated from the real-life databases. The data is collected from the weather station at Clay Center, Nebraska. We intend to discover rules that demonstrate the relations between weather events and environmental indices. We use quality metrics *Precision* and *Recall* to evaluate the rule set. Precision is defined as the percentage of interesting rules discovered among all the rules discovered by an algorithm, and recall is defined as the percentage of rules discovered by an algorithm to the number of rules that exist in the given datasets. Based on the definitions, the *recall* of the conventional rule set discovered by Apriori algorithm can be deemed as 100% because we can assume that all the rules that exist in the database are the rules discovered by Apriori. Table 1 shows the comparison

Table 1. SAR Set Evaluation

$c_{min} = 0.8$	# Rules	Recall(%)	Precision(%)
Apriori	2423	100	4.95
SAR($c_e = 0.95$)	595	91.67	20.79
SAR($c_e = 0.9$)	303	76.67	30.36
SAR($c_e = 0.8$)	157	66.67	50.96

of general rule set and SAR set with respect to the recall and precision. The precision of the SAR set is five to eleven times higher than general rule set generated by Apriori with different c_e value used. This demonstrates that SAR set is largely free of redundant rules. The recall of the SAR set is reasonably high, which shows that SAR set can discover most of the useful association rules. We also notice that the length of the longest rules in SAR set is generally half of that in general rule set. That is, SARM tends to discover short useful patterns. This fact complies with the principle of MaxPUF association rules that we believe if shorter condition can predict a consequent, then the rules with longer condition to predict the same consequent are redundant.

6 Conclusions

In this paper, we investigate the issues affecting the efficiency of association mining in terms of computation time and the number of redundant rules generated, especially when a smaller s_{min} is used in context of large databases. We show that items become less informational and thus less important as the mining proceeds. Thus, dynamically deleting such items during the mining process not only improves the mining efficiency but also effectively prevents the generation of a large number of trivial or redundant rules. Based on our hypothesis, we propose a new model called SARM, which generates a compact rule set called SAR set. We develop an algorithm to discover the SAR set, which combines the discovery of frequent itemsets and generation of rules, in a way to accelerate the mining process. The experimental results on synthetic databases show that SARM is 5 to 20 times faster than Apriori algorithm as s_{min} decreases. We evaluate the SAR set generated from real-life databases, and show that SAR set retains most of the useful rules and is largely free of redundant rules, the precision increases five to eleven times as compared to rules generated by Apriori, and at the same time, the recall value remains high. We believe that SARM is an efficient and useful model for generating most informational patterns in large databases.

References

1. F. Bacchus. *Representing and Reasoning With Probabilistic Knowledge*. MIT, 1990.
2. J. Deogun, L. Jiang, and V. Raghavan. Discovering maximal potentially useful association rules based on probability logics. In *Proc. of Rough Sets and Current Trends in Computing, 4th International Conf.*, 2004.

3. J. Deogun, L. Jiang, Y. Xie, and V. Raghavan. Probability logic modeling of knowledge discovery in databases. In *Proc. of Foundations of Intelligent Systems, 14th International Symposium (ISMIS)*, 2003.
4. R. Agrawal etc. Mining association rules between sets of items in large databases. In *Proc. of ACM SIGMOD Intern. Conf. on Management of Data*, 1993.
5. B. Ganter and R. Wille. *Formal Concept Analsis: Mathematical Foundations*. Springer-Verlag, Berlin, 1999.
6. M. Kryszkiewicz. Closed set based discovery of representative association rules. In *Advances in Intelligent Data Analysis, 4th International Conf.*, 2001.
7. S. Orlando and etc. A scalable multi-strategy algorithm for counting frequent sets. In *Proc. of the 5th Intern. Workshop on High Performance Data Mining*, 2002.
8. Zijian Zheng, Ron Kohavi, and Llew Mason. Real world performance of association rule algorithms. In *Proc. of the 7th ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining*, 2001.