算法四：非频繁模式与负关联规则挖掘的注释

自 Agrawal 等在 1993 年提出关联模式挖掘[7]，到 1998 年我开始数据挖掘研究时，这方面的研究成果相当多。读了一些主要论文后，感觉大家基本上是在做 Apriori 算法的改良。除了第一篇论文中想到的划分挖掘方法外，我没有设计出更有意义的频繁模式挖掘算法。经过对实际应用需求和数据的理解，我们发现：传统的关联规则能抓住频繁项集合中相伴性的相互关系，尽管广泛存在且切实有效，但它反映的只是潜在数据项的同现，有着明显的不足。

在现实世界中，完整的关联性应该是双重的，包括同时发生（关联规则）和排斥发生（负关联规则，它表示在某一事务中某些项集合的存在意味着另一些项集合的不存在）这两类关联。互斥性关联模式通常是隐藏在非频繁项集合中的，因此，必须提供一种全新角度的数据相关性分析方法。另一方面，注意到很多现有的数据挖掘算法，比如分类和聚类都可以使用互斥关联分析，所以，互斥性关联方面的成就肯定会对数据挖掘领域带来深远影响。针对以上发现，我们主张充分利用非频繁项集（非频繁模式），从非频繁项集合中挖掘潜在有用的模式：负关联规则，它刻画了项集合中互斥性这一相互关系。与频繁项集合相比，挖掘有用的非频繁项集合更具有挑战性，因为它必须面对一个完全的巨大搜索空间，用于频繁模式的剪枝技术在非频繁项集合挖掘中是无效的。因此，在这一篇论文中我们提出用于挖掘负关联规则的非频繁项集合表示与操作方法、测度理论、挖掘算法、剪枝技术、和评估方法。非频繁模式和频繁模式在实际应用中具有很强的互补性，它们一起形成了数据关联的完整体系，使得一些有意义的非频繁模式能在决策等应用中充分发挥作用。

文献[8]是这一篇论文的会议版本，仅供感兴趣的读者下载阅读。

参考文献

[7]. R. Agrawal, T. Imielinski, A. Swami. Mining association rules between sets of items in massive databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, ACM, Washington D.C., 1993: 207 - 216.

[8]. Xindong Wu, Chengqi Zhang and **Shichao Zhang**. Mining Both Positive and Negative Association Rules. In: Proceedings of 19th International Conference on Machine Learning (ICML), Sydney, Australia, 2002: 658-665.

Efficient Mining of Both Positive and Negative Association Rules

XINDONG WU
University of Vermont
CHENGQI ZHANG
University of Technology, Sydney, Australia
and
SHICHAO ZHANG
University of Technology, Sydney, Australia
and Tsinghua University, China

This paper presents an efficient method for mining both positive and negative association rules in databases. The method extends traditional associations to include association rules of forms $A \Rightarrow \neg B$, $\neg A \Rightarrow B$, and $\neg A \Rightarrow \neg B$, which indicate negative associations between itemsets. With a pruning strategy and an interestingness measure, our method scales to large databases. The method has been evaluated using both synthetic and real-world databases, and our experimental results demonstrate its effectiveness and efficiency.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning—*Knowledge acquisition*; H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Association rules, negative associations

1. INTRODUCTION

By definition [Agrawal et al. 1993b; Chen et al. 1996], an association rule is an implication of the form $A \Rightarrow B$, where A and B are frequent itemsets in a transaction database and $A \cap B = \emptyset$. In practical applications, the rule $A \Rightarrow B$

A preliminary version of this article has been published in the *Proceedings of the 19th International Conference on Machine Learning*, 2002, 658–665. This research is partially supported by the Australian Research Council (under grants DP0343109 and DP0449535) and the Guangxi Natural Science Funds.

Authors' addresses: Xindong Wu, Department of Computer Science, University of Vermont, 351 Votey Building, Burlington, Vermont 05405; email: xwu@cs.uvm.edu; Chengqi Zhang and Shichao Zhang, Faculty of Information Technology, University of Technology, Sydney, P.O. Box 123, Broadway NSW 2007, Australia; email: {chengqi,zhangsc}@it.uts.edu.au.

can be used to predict that 'if A occurs in a transaction, then B will likely also occur in the same transaction', and we can apply this association rule to place 'B close to A' in the store layout and product placement of supermarket management. Such applications are expected to increase product sales and provide more convenience for supermarket customers. Therefore, mining association rules in databases has received much attention recently [Aggarawal and Yu 1998; Agrawal et al. 1993a, 1993b; Bayardo 1998; Brin et al. 1997; Chen et al. 1996; Han et al. 2000; Park et al. 1997; Shintani and Kitsuregawa 1998; Srikant and Agrawal 1996, 1997; Tsur et al. 1998].

With the increasing use and development of data mining techniques and tools, much work has recently focused on finding alternative patterns, including unexpected patterns [Padmanabhan and Tuzhilin 1998, 2000], exceptional patterns [Hussain et al. 2000; Hwang et al. 1999; Liu et al. 1999; Suzuki 1997; Suzuki and Shimura 1996], and strong negative associations [Brin et al. 1997; Savasere et al. 1998].

Unexpected patterns and exceptional patterns are referred to as *exceptions of rules*, also known as *surprising patterns*. An exception is defined as a deviational pattern to a well-known fact, and exhibits unexpectedness. For example, while '$bird(x) \Rightarrow flies(x)$' is a well-known fact, an exceptional rule is '$bird(x), penguin(x) \Rightarrow \neg flies(x)$'. This exception indicates that unexpected patterns and exceptional patterns can involve negative terms and therefore can be treated as a special case of negative rules.

A strong negative association is referred to as a *negative relation* between two itemsets. This negative relation implies a negative rule between the two itemsets. However, strong negative associations reveal only the existence of negative rules in a hidden representation, and do not provide the actual negative rules.

Unlike existing mining techniques, the research in this paper extends traditional associations to include association rules of forms $A \Rightarrow \neg B$, $\neg A \Rightarrow B$, and $\neg A \Rightarrow \neg B$, which indicate negative associations between itemsets. We call rules of the form $A \Rightarrow B$ *positive rules*, and rules of the other forms *negative rules*.

While positive association rules are useful in decision-making, negative association rules also play important roles in decision-making. For example, there are typically two types of trading behaviors (insider trading and market manipulation) that impair fair and efficient trading in securities stock markets. The objective of the market surveillance team is to ensure a fair and efficient trading environment for all participants through an alert system. Negative association rules assist in determining which alerts can be ignored. Assume that each piece of evidence: A, B, C and D, can cause an alert of unfair trading X. If having rules $A \Rightarrow \neg X$ and $C \Rightarrow \neg X$, the team can make the decision of fair trading when A or C occurs, in other words, alerts caused by A or C can be ignored. This example provides an insight into the importance of negative association rule mining. On the other hand, the development of negative association rule mining will allow companies to hunt more business chances—through using infrequent itemsets of interest—than those that only take into account frequent itemsets.

Mining negative association rules is a difficult task, due to the fact that there are essential differences between positive and negative association rule mining. We illustrate this using an example as follows.

Example 1. Consider a database $TD = \{(A, B, D); (B, C, D); (B, D); (B, C, D, E); (A, B, D, F)\}$, which has 5 transactions, separated by semicolons. Each transaction contains several items, separated by commas.

For this database, negative association rules can be generated from **49 infrequent itemsets**:

$$AC, AE, AF, BE, BF, CE, CF, DE, DF, EF, ABC, ABE, ABF, ACD, ACE,$$
$$ACF, ADE, ADF, AEF, BCE, BCF, BDE, BDF, BEF, CDE, CDF, CEF,$$
$$DEF, ABCD, ABCE, ABCF, ABDE, ABDF, ABEF, ACDE, ACDF,$$
$$ACEF, ADEF, BCDE, BCDF, BDEF, CDEF,$$
$$ABCDE, ABCDF, ABCEF, ABDEF, ACDEF, BCDEF, ABCDEF.$$

There are at least **110 possible negative rules** from the infrequent itemset *ABCDEF* only, and there are at least **818** possible negative association rules from the 49 infrequent itemsets.

The above observations have shown that we must search a huge amount (at least 818) of negative association rules although the database is rather small. It would be difficult for a user to browse negative rules when the database is larger. In particular, it is a challenge to identify which of the rules are really useful to applications.

We will attack two key problems in negative association rule mining: (i) how to effectively search for interesting itemsets, and (ii) how to effectively identify negative association rules of interest. The rest of this paper is organized as follows. In the next section, we present some related concepts and definitions. We design a procedure for identifying both frequent and infrequent itemsets of interest in Section 3. In Section 4, we construct a model for measuring positive and negative association rules. Section 5 reviews related work, and Section 6 presents our experimental results.

2. PRELIMINARIES

Let $I = \{i_1, i_2, \ldots, i_N\}$ be a set of N distinct literals called *items*. D is a set of variable length transactions over I. Each transaction contains a set of items $i_1, i_2, \ldots, i_k \in I$. A transaction has an associated unique identifier called *TID*. An *association rule* is an implication of the form $A \Rightarrow B$ (or written as $A \rightarrow B$), where $A, B \subset I$, and $A \cap B = \emptyset$. A is called the *antecedent* of the rule, and B is called the *consequent* of the rule.

In general, a set of items (such as the antecedent or the consequent of a rule) is called an *itemset*. For simplicity, an itemset $\{i_1, i_2, i_3\}$ is sometimes written as $i_1 i_2 i_3$.

The number of items in an itemset is the *length* (or the *size*) of an itemset. Itemsets of length k are referred to as k-itemsets. For an itemset $A \cdot B$, if B is an m-itemset then B is called an *m-extension* of A.

Each itemset has an associated statistical measure called *support*, denoted as *supp*. For an itemset $A \subset I$, $supp(A) = s$, if the fraction of transactions in D containing A equals s.

An association rule $A \Rightarrow B$ has a measure of its strength called *confidence* (denoted as *conf*) defined as the ratio $supp(A \cup B)/supp(A)$, where $A \cup B$ means that both A and B are present.

Association rule discovery seeks rules of the form $A \Rightarrow B$ with support and confidence greater than, or equal to, user-specified minimum support (ms) and minimum confidence (mc) thresholds respectively, where

—A and B are disjoint itemsets, that is, $A \cap B = \emptyset$;
—$supp(A \Rightarrow B) = supp(A \cup B)$; and
—$conf(A \Rightarrow B) = supp(A \cup B)/supp(A)$.

This is referred to as the support-confidence framework [Agrawal et al. 1993b][1] and the rule $A \Rightarrow B$ is an *interesting positive rule*. Association analysis can be decomposed into the following two issues:

(1) Generate all large itemsets: All itemsets that have a support greater than or equal to the user specified minimum support are generated.
(2) Generate all the rules that have a minimum confidence in the following naive way: For every large itemset X and any $B \subset X$, let $A= X - B$. If the rule $A \Rightarrow B$ has the minimum confidence (or $supp(X)/supp(A) \geq mc$), then it is a valid rule.

A *frequent itemset* (also called large itemset [Chen et al. 1996]) is an itemset that meets the user-specified minimum support. Accordingly, we define an *infrequent itemset* (or small itemset) as an itemset that does not meet the user-specified minimum support.

The *negation* of an itemset A is indicated by $\neg A$. The support of $\neg A$, $supp(\neg A) = 1 - supp(A)$. In particular, for an itemset $i_1 \neg i_2 i_3$, its support is $supp(i_1 \neg i_2 i_3) = supp(i_1 i_3) - supp(i_1 i_2 i_3)$.

We call a rule of the form $A \Rightarrow B$ a *positive rule*, and rules of the other forms ($A \Rightarrow \neg B$, $\neg A \Rightarrow B$ and $\neg A \Rightarrow \neg B$) *negative rules*. For convenience, we often use only the form $A \Rightarrow \neg B$ to represent and describe negative association rules in this paper.

Like positive rules, a negative rule $A \Rightarrow \neg B$ also has a measure of its strength, *conf*, defined as the ratio $supp(A \cup \neg B)/supp(A)$.

By extending the definition in [Agrawal et al. 1993b and Chen et al. 1996], negative association rule discovery seeks rules of the form $A \Rightarrow \neg B$ with their support and confidence greater than, or equal to, user-specified minimum support and minimum confidence thresholds respectively, where

—A and B are disjoint itemsets, that is, $A \cap B = \emptyset$;
—$supp(A) \geq ms$, $supp(B) \geq ms$ and $supp(A \cup B) < ms$;
—$supp(A \Rightarrow \neg B) = supp(A \cup \neg B)$;
—$conf(A \Rightarrow \neg B) = supp(A \cup \neg B)/supp(A) \geq mc$.

The rule $A \Rightarrow \neg B$ is referred to as an *interesting negative rule*.

[1]There are also other ways, such as the FT-tree method [Han et al. 2000], that can find association rules.

Example 2. (Adapted from Brin et al. [1997].) Suppose we have a market basket database from a grocery store, consisting of n baskets. Let us focus on the purchase of tea (denoted by t) and coffee (denoted by c).

When $supp(t) = 0.25$ and $supp(t \cup c) = 0.2$, we can apply the support-confidence framework for a potential association rule $t \Rightarrow c$. The support for this rule is 0.2, which is fairly high. The confidence is the conditional probability that a customer who buys tea also buys coffee: $conf(t \Rightarrow c) = supp(t \cup c)/supp(t) = 0.2/0.25 = 0.8$, which is very high. In this case, we would conclude that the rule $t \Rightarrow c$ is a valid one.

Now consider $supp(c) = 0.6$, $supp(t) = 0.4$, $supp(t \cup c) = 0.05$, and $mc = 0.52$. The confidence of $t \Rightarrow c$ is $supp[t \cup c]/supp[t] = 0.05/0.4 = 0.125 < mc = 0.52$ and, $supp(t \cup c) = 0.05$ is low. This indicates that $t \cup c$ is an infrequent itemset and that, $t \Rightarrow c$ cannot be extracted as a rule in the support-confidence framework. However, $supp[t \cup \neg c] = supp[t] - supp[t \cup c] = 0.4 - 0.05 = 0.35$ is high, and the confidence of $t \Rightarrow \neg c$ is the ratio $supp[t \cup \neg c]/supp[t] = 0.35/0.4 = 0.875 > mc$. Therefore $t \Rightarrow \neg c$ is a valid rule from the database.

3. IDENTIFYING INTERESTING ITEMSETS

As we have seen, there can be an exponential number of infrequent itemsets in a database, and only some of them are useful for mining association rules of interest. Therefore, pruning is critical to efficient search for interesting itemsets. In this section, we design a pruning strategy, and a procedure for identifying positive and negative itemsets of interest.

3.1 A Pruning Strategy

Piatetsky-Shapiro [1991] argued that a rule $X \Rightarrow Y$ is not interesting if

$$supp(X \cup Y) \approx supp(X) \times supp(Y). \tag{1}$$

One interpretation of this proposition is that a rule is not interesting if its antecedent and consequent are approximately independent. To operationalize this concept, we can define an interestingness function $interest(X, Y) = |supp(X \cup Y) - supp(X)supp(Y)|$ and a threshold mi (minimum interestingness). If $interest(X, Y) \geq mi$, the rule $X \Rightarrow Y$ is of potential interest, and $X \cup Y$ is referred to as a **potentially interesting itemset**. Using this approach, we can establish an effective pruning strategy for efficiently identifying all frequent itemsets of potential interest in a database.

Integrating this $interest(X, Y)$ mechanism into the support-confidence framework, I is a *frequent itemset of potential interest* if:

$$\begin{aligned} fipi(I) = \; & supp(I) \geq ms \wedge \\ & \exists X, Y : X \cup Y = I \wedge \\ & fipis(X, Y) \end{aligned} \tag{2}$$

where

$$\begin{aligned} fipis(X, Y) = \; & X \cap Y = \emptyset \wedge \\ & f(X, Y, ms, mc, mi) = 1 \end{aligned} \tag{3}$$

$$f(X, Y, ms, mc, mi)$$

$$= \frac{supp(X \cup Y) + conf(X \Rightarrow Y) + interest(X, Y) - (ms + mc + mi) + 1}{|supp(X \cup Y) - ms| + |conf(X \Rightarrow Y) - mc| + |interest(X, Y) - mi| + 1}$$

where $f()$ is a constraint function concerning the support, confidence, and interestingness of $X \Rightarrow Y$.

On the other hand, to mine negative association rules, all itemsets for possible negative association rules in a given database need to be considered. For example, if $A \Rightarrow \neg B$ can be discovered as a valid rule, then $supp(A \cup \neg B) \geq ms$ must hold. If the ms is high, $supp(A \cup \neg B) \geq ms$ would mean that $supp(A \cup B) < ms$, and itemset $A \cup B$ cannot be generated as a frequent itemset in existing association analysis algorithms. In other words, $A \cup B$ is an infrequent itemset. However, there are too many infrequent itemsets in a large database, and we must define some conditions for identifying infrequent itemsets of interest.

If A is a frequent itemset and B is an infrequent itemset with a frequency of 1 in a large database, then $A \Rightarrow \neg B$ certainly looks like a valid negative rule, because $supp(A) \geq ms$, $supp(B) \approx 0$, $supp(A \cup \neg B) \approx supp(A) \geq ms$, $conf(A \Rightarrow \neg B) = supp(A \cup \neg B)/supp(A) \approx 1 \geq mc$. This could indicate that the rule $A \Rightarrow \neg B$ is valid, and the number of this type of itemsets in a given database can be very large. For example, rarely purchased products in a supermarket are always infrequent itemsets.

However, in practice, more attention is paid to frequent itemsets, and any patterns mined in a database would mostly involve frequent itemsets only. This means that if $A \Rightarrow \neg B$ (or $\neg A \Rightarrow B$, or $\neg A \Rightarrow \neg B$) is a negative rule of interest, A and B would be frequent itemsets.[2] In particular, as Example 1 has illustrated the difficulty of mining negative association rules, in this paper we focus on identifying the associations among frequent itemsets. To operationalize this insight, we can use the support measure $supp$. If $supp(X) \geq ms$ and $supp(Y) \geq ms$, the rule $X \Rightarrow \neg Y$ is of potential interest, and $X \cup Y$ is referred to as a **potentially interesting itemset**.

Integrating the above insight and the $interest(X, Y)$ mechanism into the support-confidence framework, J is an *infrequent itemset of potential interest* if:

$$\begin{aligned} iipis(J) = \; & supp(J) < ms \wedge \\ & \exists X, Y : X \cup Y = J \wedge \\ & iipis(X, Y) \end{aligned} \tag{4}$$

[2]This is a heuristic we have adopted in our method. This heuristic does not examine all possible negative itemsets. Suppose {A,B} is a frequent itemset and C is a frequent item. {A,B,C} is of interest even if it is a 3-item infrequent itemset. However, if D is an infrequent item, there is no need to consider {A,B,C,D} because D will not be in either a positive association or a negative association of interest by our heuristic. D will not be in any positive association by any existing association analysis method either. Therefore, unlike the downward closure property of the support measure, our heuristic does not possess a closure property. However, as we will demonstrate in Theorem 1 (Section 3.2), our approach will be able to find both frequent and infrequent itemsets that meet our heuristic.

where

$$iipis(X, Y) = X \cap Y = \emptyset \wedge g(X, \neg Y, ms, mc, mi) = 2 \quad (5)$$

$$g(X, \neg Y, ms, mc, mi) = f(X, \neg Y, ms, mc, mi) + \frac{supp(X) + supp(Y) - 2ms + 1}{|supp(X) - ms| + |supp(Y) - ms| + 1}$$

where $g()$ is a constraint function concerning $f()$ and the support, confidence, and interestingness of $X \Rightarrow Y$.

Note that, we can also define infrequent itemsets of potential interest for rules of the forms of $\neg X \Rightarrow Y$ and $\neg X \Rightarrow \neg Y$ accordingly. This article uses only the form of $X \Rightarrow \neg Y$ to represent and describe negative rules for convenience.

Using the *fipi* and *iipi* mechanisms for both positive and negative rule discovery, our search is constrained to seeking interesting rules on certain measures, and pruning is the removal of all uninteresting branches that cannot lead to an interesting rule that would satisfy those constraints.

3.2 Searching for Frequent and Infrequent Itemsets of Interest

Many frequent itemsets relate to positive rules that are not of interest, and many infrequent itemsets relate to negative rules that are not of interest. The search space can be significantly reduced if the extracted itemsets are restricted to frequent and infrequent itemsets of potential interest. For this reason, we now construct an efficient algorithm for finding frequent itemsets of potential interest and infrequent itemsets of potential interest in a database.

PROCEDURE 1. *AllItemsetsOfInterest*

Input: D: a database; ms: minimum support; mc: minimum confidence; mi: minimum interestingness;

Output: PL: set of frequent itemsets of interest; NL: set of infrequent itemsets of interest;

(1) **let** $PL \leftarrow \emptyset$; $NL \leftarrow \emptyset$;

(2) **let** $L_1 \leftarrow$ {frequent 1-itemsets}; $PL \leftarrow PL \cup L_1$;

(3) **for** $(k = 2; (L_{k-1} \neq \emptyset); k++)$ **do**

begin //Generate all possible frequent and infrequent k-itemsets of interest in D.

(3.1) **let** $Tem_k \leftarrow \{\{x_1, \ldots, x_{k-2}, x_{k-1}, x_k\} \mid \{x_1, \ldots, x_{k-2}, x_{k-1}\} \in L_{k-1} \wedge \{x_1, \ldots x_{k-2}, x_k\} \in L_{k-1}\}$;

(3.2) **for** each transaction t in D **do**

begin

//Check which k-itemsets are included in transaction t.

let $Tem_t \leftarrow$ the k-itemsets in t that are also contained in Tem_k;

for each itemset A in Tem_t **do**

let $A.count \leftarrow A.count + 1$;

end

(3.3) **let** $L_k \leftarrow \{c | c \in Tem_k \wedge (supp(c) = (c.count/|D|) >= ms)\}$;
let $N_k \leftarrow Tem_k - L_k$;
(3.4) //Prune all uninteresting k-itemsets in L_k
for each itemset i in L_k **do**
if NOT($fipi(I)$) **then**
let $L_k \leftarrow L_k - \{I\}$;
let $PL \leftarrow PL \cup L_k$;
(3.5) //Prune all uninteresting k-itemsets in N_k
for each itemset J in N_k **do**
if NOT($iipi(J)$) **then**
let $N_k \leftarrow N_k - \{J\}$;
let $NL \leftarrow NL \cup N_k$;
end
(4) **output** PL and NL;
(5) **return**.

The *AllItemsetsOfInterest* procedure generates all frequent and infrequent itemsets of interest in a given database D, where PL is the set of all frequent itemsets of interest in D, and NL is the set of all infrequent itemsets of interest in D. PL and NL contain only frequent and infrequent itemsets of interest respectively.

The initialization is done in Step (1). Step (2) generates L_1 of all frequent 1-itemsets in database D in the first pass of D.

Step (3) generates L_k and N_k for $k \geq 2$ by a loop, where L_k is the set of all frequent k-itemsets of interest in the kth pass of D, N_k is the set of all infrequent k-itemsets of interest, and the end-condition of the loop is $L_{k-1} = \emptyset$. For each pass of the database in Step (3), say pass k, there are five substeps as follows.

Step (3.1) generates Tem_k of all k-itemsets in D, where each k-itemset in Tem_k is generated by two frequent itemsets in L_{k-1}. Each itemset in Tem_k is counted in D by a loop in Step (3.2). Then L_k and N_k are generated in Step (3.3). L_k is the set of all potentially useful frequent k-itemsets in Tem_k, where all frequent k-itemsets in L_k meet ms. N_k is the set of all infrequent k-itemsets in Tem_k, whose supports do not meet ms, and $N_k = Tem_k - L_k$. And N_k is the set of all potentially useful infrequent k-itemsets in Tem_k.

Steps (3.4) and (3.5) select all frequent and infrequent k-itemsets of interest respectively. In Step (3.4), if an itemset I in L_k does not satisfy $fipi(I)$, then I is an uninteresting frequent itemset, and is removed from L_k. After all uninteresting frequent itemsets are removed from L_k, L_k is merged into PL. In Step (3.5), if an itemset J in N_k does not satisfy $iipi(J)$, then J is an uninteresting infrequent itemset, and is removed from N_k. After all uninteresting frequent itemsets are removed from N_k, N_k is merged into NL.

Step (4) outputs the frequent and infrequent itemsets of potential interest in PL and NL. The procedure ends in Step (5).

Table I. A Transaction Database *TD*

Transaction ID	Items
T_1	A, B, D
T_2	A, B, C, D
T_3	B, D,
T_4	B, C, D, E
T_5	A, E,
T_6	B, D, F,
T_7	A, E, F,
T_8	C, F,
T_9	B, C, F
T_{10}	A, B, C, D, F

We have a theorem for the above algorithm as follows.

THEOREM 1. *Algorithm AllItemsetsOfInterest works correctly.*

PROOF. Clearly, because this algorithm is Apriori-like, *AllItemsetsOfInterest* can generate all frequent itemsets that satisfy our constraints for interesting frequent itemsets. We need to show that all interesting infrequent itemsets are also identified.

For any itemset c in Tem_k in Step (3.3), if $supp(c) < ms$, c is appended into N_k. This means that N_k is the set of all possible infrequent k-itemsets in Tem_k. Furthermore, all infrequent itemsets of interest in N_k are selected in Step (3.5). The infrequent itemsets selected from N_k satisfy our constraints for interesting infrequent itemsets. This means that all interesting infrequent k-itemsets are identified. Therefore, *AllItemsetsOfInterest* can generate all infrequent itemsets that satisfy our constraints for interesting infrequent itemsets. □

3.3 An Example

Since the *conf* measure is constructed in the next section, in order to illustrate the use of *AllItemsetsOfInterest*, we temporarily replace the $f(X, Y, ms, mc, mi)$ constraint with the following $f(X, Y, ms, mi)$,

$$f(X, Y, ms, mi) = \frac{supp(X \cup Y) + interest(X, Y) - (ms + mi) + 1}{|supp(X \cup Y) - ms| + |interest(X, Y) - mi| + 1}$$

Example 3. Suppose we have a transaction database *TD* with 10 transactions in Table I from a grocery store. Let $A = bread$, $B = coffee$, $C = tea$, $D = sugar$, $E = beer$, $F = butter$, $ms = 0.3$ and $mi = 0.05$.

In Table I, there are six 1-itemsets: A, B, C, D, E, and F. When $ms = 0.3$, they are all frequent 1-itemsets in PL, which is listed in Table II. $L_1 = \{A, B, C, D, E, F\}$.

Tem_2 of 2-itemsets from L_1 is: $\{AB, AC, AD, AE, AF, BC, BD, BE, BF, CD, CE, CF, DE, DF, EF\}$. When $ms = 0.3$, $L_2 = \{AB, AD, BC, BD, BF, CD, CF\}$, and $N_2 = \{AC, AE, AF, BE, CE, DE, DF, EF\}$.

Table II. Single Frequent Items in *TD*

Item	Number of Transactions	Support
A	5	0.5
B	7	0.7
C	5	0.5
D	6	0.6
E	3	0.3
F	5	0.5

Table III. Frequent 2-Itemsets of Interest in *PL*

Item	Number of Transactions	Support
AB	3	0.3
BC	4	0.4
BD	6	0.6
BF	3	0.3
CF	3	0.3

When $mi = 0.07$, $L_2 = \{AB, BC, BD, BF, CF\}$ is the set of frequent 2-itemsets of interest to be appended into PL, as listed in Table III. Because

$$f(A, B, ms, mi) = \frac{0.3 + 0.05 - (0.3 + 0.05) + 1}{|0.3 - 0.3| + |0.05 - 0.05| + 1} = 1$$

$$f(A, D, ms, mi) = \frac{0.3 + 0 - (0.3 + 0.05) + 1}{|0.3 - 0.3| + |0 - 0.05| + 1} < 1$$

$$f(B, C, ms, mi) = \frac{0.4 + 0.05 - (0.3 + 0.05) + 1}{|0.4 - 0.3| + |0.05 - 0.05| + 1} = 1$$

$$f(B, D, ms, mi) = \frac{0.6 + 0.18 - (0.3 + 0.05) + 1}{|0.6 - 0.3| + |0.18 - 0.05| + 1} = 1$$

$$f(B, F, ms, mi) = \frac{0.3 + 0.05 - (0.3 + 0.05) + 1}{|0.3 - 0.3| + |0.05 - 0.05| + 1} = 1$$

$$f(C, D, ms, mi) = \frac{0.3 + 0 - (0.3 + 0.05) + 1}{|0.3 - 0.3| + |0 - 0.05| + 1} < 1$$

$$f(C, F, ms, mi) = \frac{0.3 + 0.05 - (0.3 + 0.05) + 1}{|0.3 - 0.3| + |0.05 - 0.05| + 1} = 1$$

AD and CD are not of interest, and therefore are removed from L_2.

Similarly, $N_2 = \{AC, AE, AF, BE, CE, DE, DF, EF\}$ is the set of infrequent 2-itemsets of interest to be appended into in NL, as listed in Table IV.

Tem_3 of 3-itemsets, {BCD, BCF, BDF}, is constructed from L_2. Therefore, $L_3 = \{BCD\}$ and $N_3 = \{BCF, BDF\}$. When $mi = 0.07$, $L_3 = \{BCD\}$ is the set of frequent 3-itemsets of interest to be appended into PL, as listed in Tables V and VI.

Tem_4 of 4-itemsets, {}, is constructed from L_3. Step (3) now ends. The results listed in Tables II–VI are output in Step (4).

Table IV. Infrequent 2-Itemsets of Interest in *NL*

Item	Number of Transactions	Support	Item	Number of Transactions	Support
AC	2	0.2	AE	2	0.2
AF	2	0.2	BE	1	0.1
CE	1	0.1	DE	1	0.1
DF	2	0.2	EF	1	0.1

Table V. Frequent 3-Itemsets of Interest in *PL*

Item	Number of Transactions	Support
BCD	4	0.4

Table VI. Infrequent 3-Itemsets of Interest in *NL*

Itemset	Number of Transactions	Support
BCF	2	0.2
BDF	2	0.2

From Tables II–VI, there are 10 frequent k-itemsets ($k \geq 2$), and only 6 frequent k-itemsets ($k \geq 2$) of interest in *PL*. There are 28 potentially useful infrequent k-itemsets ($k \geq 2$), and only 10 of them are of interest in *NL*. Note that we have not considered the *conf* measure in the above example, for identifying frequent and infrequent itemsets of interest. Using *conf* as a constraint, the generation of association rules will be addressed in the next section.

4. EXTRACTING POSITIVE AND NEGATIVE ASSOCIATION RULES

In this section, we present a definition of four types of association rules based on Piatetsky-Shapiro's argument and probability theory, and design an algorithm for mining both positive and negative association rules of interest in databases.

4.1 Four Types of Association Rules

Recall the relationship between $p(Y|X)$ and $p(Y)$ (or $supp(Y)$) for a possible rule $X \Rightarrow Y$ in Section 3.1. Based on Piatetsky-Shapiro's argument, we can write the interestingness of an association between X and Y in the form of their statistical dependence,

$$Dependence(X, Y) = \frac{p(X \cup Y)}{p(X)p(Y)} = \frac{p(Y|X)}{p(Y)}.$$

Consider the relationship between $p(Y|X)$ and $p(Y)$, $Dependence(X, Y)$ has the following three possible cases.

(1) If $Dependence(X, Y) = 1$ or $p(Y|X) = p(Y)$, then Y and X are independent.

(2) If $Dependence(X, Y) > 1$ or $p(Y|X) > p(Y)$, then Y is positively dependent on X, and the following holds,

$$0 < p(Y|X) - p(Y) \leq 1 - p(Y).$$

In particular, we have

$$0 < \frac{p(Y|X) - p(Y)}{1 - p(Y)} \leq 1 \tag{6}$$

The bigger the ratio $(p(Y|X) - p(Y))/(1 - p(Y))$, the higher the positive dependence.

(3) If $Dependence(X, Y) < 1$ or $p(Y|X) < p(Y)$, then Y is negatively dependent on X (or $\neg Y$ is positively dependent on X), and the following holds,

$$0 > p(Y|X) - p(Y) \geq -p(Y).$$

In particular, we have

$$0 < \frac{p(Y|X) - p(Y)}{-p(Y)} \leq 1 \tag{7}$$

The bigger the ratio $(p(Y|X) - p(Y))/(-p(Y))$, the higher the negative dependence.

In the first case, the rule $X \Rightarrow Y$ and possible negative rules between X and Y are not of interest because X and Y are independent. A small neighborhood of 1, that is, $|p(Y|X) - p(Y)| < mi$, would also indicate that $X \Rightarrow Y$ and possible negative rules between X and Y are not of interest either.

The second case has been widely explored in association analysis for positive rules, which indicates that the rule $X \Rightarrow Y$ may be an association rule of interest. The last case has received little attention. In this case, because Y is negatively dependent on X, $X \Rightarrow \neg Y$ may be a negative association rule of interest.

Putting Inequalities (6) and (7) together, we can get a *conditional-probability increment ratio* function for a pair of itemsets X and Y, denoted by $CPIR$ as follows.

$$CPIR(Y|X) = \begin{cases} \frac{p(Y|X) - p(Y)}{1 - p(Y)}, & if \ \ p(Y|X) \geq p(Y), p(Y) \neq 1 \\ \frac{p(Y|X) - p(Y)}{p(Y)}, & if \ \ p(Y) > p(Y|X), p(Y) \neq 0 \end{cases} \tag{8}$$

This is the same as the certainty factor model in Shortliffe [1976].

To discover and measure both positive and negative association rules, we can take $CPIR(Y|X)$ as the confidence of the association rule between itemsets X and Y. Clearly, $confidence(X \Rightarrow Y)$ has several special cases as follows:

—If $p(Y|X) = p(Y)$, Y and X are independent in probability theory. The confidence of the association rule $X \Rightarrow Y$ would become

$$confidence(X \Rightarrow Y) = CPIR(Y|X) = 0$$

—If $p(Y|X) - p(Y) > 0$, Y is positively dependent on X. When $p(Y|X) = 1$ which is the strongest possible condition, the confidence of the association rule $X \Rightarrow Y$ would be assigned as

$$confidence(X \Rightarrow Y) = CPIR(Y|X) = 1$$

—When $p(Y|X) = 0$, Y is negatively dependent on X, and the confidence of the association rule $X \Rightarrow \neg Y$ would be assigned as

$$confidence(X \Rightarrow \neg Y) = CPIR(Y|X) = -1$$

Because $p(\neg A) = 1 - p(A)$, for Equation (8), we can take the first half of the definition of $CPIR(Y|X)$,

$$CPIR(Y|X) = \frac{p(Y|X) - p(Y)}{1 - p(Y)}$$

or

$$CPIR(Y|X) = \frac{supp(X \cup Y) - supp(X)supp(Y)}{supp(X)(1 - supp(Y))} \tag{9}$$

as a metric for the confidence measure, *conf*, of the rule $X \Rightarrow Y$ in the following discussion when $supp(X \cup Y) \geq supp(X)supp(Y)$ and $supp(X)(1 - supp(Y)) \neq 0$, where $supp(Y|X)$ in the certainty factor model is replaced with $supp(X \cup Y)/supp(X)$ for the convenience of mining association rules. We now present a definition for association rules of interest by this metric.

Definition 1. Let I be the set of items in a database D, $i = A \cup B \subseteq I$ be an itemset, $A \cap B = \emptyset$, $supp(A) \neq 0$, $supp(B) \neq 0$, and ms, mc and $mi > 0$ be given by the user. Then,

(1) If $supp(A \cup B) \geq ms$, $interest(A, B) \geq mi$, and $CPIR(B|A) \geq mc$, then $A \Rightarrow B$ is a positive rule of interest.
(2) If $supp(A \cup \neg B) \geq ms$, $supp(A) \geq ms$, $supp(B) \geq ms$, $interest(A, \neg B) \geq mi$, and $CPIR(\neg B|A) \geq mc$, then $A \Rightarrow \neg B$ is a negative rule of interest.
(3) If $supp(\neg A \cup B) \geq ms$, $supp(A) \geq ms$, $supp(B) \geq ms$, $interest(\neg A, B) \geq mi$, and $CPIR(B|\neg A) \geq mc$, then $\neg A \Rightarrow B$ is a negative rule of interest.
(4) If $supp(\neg A \cup \neg B) \geq ms$, $supp(A) \geq ms$, $supp(B) \geq ms$, $interest(\neg A, \neg B) \geq mi$, and $CPIR(\neg B|\neg A) \geq mc$, then $\neg A \Rightarrow \neg B$ is a negative rule of interest.

This definition gives four types of valid association rules of interest. Case 1 defines positive association rules of interest. Three types of negative association rules are dealt with in Case 2, Case 3, and Case 4. In the above definition, $supp(*) \geq ms$ guarantees that an association rule describes the relationship between two frequent itemsets; the mi requirement makes sure that the association rule is of interest; and $CPIR(*) \geq mc$ specifies the confidence constraint.

We now demonstrate the use of the *CPIR* measure for identifying association rules from *PL* and *NL* in Example 3. Let $ms = 0.3$, $mc = 0.5$, and $mi = 0.05$.

Example 4. For itemset $B \cup D$ in *PL*,

$$CPIR(D|B) = \frac{supp(D \cup B) - supp(B)supp(D)}{supp(B)(1 - supp(D))} = \frac{0.6 - 0.7 * 0.6}{0.7 * (1 - 0.6)} = 0.643$$

And

$$f(B, D, ms, mi) = \frac{0.6 + 0.643 + 0.18 - (0.3 + 0.5 + 0.05) + 1}{|0.6 - 0.3| + |0.643 - 0.5| + |0.18 - 0.05| + 1} = 1$$

According to the definition of interesting positive rules and Equations (2) and (3) in Section 3.1, $B \Rightarrow D$ can be extracted as a valid positive rule of interest.

Example 5. For itemset $B \cup E$ in NL, $supp(B) = 0.7$, $supp(E) = 0.3$, $supp(\neg E) = 0.7$, $supp(B \cup \neg E) = 0.6$, and

$$CPIR(\neg E|B) = \frac{supp(B \cup \neg E) - supp(B)supp(\neg E)}{supp(B)(1 - supp(\neg E))} = \frac{0.6 - 0.7 * 0.7}{0.7 * (1 - 0.7)} = 0.524$$

Also,

$$\begin{aligned} f(B, \neg E, ms, mc, mi) &= \frac{0.6 + 0.524 + 0.11 - (0.3 + 0.5 + 0.05) + 1}{|0.6 - 0.3| + |0.524 - 0.5| + |0.11 - 0.05| + 1} \\ &= 1 \\ g(B, \neg E, ms, mc, mi) &= f(B, \neg E, ms, mc, mi) \\ &\quad + \frac{supp(B) + supp(E) - 2ms + 1}{|supp(B) - ms| + |supp(E) - ms| + 1} \\ &= 1 + \frac{0.7 + 0.3 - 2 * 0.3 + 1}{|0.7 - 0.3| + |0.3 - 0.3| + 1} \\ &= 2 \end{aligned}$$

According to Equations (4) and (5) in Section 3.1, $B \Rightarrow \neg E$ can be extracted as a valid negative rule of interest.

4.2 Algorithm Design

Mining both positive and negative association rules of interest can be decomposed into the following two subproblems, in a similar way to mining positive rules only.

(1) Generate the set PL of frequent itemsets and the set NL of infrequent itemsets.
(2) Extract positive rules of the form $A \Rightarrow B$ in PL, and negative rules of the forms $A \Rightarrow \neg B$, $\neg A \Rightarrow B$, and $\neg A \Rightarrow \neg B$ in NL.

Let D be a database, and ms, mc, and mi given by the user. Our algorithm for extracting both positive and negative association rules with the probability ratio model for confidence checking is designed as follows:

Algorithm 1. *PositiveAndNegativeAssociations*
Input*: D: a database; ms, mc, mi: threshold values;*
Output*: association rules;*

(1) **call** procedure *AllItemsetsOfInterest*;
(2) // Generate positive association rules in PL.
for each frequent itemset A in PL **do**
for each expression $X \cup Y = A$ and $X \cap Y = \emptyset$ **do**
begin
if *fipis*(X, Y) **then**
if $CPIR(Y|X) \geq mc$ **then**
output the rule $X \Rightarrow Y$
with confidence $CPIR(Y|X)$ and support $supp(A)$;
if $CPIR(X|Y) \geq mc$ **then**
output the rule $Y \Rightarrow X$
with confidence $CPIR(X|Y)$ and support $supp(A)$;
end;

(3) // Generate all negative association rules in NL.
for each itemset A in NL **do**
for each expression $X \cup Y = A$ and $X \cap Y = \emptyset$ **do**
if $iipis(X, Y)$ **then**
begin
if $CPIR(Y|\neg X) \geq mc$ **then**
output the rule $\neg X \Rightarrow Y$
with confidence $CPIR(Y|\neg X)$ and support $supp(\neg X|Y)$;
if $CPIR(\neg X|Y)| \geq mc$ **then**
output the rule $Y \Rightarrow \neg X$
with confidence $CPIR(\neg X|Y)$ and support $supp(Y \cup \neg X)$;
if $CPIR(\neg Y|X)| \geq mc$ **then**
output the rule $X \Rightarrow \neg Y$
with confidence $CPIR(\neg Y|X)$ and support $supp(X|\neg Y)$;
if $CPIR(X|\neg Y)| \geq mc$ **then**
output the rule $\neg Y \Rightarrow X$
with confidence $CPIR(X|\neg Y)$ and support $supp(\neg Y \cup X)$;
if $CPIR(\neg Y|\neg X)| \geq mc$ **then**
output the rule $\neg X \Rightarrow \neg Y$
with confidence $CPIR(\neg Y|\neg X)$ and support $supp(\neg X|\neg Y)$;
if $CPIR(\neg X|\neg Y)| \geq mc$ **then**
output the rule $\neg Y \Rightarrow \neg X$
with confidence $CPIR(\neg X|\neg Y)$ and support $supp(\neg Y \cup \neg X)$;
end;
(4) **return**.

PositiveAndNegativeAssociations generates not only all positive association rules in PL, but also negative association rules in NL. Step (1) calls procedure *AllItemsetsOfInterest* to generate the sets PL and NL with frequent and infrequent itemsets of interest respectively, in the database D.

Step (2) generates positive association rules of interest for an expression $X \cup Y$ of A in PL if $fipis(X, Y)$. If $CPIR(Y|X) \geq mc$, $X \Rightarrow Y$ is extracted as a valid rule of interest, with confidence $CPIR(Y|X)$ and support $supp(X \cup Y)$. If $CPIR(X|Y) \geq mc$, $Y \Rightarrow X$ is extracted as a valid rule of interest, with confidence $CPIR(X|Y)$ and support $supp(X \cup Y)$.

Step (3) generates negative association rules of interest for an expression $X \cup Y$ of A in NL if $iipis(X, Y)$. If $CPIR(Y|\neg X) \geq mc$, $\neg X \Rightarrow Y$ is extracted as a valid rule of interest. If $CPIR(\neg X|Y) \geq mc$, $Y \Rightarrow \neg X$ is extracted as a valid rule of interest. If $CPIR(\neg Y|X) \geq mc$, $X \Rightarrow \neg Y$ is extracted as a valid rule of interest. If $CPIR(X|\neg Y) \geq mc$, $\neg Y \Rightarrow X$ is extracted as a valid rule of interest. If $CPIR(\neg Y|\neg X) \geq mc$, $\neg X \Rightarrow \neg Y$ is extracted as a valid rule of interest. If $CPIR(\neg X|\neg Y) \geq mc$, $\neg Y \Rightarrow \neg X$ is extracted as a valid rule of interest.

5. RELATED WORK

There have been many research efforts reported in the literature to efficiently discover association rules, such as strongly collective itemsets [Aggarawal and Yu 1998], the chi-squared test model [Brin et al. 1997; Srikant and Agrawal 1997] and the share-based measure [Carter et al. 1997]. We review related work in this section.

In Piatetsky-Shapiro[1991] proposed that rules over a relation are of the form $C_1 \Rightarrow C_2$, where C_1 and C_2 are conditions on tuples of the relation. Such

a rule may be *exact*, meaning that all tuples that satisfy C_1 also satisfy C_2; may be *strong*, meaning that tuples satisfying C_1 almost always satisfy C_2; or may be *approximate*, meaning that some of the tuples satisfying C_1 also satisfy C_2. One important result in that paper is: a rule $X \Rightarrow Y$ is not interesting if $support(X \Rightarrow Y) \approx support(X) \times support(Y)$. This result has been widely taken as a major critique for mining association rules of interest, and we have adopted it in our *AllItemsetsOfInterest* procedure.

The most popular model for mining association rules is the support-confidence framework first proposed by Agrawal, Imielinski, and Swami [Agrawal et al. 1993b]. Generally speaking, an association rule is expected to capture a certain type of dependence among items in a database. Brin, Motwani and Silverstein [Brin et al. 1997] made the suggestion to measure the significance of association rules via a chi-squared test for correlations from classical statistics. The chi-squared test is useful because it not only captures correlations, but can also be used to detect negative implications.

Srikant and Agrawal [1997] applied the chi-square values to check whether association rules are statistically significant for implementing Piatetsky-Shapiro's argument.

Although the models based on the chi-square test are efficient, there are many limitations in these models, such as (1) the chi-square value for itemsets X and Y can only determine whether X and Y are independent or not; and (2) if the correlation is negative, it must apply other methods to determine which of $X \Rightarrow \neg Y$, $\neg X \Rightarrow Y$, and $\neg X \Rightarrow \neg Y$ can be extracted as a valid rule and, to compute the *support*, *confidence*, and *interest* for such a rule.

Another measurement of *interestingness* is Aggarawal and Yu's [1998] strongly collective itemset model for evaluating and finding itemsets for mining association rules. The *collective strength* $C(I)$ of an itemset I is defined as follows:

$$C(I) = \frac{1 - v(I)}{1 - E[v(I)]} \frac{E[v(I)]}{v(I)}$$

where $v(I)$ is the violation rate of I. This model doesn't consider Piatetsky-Shapiro's argument and, *supp* and *conf* must be redone when we discover association rules with infrequent itemsets.

The above models all concentrate on mining positive association rules. The chi-square test based algorithms mentioned negative relationships between two frequent itemsets, but have not addressed how to mine negative association rules. As we have demonstrated in the previous sections, mining negative association rules is different from discovering positive association rules in databases, and identifying negative associations raises new problems such as dealing with infrequent itemsets of interest and the amount of involved itemsets in databases. Therefore, exploring specific and efficient mining models is necessary to discover both positive and negative association rules in databases.

Recently, unexpected patterns [Padmanabhan and Tuzhilin 1998, 2000] and exceptional patterns [Hussain et al. 2000; Hwang et al. 1999; Liu et al. 1999; Suzuki 1997; Suzuki and Shimura 1996] have been investigated extensively. The unexpected patterns and exceptional patterns are referred to as *exceptions*

of rules, also known as *surprising patterns*. An exception is defined as a deviational pattern to a well-known fact, and exhibits unexpectedness. If it involves negative terms, it can be treated as a special case of negative rules.

[Savasere et al. 1998] addresses the issue of negative rule mining, called strong negative association mining. Previously discovered positive associations are combined with domain knowledge in the form of a taxonomy for mining association rules. This model is knowledge-dependent, and can discover negative associations of the form $A \not\Rightarrow B$. However, it is not clear in this model which one of $A \Rightarrow \neg B$, $\neg A \Rightarrow B$, and $\neg A \Rightarrow \neg B$ is the actual relationship between A and B. Obviously, this rule cannot be used in an automated reasoning system. Our model in this paper is different from the strong negative association mining model. First, our model does not require domain knowledge. Second, our negative association rules are given in more concrete expressions to indicate actual relationships between different itemsets. Third and most importantly, we have designed a general framework for mining both positive and negative association rules at the same time.

The most relevant research work on mining negative associations is indirect associations [Tan and Kumar 2002; Tan et al. 2000]. An itempair $\{a, b\}$ is indirectly associated via an itemset (called a *mediator*) Y if the following conditions hold:

(1) $sup(a, b) < minsupp$ (Itempair Support Condition), and
(2) There exists a non-empty itemset Y such that $\forall y_i \in Y$:
   (a) $sup(a, y_i) \geq minsupp, sup(b, y_i) \geq minsupp$ (Mediator Support Condition).
   (b) $d(a, y_i) \geq t_d, d(b, y_i) \geq t_d$ where $d(p, q)$ is a measure of the dependence between p and q (Dependence Condition) [Tan and Kumar 2002].

As in negative associations, an indirect association between an itempair $\{a, b\}$ also requires that $\{a, b\}$ is an infrequent itemset (the Itempair Support Condition). The most significant difference between negative associations and indirect associations is that a mediator is central to the concept of indirect associations. It is assumed that [Tan and Kumar 2002] a lattice of frequent itemsets, FI, has been generated using an existing algorithm such as Apriori. During each pass of candidate generation, it will find all frequent itemsets, $y_i \subseteq I - \{a, b\}$, such that both $\{a\} \cup y_i \in FI$ and $\{b\} \cup y_i \in FI$. Also, an indirect association deals with an itempair only, and a negative association rule indicates an association between two itemsets, each of which is not limited to one item only. Therefore, an indirect association can be treated as a special case for a negative association.

6. EXPERIMENTAL RESULTS

To study the effectiveness of our model, we have performed several experiments. For the first two sets of experiments, our server is Oracle 8.0.3, and the software was implemented on Sun Sparc using Java. JDBC API was used as the interface between our program and Oracle. For the last set of experiments, we used C++ on a Dell Workstation PWS650 with 2G of CPU and

Table VII. Synthetic Database Characteristics

Database Name	$\|R\|$	T	I	$\|r\|$
T5.I4	940	5	4	96953
T10.I4	987	10	4	98376
T20.I6	976	20	6	99997

2GB memory. To simplify our experiments, we only use negative association rules of the forms Itemset $\Rightarrow \neg$Item and $\neg$Itemset $\Rightarrow$ Item.

6.1 Effectiveness and Efficiency for Supermarket Basket Data

For the convenience of comparison, the first type of databases used in our experiments is supermarket basket data from the Synthetic Classification Data Sets on the Internet (http://www.kdnuggets.com/). The main properties of the databases are as follows. The total number of attributes, R, is approximately 1000, the average number T of attributes per row is 5, 10, and 20 respectively. The number $|r|$ of rows is approximately 100000. The average size I of maximal frequent sets is 2, 4, and 6 respectively. Table VII summarizes these parameters.

To evaluate the effectiveness, we compare our proposed approach with the support-confidence framework proposed in [Agrawal et al. 1993b] on discovering positive association rules. When mining positive association rules of interest, a rule $X \Rightarrow Y$ is of interest if and only if it satisfies four conditions: (1) $X \cap Y = \emptyset$; (2) $supp(X \cup Y) \geq ms$; (3) $|supp(X \cup Y) - supp(X)supp(Y)| \geq mi$; (4) $supp(X \cup Y)/supp(X) \geq mc$. Condition (3) is not required in the support-confidence framework, but for comparison purposes, it was added into the support-confidence framework in our experiments. Also, the domain of $CPIR(Y|X)$ is $[-1, 1]$. In our experiments, we have transformed it into interval $[0, 1]$ by using $confidence(Y|X) = (|CPIR(Y|X)| + 1)/2$. From the experimental results, the interesting positive association rules in the two models are identical.

To assess the efficiency of our proposed approach, we use two algorithms to generate all (frequent and infrequent) itemsets of interest. The first algorithm is Apriori-like, which also generates infrequent itemsets $(A \cup B)$ that satisfy conditions: (1) $A \cap B = \emptyset$; (2) $supp(A) \geq ms$ and $supp(B) \geq ms$; (3) $supp(A \cup \neg B) \geq ms$ (or $supp(\neg A \cup B) \geq ms$, or $supp(\neg A \cup \neg B) \geq ms$). This algorithm does not have any specific pruning facility, and we denote it by *MNP* (Mining with No-Pruning). The other algorithm is our *AllItemsetsOfInterest* procedure with a pruning strategy. We denote our *AllItemsetsOfInterest* procedure by *MBP* (Mining By Pruning).

Table VIII shows the running time of *MNP* and *MBP* in seconds in generating frequent itemsets.

Figures 1 and 2 illustrate the running time results of the two algorithms.

Another experiment was performed on the databases with the number of transactions ranging from 10^4 to 10^6, $R = 2000$, $T = 25$, and $I = 8$ to compare the performance of *MNP* and *MBP*. The experimental results are given in Table IX. Figure 3 depicts the experimental results, from which we can conclude that *MBP* achieves much better performance while the number of transactions is getting larger.

Table VIII. Running Time (Seconds) (0.001 and 0.0015 are two *ms* values)

Database Name	*MBP* 0.0015	*MBP* 0.001	*MNP* 0.0015	*MNP* 0.001
T5.I2.D100K	121	238	725	1987
T10.I4.D100K	388	651	2171	4278
T20.I6.D100K	759	1094	4582	7639

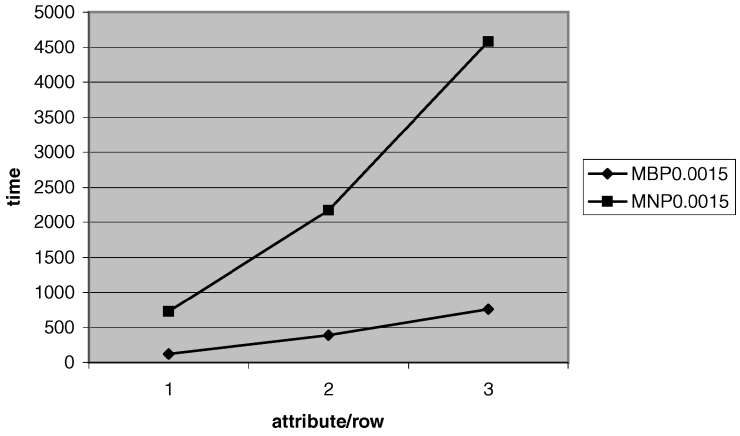

Fig. 1. The comparison of *MBP* and *MNP* when $ms = 0.0015$.

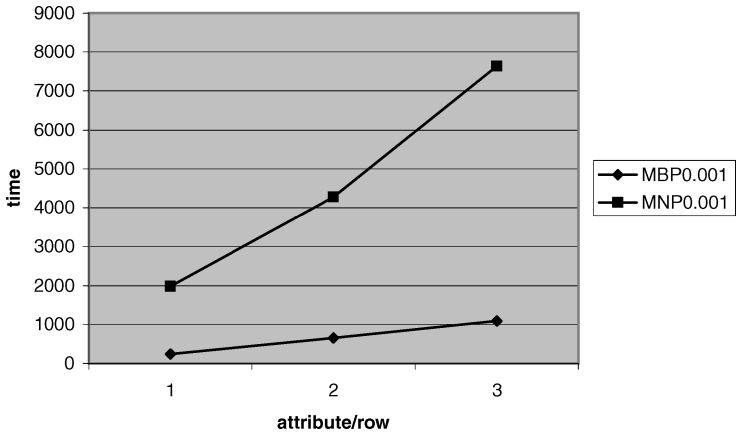

Fig. 2. The comparison of *MBP* and *MNP* when $ms = 0.001$.

6.2 Effectiveness and Efficiency of Relational Databases

One of the second type of databases used in our experiments has the following conceptual scheme, which is taken from our relational databases,

$$Report(sno, test, grade, area)$$

where *sno* is the primary key for student numbers, *test* is an attribute about examinations of subjects, *grade* is an attribute about students' grades with

Table IX. Running Time on Databases ($ms = 0.0015$)

Number of Transactions	*MBP* (seconds)	*MNP* (seconds)
10000	11.59	94.06
50000	27.48	363.37
100000	56.55	726.35
500000	221.45	3620.32
600000	268.52	4350.03
700000	312.47	5076.21
800000	357.43	5784.04
900000	402.38	6571.14
1000000	447.06	7295.31

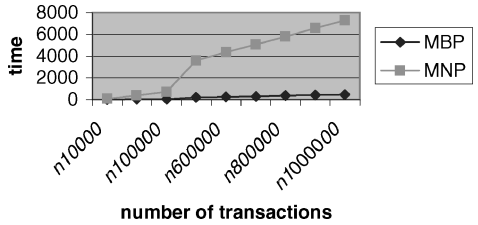

Fig. 3. Performance evaluation of *MBP* and *MNP*.

(A, B, C, D, E) as its domain, *area* is an attribute about students' nationality with a domain $(China, Singapore, \ldots)$. In order to illustrate the efficiency of our approximate rule model, we list partially the experimental results, which are the large itemsets and their supports.

We have evaluated the two methods, *MBP* and *MNP*, which are described in Section 6.1. Let $ms = 0.2$ and $mc = 0.6$. Some results are listed in Table X.

As shown in Table X, the interesting itemsets in the two models are identical. This shows that our model is effective. We also note that the running time of *MNP* is 821.23 seconds, and the time for *MBP* is 58.60 seconds. The significant reduction is clearly due to the pruning strategy, making *MBP* a promising approach for mining both positive and negative association rules.

Referring to Table X, some of the rules of interest are $area = China \Rightarrow grade = B$, and $area = China \Rightarrow \neg grade = C$.

Due to the probability significance and the constraint condition of *minsupp*, some rules such as $area = China \Rightarrow \neg grade = D$, $area = Singapore \Rightarrow \neg grade = D$, $area = China \Rightarrow \neg grade = E$ and $area = Singapore \Rightarrow \neg grade = E$, can't be extracted as negative rules of interest in our model. In some context, these rules are useful for applications though D and E are infrequent items.

6.3 Effectiveness and Efficiency of Relational Databases

The last type of databases in our experiments is the Aggregated Test Data sets that have been used for KDD Cup 2000 Data and Questions, downloaded from http://www.ecn.purdue.edu/KDDCUP/. We implemented our approach in C++ on Dell Workstation PWS650 with 2G of CPU and 2GB memory.

The selected databases are as follows:

Table X. Some Itemsets in the Original Database

Model	Useful Itemset	Support	Size of Database	Running Time
MNP	China	0.37	100000	821.23
	Singapore	0.50		
	B	0.332		
	C	0.421		
	China, *B*	0.278		
	Singapore, *C*	0.35		
	China, Singapore	0		
	China, *C*	0.31		
	B, *C*	0		
MBP	China	0.37	100000	58.60
	Singapore	0.50		
	B	0.332		
	C	0.421		
	China, *B*	0.278		
	Singapore, *C*	0.35		
	China, Singapore	0		
	China, *C*	0.31		
	B, *C*	0		

Table XI. Characteristics of the Aggregated Test Data Sets

	DB1	DB2	DB3
Items/Transaction	155	111	117
Transaction Number	1781	50558	62913

DB1: Question 3 Aggregated Test Data

DB2: Question 1 Aggregated Test Data

DB3: Question 2 Aggregated Test Data

Table XI outlines these databases.

We have evaluated the efficiency of *MNP* and *MBP* using DB1, DB2 and DB3, and the experimental results are given in Table XII.

To examine the efficiency of the pruning strategy on the support, confidence and interestingness constraints, we have run DB3 with different minimum interestingnesses. When $ms = 0.05$ and $mc = 0.4$, the experimental results are given in Table XIII.

Where 'pii' stands for positive itemsets of interest and 'nii' stands for negative itemsets of interest.

Figures 4, 5 and 6 illustrate the above results.

Table XII has demonstrated that our MBP strategy is more efficient than the Apriori-like MNP, and Table XIII has shown that the support, confidence and interestingness constraints can further improve the search efficiency.

6.4 Analysis

The results from our proposed approach for mining both positive and negative association rules of interest are promising. First, as shown in Sections 6.1 and 6.2, the positive association rules mined by the proposed model are identical to that by the support-confidence framework proposed in [Agrawal et al. 1993b]

Table XII. Efficiency of *MNP* and *MBP* on DB1, DB2 and DB3

	DB1	DB2	DB3
ms	0.1	0.05	0.05
mc	0.4	0.4	0.4
mi	0.05	0.01	0.01
MNP Running Time (s)	88.54	212.92	198.52
MBP Running Time (s)	11.11	23.48	25.48

Table XIII. Efficiency of the Pruning Strategy

mi	Number of pii	Number of nii	Running Time (s)
0.01	6807	7382	25.50
0.02	2935	1112	19.02
0.03	1319	225	16.16
0.04	551	72	13.67

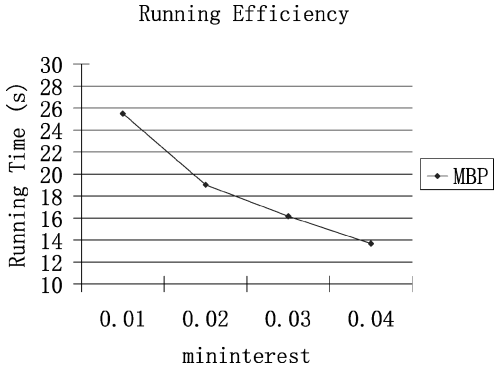

Fig. 4. The change of positive itemsets of interest with different minimum interestingnesses.

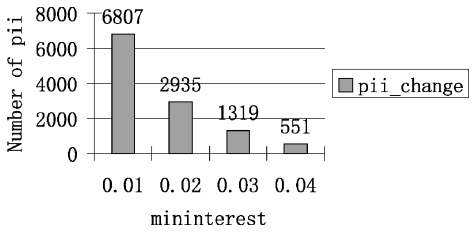

Fig. 5. The change of negative itemsets of interest with different minimum interestingnesses.

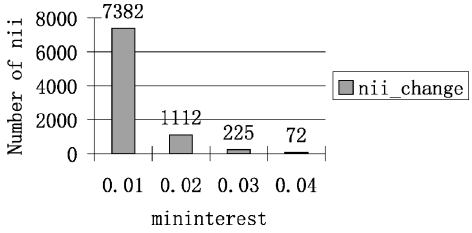

Fig. 6. Running time with different minimum interestingnesses.

when the condition $|supp(X \cup Y) - supp(X)supp(Y)| \geq mi$ is added into the support-confidence framework in our experiments. However, our proposed approach can also discover negative association rules.

Identifying frequent itemsets is a procedure for searching an exponential space that consists of all possible combinations of items and itemsets in a given database. This is necessary because the items are irregularly combined in the transactions of a database, and the considered itemsets in the database are apparently of an exponential amount. In particular, when infrequent itemsets are also considered for identifying negative association rules, we need to search the entire itemset space. For example, in the database T5.I2.D100K (see Section 6.1), there are 1000 distinct attributes (items). There are then 2^{1000} possible itemsets occurring in the database T5.I2.D100K. Clearly it is impossible to explore such a search space using the support-confidence model.

We have designed a pruning strategy (see Section 3.1) to efficiently reduce the search space. Further efficiency is gained by our interestingness measure, which allows us to greatly reduce the number of associations that the users need to consider. For example, as we have argued (in Example 1), a 6-itemset can generate 110 possible negative rules. This also leads to an exponential analysis space that consists of all possible negative association rules for a long itemset. The efficiency of our method has been presented from the experimental results in Section 6.3 that our proposed approach is more efficient than Apriori-like algorithms in generating itemsets of interest. From Section 6.3, each of support, confidence, and interestingness constraints can efficiently reduce the search space.

Association rule mining has firmly rooted in market data analysis. Negative association rule mining potentially assists automated prediction of trends and behaviors. Existing mining techniques and this research have paved a way to tackle real-world data. However, including our method in this paper, existing association analysis algorithms have made the assumption that the users can specify the minimum-support, minimum-confidence and minimum-interest thresholds. In real-world applications, mining different databases requires different thresholds. This means that the user-specified thresholds are appropriate to a database only if the distribution of itemsets in the database are known. In other words, the specification of suitable thresholds is database-dependent. We are currently developing database-independent mining strategies.

7. CONCLUSIONS

Decision making in many applications such as product placement and investment analysis often involves a number of factors, some of which play beneficial roles and others play harmful roles. We need to minimize the harmful impacts as well as maximize possible benefits. Negative association rules such as $A \Rightarrow \neg C$ are very important in decision making because $A \Rightarrow \neg C$ can tell us that C (which may be a harmful factor) rarely occurs when A (which may be an beneficial factor) occurs.

In this paper, we have designed a new method for efficiently mining both positive and negative association rules in databases. Our approach is novel and different from existing research efforts on association analysis. Some infrequent itemsets are of interest in our method but not in existing research efforts. We have designed constraints for reducing the search space, and have used the increasing degree of the conditional probability relative to the prior probability to estimate the confidence of positive and negative association rules. Our experimental results have demonstrated that the proposed approach is effective, efficient and promising.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their constructive comments on the first version of this paper. These comments have contributed to a vast improvement of this paper.

Also, we would like to thank Jingli Lu and Ling Yu for their contributions to the experiments of this paper.

REFERENCES

AGGARAWAL, C. AND YU, P. 1998. A new framework for itemset generation. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. ACM, Seattle, Washington, 18–24.

AGRAWAL, R., IMIELINSKI, T., AND SWAMI, A. 1993a. Database mining: A performance perspective. *IEEE Trans. Knowledge and Data Eng. 5*, 6 (Nov.), 914–925.

AGRAWAL, R., IMIELINSKI, T., AND SWAMI, A. 1993b. Mining association rules between sets of items in massive databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*. ACM, Washington D.C., 207–216.

BAYARDO, B. 1998. Efficiently mining long patterns from databases. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*. ACM, Seattle, Washington, 85–93.

BRIN, S., MOTWANI, R., AND SILVERSTEIN, C. 1997. Beyond market baskets: Generalizing association rules to correlations. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*. ACM, Tucson, Arizona, 265–276.

CARTER, C., HAMILTON, H., AND CERCONE, N. 1997. Share based measures for itemsets. In *Principles of Data Mining and Knowledge Discovery*. Springer, Trondheim, Norway, 14–24.

CHEN, M., HAN, J., AND YU, P. 1996. Data mining: An overview from a database perspective. *IEEE Trans. Knowledge and Data Eng. 8*, 6 (Nov.), 866–881.

HAN, J., PEI, J., AND YIN, Y. 2000. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. ACM, Dallas, Texas, 1–12.

HUSSAIN, F., LIU, H., SUZUKI, E., AND LU, H. 2000. Exception rule mining with a relative interestingness measure. In *Proceedings of The Third Pacific Asia Conference on Knowledge Discovery and Data Mining, PADKK 2000*. Springer, Kyoto, Japan, 86–97.

HWANG, S., HO, S., AND TANG, J. 1999. Mining exception instances to facilitate workflow exception handling. In *Proceedings of the Sixth International Conference on Database Systems for Advanced Applications (DASFAA)*. IEEE Computer Society, Hsinchu, Taiwan, 45–52.

LIU, H., LU, H., FENG, L., AND HUSSAIN, F. 1999. Efficient search of reliable exceptions. In *Proceedings of The Third Pacific Asia Conference on Knowledge Discovery and Data Mining, PADKK 1999*. Springer, Beijing, China, 194–204.

PADMANABHAN, B. AND TUZHILIN, A. 1998. A belief-driven method for discovering unexpected patterns. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*. AAAI, Newport Beach, California, USA, 94–100.

PADMANABHAN, B. AND TUZHILIN, A. 2000. Small is beautiful: discovering the minimal set of unexpected patterns. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Boston, MA, USA, 54–63.

PARK, J., CHEN, M., AND YU, P . 1997. Using a hash-based method with transaction trimming for mining association rules. *IEEE Trans. Knowl. Data Eng. 9*, 5 (Sept.), 813–824.

PIATETSKY-SHAPIRO, G. 1991. Discovery, analysis, and presentation of strong rules. In *Knowledge discovery in Databases*. AAAI/MIT, Menlo Park, Calif., USA, 229–248.

SAVASERE, A., OMIECINSKI, E., AND NAVATHE, S. 1998. Mining for strong negative associations in a large database of customer transactions. In *Proceedings of the Fourteenth International Conference on Data Engineering*. IEEE Computer Society, Orlando, Florida, 494–502.

SHINTANI, T. AND KITSUREGAWA, M. 1998. Parallel mining algorithms for generalized association rules with classification hierarchy. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*. ACM, Seattle, Washington, 25–36.

SHORTLIFFE, E. 1976. *Computer Based Medical Consultations: MYCIN*. Elsevier, New York.

SRIKANT, R. AND AGRAWAL, R. 1996. Mining quantitative association rules in large relational tables. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*. ACM, Montreal, Quebec, Canada, 1–12.

SRIKANT, R. AND AGRAWAL, R. 1997. Mining generalized association rules. *Future Generation Computer Systems 13*, 2-3 (Nov.), 161–180.

SUZUKI, E. 1997. Autonomous discovery of reliable exception rules. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*. AAAI, Newport Beach, California, USA, 259–262.

SUZUKI, E. AND SHIMURA, M. 1996. Exceptional knowledge discovery in databases based on information theory. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*. AAAI, Portland, Oregon, USA, 275–278.

TAN, P., KUMAR, V., AND SRIVASTAVA, J . 2000. Indirect association: Mining higher order dependencies in data. In *Principles of Data Mining and Knowledge Discovery*. Springer, Lyon, France, 632–637.

TAN, P. AND KUMAR, V. 2002. Mining indirect associations in web data. In *WEBKDD 2001—Mining Web Log Data Across All Customers Touch Points, Third International Workshop, San Francisco, CA, USA, August 26, 2001, Revised Papers*. Springer, San Francisco, CA, 145–166.

TSUR, D., ULLMAN, J., ABITEBOUL, S., CLIFTON, C., MOTWANI, R., NESTOROV, S., AND ROSENTHAL, A. 1998. Query flocks: A generalization of association-rule mining. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*. ACM, Seattle, Washington, USA, 1–12.

Received April 2003; revised October 2003; accepted December 2003