A statistical framework for mining substitution rules

Wei-Guang Teng, Ming-Jyh Hsieh, Ming-Syan Chen

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

Abstract. In this paper, a new mining capability, called mining of substitution rules, is explored. A substitution refers to the choice made by a customer to replace the purchase of some items with that of others. The mining of substitution rules in a transaction database, the same as that of association rules, will lead to very valuable knowledge in various aspects, including market prediction, user behaviour analysis and decision support. The process of mining substitution rules can be decomposed into two procedures. The first procedure is to identify *concrete itemsets* among a large number of frequent itemsets, where a concrete itemset is a frequent itemset whose items are statistically dependent. The second procedure is then on the substitution rule generation. In this paper, we first derive theoretical properties for the model of substitution rule mining and devise a technique on the induction of positive itemset supports to improve the efficiency of support counting for negative itemsets. Then, in light of these properties, the SRM (substitution rule mining) algorithm is designed and implemented to discover the substitution rules efficiently while attaining good statistical significance. Empirical studies are performed to evaluate the performance of the SRM algorithm proposed. It is shown that the SRM algorithm not only has very good execution efficiency but also produces substitution rules of very high quality.

Keywords: Concrete itemset; Correlation analysis; Statistical significance; Substitution rule

1. Introduction

The importance of discovering knowledge from a large database is growing at a rapid pace due to the fast increase of using computing for various applications. It is noted that analysis of past transaction data can provide very valuable information on customer buying behaviour and thus improve the quality of business decisions. In essence, it is necessary to collect and analyse a sufficient amount of sales data before any meaningful conclusion can be drawn therefrom. Because the amount of these processed data tends to be huge, it is important to devise efficient algorithms to conduct mining on these data.

Received 9 December 2002

Revised 10 February 2003

Accepted 20 June 2003

Published online 27 April 2004

Various data mining capabilities have been explored in the literature (Bayardo et al. 1999; Chen et al. 1996; Han and Kamber 2000; Ma and Hellerstein 2001). Among them, the one receiving a significant amount of research attention is on mining association rules (Agrawal et al. 1993). Given a database of sales transactions, the goal of mining an association rule is to discover the relationship that the presence of some items in a transaction will imply the presence of other items in the same transaction. Since the earlier work in Agrawal et al. (1993), several technologies on association rule mining have been developed, including (1) improvement on association rule mining (Agrawal and Srikant 1994; Han et al. 2000; Park et al. 1997); (2) incremental updating (Ayad et al. 2001; Lee et al. 2001); (3) mining of generalized (Srikant and Agrawal 1995), multilevel rules (Han and Fu 1995); (4) constraint-based rule mining (Lakshmanan et al. 1999; Pei and Han 2000) and multiple minimum supports issues (Liu et al. 1999; Wang et al. 2000); (5) temporal association rule mining (Ale and Rossi 2000; Chen and Petrounias 2000; Lee et al. 2001); and (6) frequent episodes discovery (Mannila and Rusakov 2001).

Note that, in addition to the association rules, the data in a transaction database also possesses some other consumer purchase behaviours. Specifically, it is important to understand the choice made by consumers, which, corresponding to the purchase of some items instead of others, is termed *substitution* in this paper. For example, in a grocery store, the purchase of apples may be substituted for that of pears. Intuitively, the substitutes are of analogous properties and therefore are often possible choices for customers. However, in some cases, the substitutes could be formed due to purchasing purposes. For example, the purchase of roses may be substituted for that of a Teddy bear and a box of chocolates. Also, to maximize the profit while minimizing the risk, combinations of futures, government bonds, stocks and fixed assets are usual alternatives of investment targets for investors. The mining of substitution rules in a transaction database, as that of association rules, will lead to very valuable knowledge in various aspects, including market prediction, user behaviour analysis and decision support, to name a few. Despite its importance, the mining of substitution rules, unlike that of association rules, has been little explored in the literature.

Mining negative association rules of the form $X \rightarrow \bar{Y}$, where $\bar{Y}$ means the absence of itemset Y , is useful for the mining of substitutive itemsets because, in a negative association rule, the presence of the antecedent itemset implies the absence of the positive counterpart of the consequent itemset, meaning that X could be a substitute for Y . It is noted that some efforts were elaborated upon the mining of negative association rules where both items and their complements in a transaction database are considered. In Savasere et al. (1998), the taxonomy of items is introduced and a heuristic of using similarity among items in the same category is utilized to facilitate the mining of negative association rules. On the other hand, a constraint-based approach is adopted in Boulicaut et al. (2000) to discover the negative association rules. Notice, however, that in the negative association rule mining, the dependency of items in an itemset is not considered because the itemset frequency is the only measurement when generating frequent itemsets. In contrast, to discover substitution rules, one should first determine possible itemsets that could be choices for customers. The purchasing frequency, i.e., support of an itemset, is not adequate to identify these possible substitutes. The dependency of items has to be examined to identify concrete sets of items that are really purchased together by customers. Specifically, a frequent itemset, the items of which are statistically dependent¹, is called a *concrete* itemset in this paper. Note that, if a frequent itemset is

¹ The threshold of such a dependency required will be described later.

not concrete, that itemset is likely to consist of frequent items that, though appearing together frequently due to their high individual occurrence counts, do not possess adequate dependency among themselves and are thus of little practical implication to be used as a whole in *either* the antecedent *or* the consequent of a substitution rule. In addition, the negative correlation of two itemsets should be verified if these two itemsets are considered to be substitutes for each other. As a result, to generate the substitution rules, both the concreteness of itemsets and the correlation between two itemsets should be taken into account. Without considering these aspects, the mining of negative association rules is not applicable to the mining of substitution rules.

Consequently, we develop in this paper a new model of mining substitution rules. The process of mining substitution rules can be decomposed into two procedures. The first procedure is to identify concrete itemsets among a large number of frequent itemsets. The second procedure is the substitution rule generation. Two concrete itemsets, X and Y , form a substitution rule, denoted by $X \triangleright Y$ to mean that X is a substitute for Y , if and only if (1) X and Y are negatively correlated and (2) the negative association rule $X \rightarrow \bar{Y}$ exists. Without loss of generality, the chi-square test (Hogg and Tanis 2000) is employed to identify concrete itemsets by statistically evaluating the dependency among items in individual itemsets. Moreover, the Pearson product moment correlation coefficient (Hogg and Tanis 2000; Johnson and Wichern 2002) is utilized to measure the correlation between two itemsets. Explicitly, we first derive theoretical properties for the model of mining substitution rules and devise a technique on the induction of positive itemset supports to improve the efficiency of support counting for negative itemsets. Then, in light of these properties, the SRM (substitution rule mining) algorithm is designed and implemented to discover the substitution rules efficiently while attaining good statistical significance. For comparison purposes, a companion method, which is extended from the Apriori algorithm, called the Apriori-Dual algorithm, is also implemented.

Extensive experimental studies have been conducted to provide many insights into the SRM algorithm proposed. It is shown by empirical results that the overhead of processing complement items can be greatly reduced by the use of the technique on the induction of positive itemset supports. The quality of substitution rules in terms of statistical measurements is also evaluated. It is shown by experiments that the SRM algorithm significantly outperforms the Apriori-Dual algorithm. It is noted that the SRM algorithm not only has very execution efficiency but also produces substitution rules of very high quality, as measured by the correlation and the violation ratio (Aggarwal and Yu 1998). The advantage of SRM is even more prominent when the transaction database is more sparse.

The rest of the paper is organized as follows. The framework of mining negative association rules is explored and the model of substitution rule mining is presented in Sect. 2. The SRM algorithm and an illustrative example are described in detail in Sect. 3. Several experiments are conducted in Sect. 4. This paper concludes with Sect. 5.

2. Model of substitution rule mining

To facilitate our discussion, we shall first review the framework of association rules mining in Sect. 2.1. The mining of negative association rules is described in Sect. 2.2. The model of substitution rule mining is then presented in Sect. 2.3.

2.1. Mining of association rules

As in most prior work (Agrawal and Srikant 1994; Chen et al. 1996), an itemset is a set containing one or many items. The support of an itemset X , denoted by S_X , is the fraction of transactions containing X in the whole dataset. The itemsets that meet the minimum support constraint are called frequent itemsets. A frequent itemset is also termed a large itemset in related papers (Chen et al. 1996). An association rule is an implication of the form $X \rightarrow Y$ with $X \cap Y = \emptyset$, where X and Y are both frequent itemsets. The support of the rule $X \rightarrow Y$, i.e., $\text{Sup}(X \rightarrow Y)$, is $S_{X \cup Y}$, and the confidence of the rule $X \rightarrow Y$, i.e., $\text{Conf}(X \rightarrow Y)$, is $\frac{S_{X \cup Y}}{S_X}$. Given a large database of transactions, the goal of mining association rules is to generate all rules that satisfy the user-specified constraints of minimum support and the minimum confidence, i.e., $\text{Sup}(X \rightarrow Y) \geq \text{MinSup}$ and $\text{Conf}(X \rightarrow Y) \geq \text{MinConf}$.

2.2. Mining of negative association rules

Before the description of the mining of negative association rules, we define positive and negative itemsets as follows.

Definition 2.1. An itemset X is *positive* if and only if it contains no complement items, i.e., $X = \{i_1, i_2, \dots, i_k\}$, where i_j is an item for $1 \leq j \leq k$. On the other hand, the *negative* itemset is an itemset containing one or more complement items. If a negative itemset is composed of complement items only, i.e., $\{\bar{i}_1, \bar{i}_2, \dots, \bar{i}_k\}$, then this negative itemset is called a *pure negative* itemset and can be denoted by $\bar{X}$.

2.2.1. Finding negative itemsets

According to Definition 2.1, all itemsets can be classified as either positive or negative ones. An itemset could be either positive or negative in this paper unless specified explicitly. Then, a negative association rule refers to an association rule of which either the antecedent itemset, the consequent itemset, or both are negative. An example of mining negative itemsets through a naive approach is given below for illustrative purposes. This negative association rule mining process will be extended and comparatively analysed with the SRM algorithm in Sect. 4.

Example 1. Consider the transaction database in Table 1(a). We first append the complement items to each transaction as shown in Table 1(b). For example, the transaction with $\text{TID} = 1$, i.e., $\{b, f\}$ in Table 1(a), becomes $\{\bar{a}, b, \bar{c}, \bar{d}, \bar{e}, f\}$ in Table 1(b). The resulting database in Table 1(b) is the input to the itemset generation algorithm.

Given that $\text{MinSup} = 0.3$, all the frequent itemsets can then be discovered from Table 1(b) as summarized in Table 2. Note that we are only interested in those complement items for which positive counterparts are frequent for market-basket analysis. As a result, the complement item $\bar{e}$ is not shown in Table 2 because the item e is not frequent.

Clearly, with this straightforward addition of complement items into the database, the mining of negative association rules can be performed by directly using methods devised for mining conventional association rules (Agrawal and Srikant 1994; Han

Table 1. (a) The original transaction database; (b) after complement items are added

TID	Items		TID	Items
1	b, f		1	$\bar{a}, b, \bar{c}, \bar{d}, \bar{e}, f$
2	b, c		2	$\bar{a}, b, c, \bar{d}, \bar{e}, \bar{f}$
3	c		3	$\bar{a}, \bar{b}, c, \bar{d}, \bar{e}, \bar{f}$
4	a, b, f		4	$a, b, \bar{c}, \bar{d}, \bar{e}, f$
5	a, c, d	$\Rightarrow$	5	$a, \bar{b}, c, d, \bar{e}, \bar{f}$
6	e		6	$\bar{a}, \bar{b}, \bar{c}, \bar{d}, e, \bar{f}$
7	a, c, d		7	$a, \bar{b}, c, d, \bar{e}, \bar{f}$
8	b, c, f		8	$\bar{a}, b, c, \bar{d}, \bar{e}, f$
9	a, b, e		9	$a, b, \bar{c}, \bar{d}, e, \bar{f}$
10	a, d		10	$a, \bar{b}, \bar{c}, d, \bar{e}, \bar{f}$
(a)			(b)	

Table 2. Frequent itemsets generated from the database in Table 1(b) (MinSup = 0.3)

I_1	S_I	I_2	S_I	I_2	S_I	I_3	S_I	I_4	S_I
a	0.5	a, d	0.3	$c, \bar{f}$	0.4	$a, d, \bar{b}$	0.3	$a, d, \bar{b}, \bar{f}$	0.3
b	0.5	$a, \bar{b}$	0.3	$d, \bar{b}$	0.3	$a, d, \bar{f}$	0.3		
c	0.5	$a, \bar{c}$	0.3	$d, \bar{f}$	0.3	$a, \bar{b}, \bar{f}$	0.3		
d	0.3	$a, \bar{f}$	0.4	$f, \bar{d}$	0.3	$b, f, \bar{d}$	0.3		
f	0.3	b, f	0.3	$\bar{a}, \bar{d}$	0.5	$b, \bar{a}, \bar{d}$	0.3		
$\bar{a}$	0.5	$b, \bar{a}$	0.3	$\bar{a}, \bar{f}$	0.3	$b, \bar{c}, \bar{d}$	0.3		
$\bar{b}$	0.5	$b, \bar{c}$	0.3	$\bar{b}, \bar{f}$	0.5	$c, \bar{a}, \bar{d}$	0.3		
$\bar{c}$	0.5	$b, \bar{d}$	0.5	$\bar{c}, \bar{d}$	0.4	$c, \bar{b}, \bar{f}$	0.3		
$\bar{d}$	0.7	$c, \bar{a}$	0.3	$\bar{c}, \bar{f}$	0.3	$d, \bar{b}, \bar{f}$	0.3		
$\bar{f}$	0.7	$c, \bar{b}$	0.3	$\bar{d}, \bar{f}$	0.4	$\bar{a}, \bar{d}, \bar{f}$	0.3		
		$c, \bar{d}$	0.3						

et al. 2000). However, this benefit may not be able to justify several drawbacks of this naive approach in practice. First, excessive storage space is required to store complement items and also the additional itemsets resulted. Next, many of the frequent itemsets generated are composed of complement items only. These itemsets, while incurring much computational cost for their generation, are usually of little use in real applications. Finally, extra database scans are needed for the mining process. In this example, the largest negative itemset is of size four while the largest positive itemset is of size two, meaning that two extra database scans are needed for the discovery of negative itemsets in the mining process. In real applications, this naive approach will suffer a prolonged execution time and make mining of negative association rules an infeasible task.

2.2.2. Discovering negative association rules

Once the negative itemsets are generated, one can discover all negative association rules in a straightforward manner. For two itemsets X and Y , where $Y \subset X$, the rule $Y \rightarrow (X - Y)$ is output if the required MinConf is satisfied. However, for our purpose of discovering substitution rules, two positive itemsets are required to form a substitute pair. Thus, the Apriori-Dual algorithm, i.e., a companion method extended from algorithm Apriori, is proposed to generate only rules whose antecedent is positive

and consequent is pure negative, i.e., $X \rightarrow \bar{Y}$ where X and Y are positive itemsets. This means that the presence of the antecedent itemset may lead to the absence of the positive counterpart of the consequent itemset, i.e., X is a substitute for Y . On the other hand, rules composed of both negative itemsets or a partial negative itemset provide no hint on substitutive itemsets and are thus ignored. Consequently, the rules generated by the Apriori-Dual algorithm form only a subset of all negative association rules. Also, the computation cost of algorithm Apriori-Dual is less than that of a naive negative association rule generation process.

Algorithm 2.1. Apriori-Dual

```

// Procedure of generating all frequent itemsets, including the negative ones
// Input: MinSup and MinConf
1. Append the complement items whose positive counterpart is not original
   present to each transaction.
2. Generate the set of frequent (positive and negative) items, i.e.,  $L_1$ .
3. Remove the negative items whose positive counterpart is not frequent from
    $L_1$ .
4. For  $k \geq 2$  do{
5.   Generate the candidate set of  $k$ -itemsets from  $L_{k-1}$ , i.e.,  $C_k = L_{k-1} \bowtie L_{k-1}$ .
6.   If ( $C_k$  is empty) then break;
7.   Scan the transactions to calculate supports of all candidate  $k$ -itemsets.
8.    $L_k = \{c \in C_k \mid S_c \geq \text{MinSup}\}$ ;
9. }
// Procedure of negative association rule generation
10. For each negative itemset  $X$  in  $L_k$ s do{
11.   Let  $\bar{Y}$  be the largest pure negative itemset where  $\bar{Y} \subset X$ .
12.   If ( $X - \bar{Y}$ ) is not an empty set, // ( $X - \bar{Y}$ ) is positive.
13.     If ( $\text{Conf}((X - \bar{Y}) \rightarrow \bar{Y}) \geq \text{MinConf}$ )
14.       output the rule  $(X - \bar{Y}) \rightarrow \bar{Y}$ .
15. }
```

As pointed out before, the problem formulation of the negative association rule mining is different from that of the substitution rule mining. In the negative association rule mining, the itemset frequency, i.e., support of an itemset, is the only measurement when determining whether an itemset is meaningful or not. In addition, because complement items are appended to the original transaction database, the computation cost of algorithm Apriori-Dual is, as confirmed by our experimental results, very high. These drawbacks reduce the practicability of using the Apriori-Dual algorithm for identifying substitute itemsets. Consequently, a new algorithm for mining substitution rules is proposed and will be described in later sections.

2.3. Mining of substitution rules

As mentioned before, the process of mining substitution rules can be decomposed into two procedures. The first one is to identify concrete itemsets among large amounts of itemsets. The second one is the substitution rule generation. The chi-square test (Hogg and Tanis 2000) is employed to identify concrete itemsets by statistically evaluating the dependency among items in individual itemsets. Also, the

Pearson product moment correlation coefficient (Hogg and Tanis 2000; Johnson and Wichern 2002) is utilized to measure the correlation between two itemsets.

2.3.1. Identification of concrete itemsets

Concrete itemsets are those possible itemsets that could be choices for customers with some purchasing purposes. To qualify an itemset as a concrete one, not only the purchasing frequency, i.e., support of an itemset, but also the dependency of items has to be examined to declare that these items are *purposely* purchased together by customers. To evaluate the dependence among items in an itemset (Meo 2000), one common approach is to utilize the chi-square test (Brin et al. 1997; Jermaine 2001; Liu et al. 2001). Specifically, the chi-square test of an itemset can be derived in terms of supports and expected supports of its corresponding itemsets, as stated in Theorem 1.

Theorem 2.1. Let $X = \{x_1, x_2, \dots, x_k\}$ be a positive k -itemset, the chi-square value for X is computed as

$$Chi(X) = n \times \left[\left(\sum_{I \in \{Y | Y^+ = X\}} \frac{S_I^2}{E_I} \right) - 1 \right],$$

where n is the number of total transactions, Y^+ denotes the positive itemset where all complement items in itemset Y are replaced by their positive counterparts, e.g., $\{a, \bar{b}, \bar{c}\}^+ = \{a, b, c\}$ where a, b and c are positive items, and $E_I = \prod_{i \in I} S_i$ is the expected support of I .

Proof. Because the itemset X is of size k and the presence of an item in each transaction is 0-1 valued, a corresponding $2 \times 2 \times \dots \times 2k$ -dimensional contingency table can be constructed. Each dimension of this contingency table corresponds to the presence of an item, i.e., $x_i \in X$, in each transaction. The values of these 2^k cells are exactly the supports of itemsets $\{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_k\}, \{\bar{x}_1, \bar{x}_2, \dots, x_k\}, \dots, \{x_1, x_2, \dots, x_k\}$ and the summation of these values is n , i.e., the number of transactions. Also, the corresponding itemsets above can be formulated as $\{Y \mid Y^+ = X\}$. The chi-square value is then computed by

$$Chi(X) = \sum_{c \in cells} \frac{(O_c - E_c)^2}{E_c},$$

where O_c is the observed value and E_c is the expected value of cell c in the contingency table. For any itemset I and its corresponding cell c that $I^+ = X$, we have

$$O_c = n \times S_I \text{ and } E_c = n \times \prod_{i \in I} S_i.$$

With some algebraic manipulations, we have

$$\begin{aligned} Chi(X) &= \sum_{c \in cells} \frac{O_c^2 - 2O_cE_c + E_c^2}{E_c} \\ &= \sum_{c \in cells} \frac{O_c^2}{E_c} - 2 \sum_{c \in cells} O_c + \sum_{c \in cells} E_c \end{aligned}$$

$$\begin{aligned}
&= \left(\sum_{c \in \text{cells}} \frac{O_c^2}{E_c} \right) - n \quad \left(\because \sum_{c \in \text{cells}} O_c = \sum_{c \in \text{cells}} E_c = n \right) \\
&= \left(\sum_{I \in \{Y \mid Y^+ = X\}} \frac{n^2 \times S_I^2}{n \times \prod_{i \in I} S_i} \right) - n \\
&= n \times \left[\left(\sum_{I \in \{Y \mid Y^+ = X\}} \frac{S_I^2}{E_I} \right) - 1 \right].
\end{aligned}$$

This completes the proof. $\square$

To utilize the chi-square test to verify whether the occurrences of given items are dependent, two contradictory hypotheses are made

$$\begin{cases} H_0: \text{The occurrences of all items } (x_1 \sim x_k) \text{ are independent,} \\ H_1: H_0 \text{ is rejected.} \end{cases}$$

With Theorem 1, to declare the dependency among items in an itemset X or to support hypothesis H_1 , the chi-square value for X is required to be no less than a threshold, i.e., $Chi(X) \geq \chi_{df(X), \alpha}^2$.

In addition, it follows from advanced statistics and information theory (Hosseini et al. 1991) that the corresponding degree of freedom for this test can be denoted by

$$df(X) = \prod_i c(v_i) - \sum_i [c(v_i) - 1] - 1 = 2^k - k - 1,$$

where $c(v_i)$ is the number of categories in dimension i , i.e., $c(v_i) = 2$ for all dimensions because the presence of an item in each transaction is 0-1 valued. This result of corresponding degree of freedom can be also intuitively explained as the 2^k cells in the contingency table subtracted by the number of constraints, i.e., k for k individual items and one for the total number of transactions.

We comment that the results derived in Theorem 1 are essential for our mining of substitution rules and are not subsumed by the work in Brin et al. (1997). In Brin et al. (1997), it was stated that “if S is correlated with significance level α , any superset of S is also correlated with significance level α .” From Theorem 1 in Brin et al. (1997), one may mistakenly assume that the chi-square test for itemsets at a given significance level is upward closed (as stated in Theorem 1 in Brin et al. (1997).) However, as also noted in DuMouchel and Pregibon (2001) this upward closure property is not fully correct. An example showing the chi-squared statistic is not upward closed is given in the Appendix for interested readers. Specifically, as opposed to what Theorem 1 in Brin et al. (1997) suggests, all correlated itemsets, rather than only *minimally* correlated ones, should be discovered. This in turn justifies the necessity of our development of the process to identify concrete itemsets in this paper.

Without loss of generality, a *concrete* itemset is thus defined to be a frequent itemset that is positively correlated given a significance level α (usually $\alpha = 0.05$), if it contains more than one item. Note that the significance level of a concrete itemset is expected to be at least no less than that of its subsets. For example, if the itemset {flashlight, battery} has a quite high chi-square value, then its superset,

e.g., {flashlight, battery, pencil}, could still have a high chi-square value ($> \chi_{df(X), \alpha}^2$) even though pencil is not so correlated with the other items.

Definition 2.2. A positive frequent itemset $X = \{x_1, x_2, \dots, x_k\}$ is called a *concrete* itemset if and only if

- (1) $k = 1$, or
- (2) $k \geq 2$,

$$S_X > \prod_{x_i \in X} S_{x_i} \text{ and } Chi(X) \geq \chi_{df(X), \alpha}^2,$$

where $\prod_{x_i \in X} S_{x_i}$ corresponds to the *expected* support for itemset X and $\chi_{df(X), \alpha}^2$ is the value of chi-square distribution with degree of freedom $df(X)$ at probability α . Note that $S_X > \prod_{x_i \in X} S_{x_i}$ is required to ensure that all $x_i \in X$ are positively correlated.

The value of $\chi_{df(X), \alpha}^2$ can be obtained by table look-up. As mentioned earlier, the usual value $\alpha = 0.05$ is used in this study for statistical significance. Considering itemset $\{a, d\}$ in Table 2, for example, $S_{ad} = 0.3 > S_a \times S_d = 0.6 \times 0.3$. Also, the chi-square value for $\{a, d\}$ is

$$\begin{aligned} Chi(\{a, d\}) &= n \times \left[\left(\frac{S_{ad}^2}{E_{ad}} + \frac{S_{ad}^2}{E_{ad}} + \frac{S_{ad}^2}{E_{ad}} + \frac{S_{ad}^2}{E_{ad}} \right) - 1 \right] \\ &= 10 \times \left(\frac{0.5^2}{0.5 \times 0.7} + 0 + \frac{0.2^2}{0.5 \times 0.7} + \frac{0.3^2}{0.5 \times 0.3} - 1 \right) \\ &= 4.29 > \chi_{1, 0.05}^2 = 3.84. \end{aligned}$$

Thus, $\{a, d\}$ is a concrete itemset.

2.3.2. Testing of negative correlation

To evaluate the correlation between two *concrete* itemsets, we adopt the measurement of Pearson product moment correlation coefficient (Hogg and Tanis 2000). Theorem 2 states that the correlation coefficient of two itemsets can be determined by their supports.

Theorem 2.2. Let X and Y be two itemsets with $X \cap Y = \emptyset$. The correlation coefficient of X and Y can be formulated in terms of their supports. Explicitly,

$$\rho(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var(X) \cdot Var(Y)}} = \frac{S_{XY} - S_X \cdot S_Y}{\sqrt{S_X(1 - S_X)S_Y(1 - S_Y)}}.$$

Proof. According to the definition of correlation coefficient, we have $\rho(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var(X) \cdot Var(Y)}}$, where the covariance is

$$Cov(X, Y) = E[(X - EX)(Y - EY)] = E(XY) - (EX)(EY),$$

and the variances are

$$VarX = E[(X - EX)^2] = EX^2 - (EX)^2 \text{ and}$$

$$VarY = E[(Y - EY)^2] = EY^2 - (EY)^2.$$

In the above formulas, E stands for the expected value. Because variables corresponding to occurrence of items in a transaction database are all 0-1 valued, it follows that

$$EX = EX^2 = S_X, EY = EY^2 = S_Y \text{ and } E(XY) = S_{XY}.$$

Consequently, we have

$$\begin{aligned} \rho(X, Y) &= \frac{Cov(X, Y)}{\sqrt{Var(X) \cdot Var(Y)}} \\ &= \frac{S_{XY} - S_X \cdot S_Y}{\sqrt{[S_X - (S_X)^2][S_Y - (S_Y)^2]}} \\ &= \frac{S_{XY} - S_X \cdot S_Y}{\sqrt{S_X(1 - S_X)S_Y(1 - S_Y)}}. \end{aligned}$$

This completes the proof. $\square$

Note that, when both variables to be correlated are binary, as in this case, we may use the phi coefficient of correlation as stated in Johnson and Wichern (2002) instead of $\rho(X, Y)$ in Theorem 2. However, the phi coefficient of correlation and the Pearson product moment correlation coefficient are in fact algebraically equivalent and give identical numerical results. Therefore, for notational simplicity, we employ $\rho(X, Y)$ to express the results of Theorem 2.

Consequently, a substitution rule can be defined as below.

Definition 2.3. Given two itemsets X and Y and $X \cap Y = \emptyset$, X is a substitute for Y , denoted by $X \triangleright Y$, if and only if

- (1) Both X and Y are concrete.
- (2) X and Y are negatively correlated, i.e., $\rho(X, Y) < -\rho_{\min} \leq 0$ (usually $\rho_{\min} = 0$ for simplicity). And
- (3) the negative association rule $X \rightarrow \bar{Y}$ is valid, i.e., $Sup(X \rightarrow \bar{Y}) \geq \text{MinSup}$ and $\text{Conf}(X \rightarrow \bar{Y}) \geq \text{MinConf}$.

3. SRM: substitution rule mining

Given the definitions of concrete itemsets and substitution rules, a detailed description of the SRM algorithm for mining substitution rules is given in Sect. 3.1. To reduce the overhead of support counting for negative itemsets, a technique of the induction of positive itemset supports is developed in Sect. 3.2.

3.1. Algorithm of substitution rule mining

The algorithmic form of mining substitution rules can be outlined below, where the procedure of identifying concrete itemsets and the procedure of substitution rule generation are presented.

Table 3. Frequent (positive) itemsets, their supports and chi-square values generated from Table 1(a) (concrete itemsets are in italics)

I_1	S_I	I_2	S_I	$Chi(I)$	I_3	S_I	$Chi(I)$
<i>a</i>	0.5	<i>a, b</i>	0.2	0.4	<i>a, c, d</i>	0.2	6.38
<i>b</i>	0.5	<i>a, c</i>	0.2	0.4			
<i>c</i>	0.5	<i>a, d</i>	0.3	4.29			
<i>d</i>	0.3	<i>b, c</i>	0.2	0.4			
<i>e</i>	0.2	<i>b, f</i>	0.3	4.29			
<i>f</i>	0.3	<i>c, d</i>	0.2	0.48			

Algorithm 3.1. SRM

```

// Procedure of identifying concrete itemsets
// Input: MinSup, MinConf, and  $\rho_{\min}$ 
1. Generate the set of all frequent (positive) items, i.e.,  $L_1$ 
2. For  $k \geq 2$  do{
3.   Generate the candidate set of  $k$ -itemsets from  $L_{k-1}$ , i.e.,  $C_k = L_{k-1} \bowtie L_{k-1}$ .
4.   If ( $C_k$  is empty), then break.
5.   Scan the transactions to calculate supports of all candidate  $k$ -itemsets.
6.    $L_k = \{c \in C_k \mid S_c \geq \text{MinSup}\}$ .
7.   For each frequent itemset  $X$  in  $L_k$  do{
8.     If ( $S_X > \prod_{x_i \in X} S_{x_i}$ ) && ( $Chi(X) \geq \chi_{df(X), \alpha}^2$ )
9.       Add  $X$  to the set of concrete itemsets.
10.  }
11. }
// Procedure of substitution rule generation
12. For each pair of concrete itemsets  $X, Y$  do{
13.   If ( $\rho(X, Y) < -\rho_{\min}$ )
14.     If ( $\text{Sup}(X \rightarrow \bar{Y}) \geq \text{MinSup}$ ) && ( $\text{Conf}(X \rightarrow \bar{Y}) \geq \text{MinConf}$ .)
15.       output the substitution rule  $X \triangleright Y$ ; //  $X \rightarrow \bar{Y}$  is valid.
16. }
```

The execution of the SRM algorithm can be best understood by the following example:

Example 3.1. Consider the transaction database in Table 1(a). The SRM algorithm first performs the procedure of identifying concrete itemsets, i.e., operations from line 1 to line 11 in the SRM algorithm. Given $\text{MinSup} = 0.2$ and $\text{MinConf} = 0.7$, the frequent itemsets can be first obtained as in Table 3.

The dependency among items in these frequent itemsets is then evaluated. By Definition 2.2, chi-square tests of concreteness are performed on each k -itemset for $k \geq 2$. The chi-square values of these frequent itemsets are shown in Table 3, where only two frequent itemsets are found concrete. Note that $\{a, c, d\}$ fails to pass the test because $df(\{a, c, d\}) = 2^3 - 3 - 1 = 4$ and $Chi(\{a, c, d\}) = 6.38 < \chi_{4, 0.05}^2 = 9.49$.

Next, in the procedure of substitution rule generation, i.e., operations from line 12 to line 16 in the SRM algorithm, the candidate substitution pairs can then be generated by joining these concrete itemsets. By examining the support, confidence and correlation of these candidate pairs, substitution rules can be generated as in Table 4.

Table 4. Substitution rules discovered with MinSup = 0.2, MinConf = 0.7, and $-\rho_{\min} = -0.5$

Rule($X \triangleright Y$)	Sup	Conf	Correlation(X, Y)
$\{b\} \triangleright \{d\}$	0.5	1	-0.65
$\{d\} \triangleright \{b\}$	0.3	1	-0.65
$\{a, d\} \triangleright \{b\}$	0.3	1	-0.65

3.2. Overhead reduction of support counting

3.2.1. Induction of positive itemset supports

Note that, in the operations of line 8 and line 14 in the SRM algorithm, support evaluation of negative itemsets is required to calculate the chi-square value and to validate the negative association rules, respectively. This support evaluation is costly because extra database scans for support counting are required. Note that, with proper manipulation of Boolean logic, the support of a negative itemset can be represented in terms of supports of corresponding positive itemsets. This technique of obtaining the support of a negative itemset is called “the induction of positive itemset supports.” For example, it can be verified that $S_{\overline{xy}} = 1 - S_x - S_y + S_{xy}$, where x and y are both items. Consequently, we devise the following theorem, which exploits the relationship among items and their complements to reduce the corresponding overhead of the database scan for the supports of negative itemsets.

Theorem 3.1. (Induction of positive itemset supports) Let X be a negative itemset and $X = Y \cup \overline{Z}$, where Y and Z are both positive itemsets. The support of X can be represented in terms of supports of certain positive itemsets. Explicitly, the relationship can be formulated as

$$S_X = \sum_{Z \subseteq I \subseteq (Y \cup Z)} (-1)^{[\text{len}(I) - \text{len}(Y)]} \times S_I,$$

where $\text{len}(I)$ and $\text{len}(Y)$ are lengths of itemsets I and Y , respectively.

Proof. From probability theory, the DeMorgan’s laws states that if $\{T_i\}, i \geq 1$, is any sequence of sets, then

$$\left(\bigcap_i \overline{T_i}\right) = \overline{\left(\bigcup_i T_i\right)}.$$

Also, the probability of the union of events can be generalized as

$$\begin{aligned} p(T_1 \cup T_2 \cup \dots \cup T_n) &= \sum_i p(T_i) - \sum_{i_1 < i_2} p(T_{i_1} \cap T_{i_2}) \\ &\quad + \dots + (-1)^{r-1} \cdot \sum_{i_1 < i_2 < \dots < i_r} p(T_{i_1} \cap T_{i_2} \cap \dots \cap T_{i_r}) \\ &\quad + \dots + (-1)^{n-1} \cdot p(T_{i_1} \cap T_{i_2} \cap \dots \cap T_{i_n}). \end{aligned}$$

Let $Z = \{z_1, z_2, \dots, z_k\}$. We define T_0 to be the event that itemset Y occurs in the transactions and T_i to be the events that item z_i occurs in the transactions for $1 \leq i \leq k$. By analogy, the probabilities of these events are

$$p(T_0) = S_Y \quad \text{and} \quad p(T_i) = S_{z_i} \quad \text{for} \quad 1 \leq i \leq k.$$

The joint probability of some T_i s corresponds to the support of corresponding itemset. For example, $p(T_0 \cap T_1 \cap T_3) = S_{Y \cup \{z_1, z_3\}}$.

Consequently,

$$\begin{aligned} S_X &= p(T_0 \cap \overline{T_1} \cap \dots \cap \overline{T_k}) \\ &= p(T_0 \cap \overline{(T_1 \cup \dots \cup T_k)}) \\ &= p(T_0) - p(T_0 \cap (T_1 \cup \dots \cup T_k)) \\ &= S_Y - \sum_i S_{Y \cup \{z_i\}} + \sum_{i_1 < i_2} S_{Y \cup \{z_{i_1}, z_{i_2}\}} - \dots \\ &\quad + (-1)^r \sum_{i_1 < i_2 < \dots < i_r} S_{Y \cup \{z_{i_1}, z_{i_2}, \dots, z_{i_r}\}} + \dots + (-1)^k \times S_{Y \cup Z} \\ &= \sum_{Y \subseteq I \subseteq (Y \cup Z)} (-1)^{[\text{len}(I) - \text{len}(Y)]} \times S_I. \end{aligned}$$

This completes the proof. $\square$

It is important to note that, in light of Theorem 3, repetitive database scans can be avoided because the supports of negative itemsets can be computed by those of corresponding positive itemsets. The overhead of generating negative itemsets, as mentioned in Sect. 2.2, can hence be greatly reduced.

3.2.2. Support estimation with itemsets in increasing order

As mentioned earlier, there are two operations in the SRM algorithm where support evaluation of negative itemsets is required. For the first purpose, that of calculating the chi-square value for an itemset X , the support can be directly evaluated through Theorem 3. It is because X is frequent, each itemset in that formula, which is a subset of X , is also frequent and the corresponding support is already evaluated during the generation of frequent itemsets. However, for the second purpose, that of validating the negative association rules, because some positive itemsets required could be infrequent, extra database scans of these missing terms are needed. Nevertheless, the number of itemsets to be scanned can be greatly reduced if a proper estimation of supports is made.

Note that, in Theorem 3, each term is separated by alternate plus sign and minus sign, i.e., $(-1)^r = 1$ when r is even and $(-1)^r = -1$ when r is odd. Also, the length of itemsets appearing in each term is in increasing order, which in turn suggests that the corresponding supports tend to be in decreasing order. As a result, it is expected that the sum of this series usually converges to its final value rapidly in practice. For a rule $Y \rightarrow \overline{Z}$ to be valid, it is required that $S_{Z\overline{Z}} \geq \text{MinSup}$ and $\frac{S_{Z\overline{Z}}}{S_Y} \geq \text{MinConf}$. In other words, as long as $S_{Z\overline{Z}} \geq \text{Max}(\text{MinSup}, S_Y \times \text{MinConf})$ is true, the corresponding rule is valid. In practice, we can scan only the very first few terms, e.g., $\text{len}(I) - \text{len}(Y) = 1, 2$ and 3 , in Theorem 3 during the concrete itemset

Table 5. Parameter settings of the synthetic datasets

Parameter	D_{dense}	D_{sparse}	Meaning
T	10	5	Average size of transactions
I	50	100	Number of items
D	10,000	10,000	Number of transactions

identification phase, and thus obtain a fairly good estimation of support value of required itemset. Higher order terms are being scanned in batch only when a more precise estimation is needed. Consequently, the number of extra scans required can be limited to one in practice. As will be validated by the experimental studies in Sect. 4, this technique on the induction of positive itemset supports can lead to prominent improvement of execution efficiency of the SRM algorithm.

4. Experimental results

The simulation model of our experimental studies is described in Sect. 4.1. To assess the performance of SRM, we conduct two empirical studies based on the synthetic datasets. The execution efficiency of the SRM algorithm is examined in Sect. 4.2. The quality of substitution rules generated is evaluated in Sect. 4.3.

4.1. Simulation model

To assess the performance of SRM, an algorithm, referred to as Apriori-Dual algorithm, is extended from the Apriori algorithm for comparison purposes. Recall that the Apriori algorithm primarily deals with positive items and generates positive association rules only. As mentioned in Sect. 2.2, mining negative association rules by appending complement items to the original transaction database incur both an excessive storage space and a huge computational cost. Without the process of generating rules with the required form as adopted by Apriori-Dual, the computation time of the naive approach for generating negative association rules is shown by our experiments to be longer, by several orders, than those of both the Apriori-Dual and SRM algorithms. Therefore, only the Apriori-Dual and SRM algorithms are being compared in the following experiments.

We use two synthetic datasets generated by a randomised transaction generation algorithm in Srikant and Agrawal (1995). The first one is a dense dataset with T10.I50.D10K, denoted as D_{dense} . The second one is a sparse dataset, denoted as D_{sparse} , with T5.I100.D10K. The values of parameters used to generate the datasets are summarized in Table 5, where both the environments of dense and sparse dataset distributions are considered. Note that, although the SRM algorithm is particularly useful for a sparse transaction database, we consider the case of a dense database as well to ensure the generality of our study.

4.2. Execution efficiency

The first experiment is to evaluate the execution efficiency of the SRM algorithm. Figure 1 indicates that the SRM algorithm outperforms the Apriori-Dual algorithm

Fig. 1. Total elapsed time with MinConf = 50%, $\alpha = 0.05$ and various values of MinSup

in execution efficiency by a margin from 30 to 500% for the dense dataset D_{dense} . Furthermore, in a sparse dataset D_{sparse} , the SRM algorithm is 20 times more efficient than Apriori-Dual. It is noted that the sparser the dataset, the further the SRM algorithm outperforms Apriori-Dual. This agrees with our intuition because, as the data is sparser, the overhead of negative itemset generation will become more severe, thus increasing the advantage of the technique on the induction of positive itemset supports achievable by the SRM algorithm.

4.3. Evaluation of rule quality

To evaluate the quality of a substitution rule, we may count the number of transactions that contain only one of the substitutive itemsets in the rule, i.e., the antecedent or the consequent. Hence, the violation ratio proposed in Aggarwal and Yu (1998) is adopted. An itemset I is said to be in violation of a transaction if some of the items are present in the transaction and others are not. Also, the violation of an itemset in a transaction is a bad event from the perspective of trying to establish a high correlation among the corresponding items. The fraction of violations of the itemset I over all transactions is denoted by $V(I)$. The violation ratio is defined as the ratio of the number of real violations to the expected number of violations and has the form

$$\text{violation ratio} = \frac{V(I)}{E(V(I))}.$$

Specifically, a pair of substitutive itemsets is said to be in violation if exactly only one of them is present in a transaction. Thus, the larger the value of the violation ratio of a rule, the more likely its antecedent and consequent itemsets are substitutes for each other. Note that the violation ratio of an interesting substitution rule should be larger than one.

The distribution curves for results of both sparse and dense datasets are depicted in Fig. 2. Because the numbers of rules generated by the Apriori-Dual and the SRM algorithms could be quite different in most cases, the percentage of population rather than the actual number of rules is used as the measurement for vertical axes in both charts. Note that, to provide a remarkable index for evaluating the quality of rules, proportions of interesting rules, i.e., for which violation ratios are larger than 1, uninteresting ones are also presented as pie charts in Fig. 2.

Fig. 2. Violation ratio distribution curves for results of both datasets

Note that more than half of the rules generated by the Apriori-Dual algorithm in both datasets have a violation ratio less than 1. For the dense dataset, nearly 80% of rules generated by the Apriori-Dual algorithm are uninteresting. In contrast, more than 98% of rules generated by the SRM algorithm are interesting for both datasets. Also note that the Apriori-Dual algorithm favours dense databases while the SRM algorithm performs well in each dataset, showing that the SRM algorithm is more adaptive and robust.

To provide further insights into the distribution of rules generated by the Apriori-Dual algorithm and the SRM algorithm, the scatter plots of results are presented in Figs. 3 and 4. In addition to the violation ratio as mentioned earlier, the correlation between the antecedent itemset and the consequent itemset of a substitution rule is taken as another measurement of interestingness.

Experiments on the two datasets are conducted with $\text{MinSup} = 15\%(D_{dense})$, $\text{MinSup} = 10\%(D_{sparse})$ and $\text{MinConf} = 50\%$. The resulting rules by Apriori-Dual and SRM for the dense dataset are plotted in Fig. 3 and those for the sparse dataset are shown in Fig. 4, where each point corresponds to a rule produced. In Figs. 3 and 4, the y-axis indicates the violation ratio and the x-axis shows the correlation of the antecedent and the consequent itemsets of the rule.

Note that targeting at binary attributes, i.e., the presence of itemsets, could have the absolute values of correlation coefficients smaller than those obtained by targeting at continuous attributes. This observation partly accounts for the reason that the correlation constraint in the SRM algorithm, i.e., $\rho_{\min}$, is not necessarily high.

Each figure is divided into four areas. In Area-I, the rules are the most interesting ones among those in all areas. The antecedent and the consequent of each substitution rule in Area-I are negatively correlated and both are good substitutes for each other due to high violation ratios. In Area-II, though the two itemsets of each rule are negatively correlated, low violation ratios reveal that the antecedent

Fig. 3. Quality matrix of Apriori-Dual and SRM in the dense dataset

Fig. 4. Quality matrix of Apriori-Dual and SRM in the sparse dataset

and the consequent may not be proper substitutes for each other. Note that the rules generated by the SRM algorithm are mostly in Area I. In contrast, more than half of rules generated by the Apriori-Dual algorithm are located in Area-III and Area-IV, whereas the antecedent and the consequent of rules are positively correlated, these itemsets are deemed inappropriate for substitution rules.

Note that rules generated by the Apriori-Dual algorithm and the SRM algorithm are subsets of negative association rules. Also, only such a rule with a high violation ratio and a negative correlation is appropriate to become a substitution rule. It can be seen from Figs. 3 and 4 that the SRM algorithm can generate the most appropriate ones on the basis of negative association rules. Consequently, rules generated by the SRM algorithm are deemed to possess much better quality than those by the Apriori-Dual algorithm.

5. Conclusions

In this paper, a new mining capability, called mining of substitution rules, is explored. The notion of evaluating the dependency among items in a concrete itemset proposed in this paper offers another dimension for itemset selection (in addition to the one of using the support threshold), thereby being able to lead to more interesting results in the subsequent rule derivation based on these itemsets. We have derived theoretical properties for the model of substitution rule mining and devised a technique on the induction of positive itemset supports to improve the efficiency of support counting for negative itemsets. In light of these properties, the SRM algorithm is proposed to

discover the substitution rules efficiently while attaining good statistical significance. It is shown by empirical studies that the SRM algorithm not only has very good execution efficiency but also produces substitution rules of very high quality.

Acknowledgements. The authors are supported in part by the National Science Council, Project No. NSC 91-2213-E-002-034 and NSC 91-2213-E-002-045, Taiwan, Republic of China.

Appendix: an example showing the chi-squared statistic is not upward closed

Consider a contingency table that is slightly modified from the one provided in Brin et al. (1997).

Table 6. An example contingency table of market basket data for coffee (c), tea (t) and doughnuts (d)

d	c	$\bar{c}$	Σrow
t	8	2	10
$\bar{t}$	40	2	42
Σcol	48	4	52

$\bar{d}$	c	$\bar{c}$	Σrow
t	10	2	12
$\bar{t}$	34	2	36
Σcol	44	4	48

From Theorem 1 in this paper, we have

$$\begin{aligned}
 Chi(\{c, t\}) &= \frac{(8+10)^2}{100 \times \frac{(48+44)}{100} \times \frac{(10+12)}{100}} + \frac{(40+34)^2}{100 \times \frac{(48+44)}{100} \times \frac{(42+36)}{100}} \\
 &\quad + \frac{(2+2)^2}{100 \times \frac{(4+4)}{100} \times \frac{(10+12)}{100}} + \frac{(2+2)^2}{100 \times \frac{(4+4)}{100} \times \frac{(42+36)}{100}} - 100 \\
 &= 3.98, \text{ and} \\
 Chi(\{d, c, t\}) &= \frac{8^2}{100 \times \frac{52}{100} \times \frac{(48+44)}{100} \times \frac{(10+12)}{100}} + \frac{40^2}{100 \times \frac{52}{100} \times \frac{(48+44)}{100} \times \frac{(42+36)}{100}} \\
 &\quad + \frac{2^2}{100 \times \frac{52}{100} \times \frac{(4+4)}{100} \times \frac{(10+12)}{100}} + \frac{2^2}{100 \times \frac{52}{100} \times \frac{(4+4)}{100} \times \frac{(42+36)}{100}} \\
 &\quad + \frac{10^2}{100 \times \frac{48}{100} \times \frac{(48+44)}{100} \times \frac{(10+12)}{100}} \\
 &\quad + \frac{34^2}{100 \times \frac{48}{100} \times \frac{(48+44)}{100} \times \frac{(42+36)}{100}} \\
 &\quad + \frac{2^2}{100 \times \frac{48}{100} \times \frac{(4+4)}{100} \times \frac{(10+12)}{100}} + \frac{2^2}{100 \times \frac{48}{100} \times \frac{(4+4)}{100} \times \frac{(42+36)}{100}} \\
 &\quad - 100 \\
 &= 4.49.
 \end{aligned}$$

As proven in Theorem 1, the corresponding degrees of freedom should increase with k , i.e., $df(\{c, t\}) = 1$ and $df(\{d, c, t\}) = 4$, respectively. Given a significance

level $\alpha = 0.05$, it can be verified that $Chi(\{c, t\}) = 3.98 > \chi_{1,0.05}^2 = 3.84$ and $Chi(\{d, c, t\}) = 4.49 < \chi_{4,0.05}^2 = 9.49$. Note that $\{c, t\}$ passed the chi-square test and $\{d, c, t\}$ did not, meaning that the chi-squared statistic is not upward closed.

References

- Aggarwal CC, Yu PS (1998) A new framework for itemset generation. In: Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems, Seattle, WA, June 1998, pp 18–24
- Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. In: Buneman P, Jajodia S (eds) Proceedings of the 1993 ACM SIGMOD international conference on management of data, Washington DC, May 1993, pp 207–216
- Agrawal R, Srikant R (1994) Fast algorithms for mining association rules in large databases. In: Bocca JB, Jarke M, Zaniolo C (eds) Proceedings of the 20th international conference on very large data bases, Santiago de Chile, Chile, September 1994, pp 478–499
- Ale JM, Rossi G (2000) An approach to discovering temporal association rules. In: Proceedings of the 2000 ACM symposium on applied computing, Villa Olmo, Como, Italy, March 2000, pp 294–300
- Ayad AM, El-Makky NM, Taha Y (2001) Incremental mining of constrained association rules. In: Proceedings of the 1st SIAM conference on data mining, Chicago, IL, April 2001
- Bayardo RJ, Agrawal R, Gunopulos D (1999) Constraint-based rule mining in large, dense databases. In: Proceedings of the 15th international conference on data engineering, Sydney, Australia, March 1999, pp 188–197
- Boulicaut J-F, Bykowski A, Jeudy B (2000) Towards the tractable discovery of association rules with negations. In: Larsen HL, Kacprzyk J, Zadrozny S et al (eds) Proceedings of the 4th international conference on flexible query answering systems, Warsaw, Poland, October 2000, pp 425–434
- Brin S, Motwani R, Silverstein C (1997) Beyond market baskets: generalizing association rules to correlations. In: Peckham J (ed) Proceedings of the 1997 ACM SIGMOD international conference on the management of data, Tucson, AZ, May 1997, pp 265–276
- Chen M-S, Han J, Yu PS (1996) Data mining: an overview from database perspective. *IEEE Trans Knowl Data Eng* 8(6):866–883
- Chen X, Petrounias I (2000) Discovering temporal association rules: algorithms, language and system. In: Proceedings of the 16th international conference on data engineering, San Diego, CA, February 2000, pp 306
- DuMouchel W, Pregibon D (2001) Empirical Bayes screening for multi-item associations. In: Proceedings of the 7th ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, August 2001, pp 67–76
- Han J, Fu Y (1995) Discovery of Multiple-level association rules from large databases. In: Dayal U, Gray PMD, Nishio S (eds) Proceedings of the 21st international conference on very large data bases, Zurich, Switzerland, September 1995, pp 420–431
- Han J, Kamber M (2000) Data mining: concepts and techniques. Morgan Kaufmann Publishers, San Francisco, CA
- Han J, Pei J, Yin Y (2000) Mining frequent patterns without candidate generation. In: Chen W, Naughton JF, Bernstein PA (eds) Proceedings of the 2000 ACM-SIGMOD international conference on management of data, Dallas, TX, May 2000, pp 1–12
- Hogg RV, Tanis EA (2000) Probability and statistical inference, 6/e. Prentice Hall, Upper Saddle River, NJ
- Hosseini JC, Harmon RR, Zwick M (1991) An information theoretic framework for exploratory multivariate market segmentation research. *Decision Sci* 22:663–677
- Jermaine C (2001) The computational complexity of high-dimensional correlation search. In: Cercone N, Lin T-Y, Wu X (eds) Proceedings of the 1st IEEE international conference on data Mining, San Jose, CA, November 2001, pp 249–256
- Johnson RA, Wichern DW (2002) Applied multivariate statistical analysis, 5/e. Prentice Hall, Upper Saddle River, NJ
- Lakshmanan LVS, Ng R, Han J et al (1999) Optimization of constrained frequent set queries with 2-variable constraints. In: Delis A, Faloutsos C, Ghandeharizadeh S (eds) Proceedings of the 1999 ACM SIGMOD international conference on management of data, Philadelphia, PA, June 1999, pp 157–168
- Lee C-H, Lin C-R, Chen M-S (2001) On mining general temporal association rules in a publication database. In: Cercone N, Lin T-Y, Wu X (eds) Proceedings of the 1st IEEE international conference on data mining, San Jose, CA, November 2001, pp 337–344

- Lee C-H, Lin C-R, Chen M-S (2001) Sliding-window filtering: an efficient algorithm for incremental mining. In: Proceedings of ACM 10th international conference on information and knowledge management, Atlanta, GA, November 2001, pp 263–270
- Liu B, Hsu W, Ma Y (1999) Mining association rules with multiple minimum supports. In: Proceedings of the 5th ACM SIGKDD international conference on knowledge discovery and data mining, San Diego, CA, August 1999, pp 337–341
- Liu B, Hsu W, Ma Y (2001) Identifying non-actionable association rules. In: Proceedings of the 7th ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, August 2001, pp 329–334
- Ma S, Hellerstein JL (2001) Mining mutually dependent patterns. In: Cercone N, Lin T-Y, Wu X (eds) Proceedings of the 1st IEEE international conference on data mining, San Jose, CA, November 2001, pp 409–416
- Mannila H, Rusakov D (2001) Decomposition of event sequences into independent components. In: Proceedings of the 1st SIAM conference on data mining, Chicago, IL, April 2001
- Meo R (2000) Theory of dependence values. *ACM Trans Database Syst* 25(3):380–406
- Park J-S, Chen M-S, Yu PS (1997) Using a hash-based method with transaction trimming for mining association rules. *IEEE Trans Knowl Data Eng* 9(5):813–825
- Pei J, Han J (2000) Can we push more constraints into frequent pattern mining? In: Proceedings of the 6th ACM SIGKDD international conference on knowledge discovery and data mining, Boston, MA, August 2000, pp 350–354
- Savasere A, Omiecinski E, Navathe S (1998) Mining for strong negative associations in a large database of customer transactions. In: Proceedings of the 14th international conference on data engineering, Orlando, FL, February 1998, pp 494–502
- Srikant R, Agrawal R (1995) Mining generalized association rules. In: Dayal U, Gray PMD, Nishio S (eds) Proceedings of the 21st international conference on very large data bases, Zurich, Switzerland, September 1995, pp 407–419
- Wang K, He Y, Han J (2000) Mining frequent itemsets using support constraints. In: Abbadi AE, Brodie ML, Chakravarthy S et al (eds) Proceedings of the 26th international conference on very large data bases, Cairo, Egypt, September 2000, pp 43–52

Author biographies

Wei-Guang Teng was born in Taipei, Taiwan, R.O.C., in 1976. He received his B.S. degree from the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C., in 1998. He is currently pursuing the Ph.D. degree in the same department. His research interests include data mining, multimedia networking and database.

Ming-Jyh Hsieh was born in I-Lan, Taiwan, R.O.C., in 1975. He received his B.S. degree from the Department of Physics, National Taiwan University, Taipei, Taiwan, R.O.C., in 1997. He is currently pursuing the Ph.D. degree in the Electrical Engineering Department, National Taiwan University, Taipei, Taiwan, R.O.C. His research interests include data mining, data warehousing and database.

Ming-Syan Chen received the B.S. degree from the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C., in 1982, and the M.S. and Ph.D. degrees from the EECS Department, the University of Michigan, Ann Arbor, Michigan, in 1985 and 1988, respectively. From 1988 to 1996, he was with IBM Thomas J. Watson Research Center, Yorktown Heights, New York, involved in projects on parallel databases, multimedia systems, data mining and Internet applications. Dr. Chen is an editor of IEEE Transactions on Knowledge and Data Engineering and also the Journal of Information Science and Engineering (JISE), was a Distinguished Visitor in IEEE Computer Society for the Asia-Pacific region from September 1998, and also the program chair of the 6th Pacific Area Conference on Knowledge Discovery and Data Mining (PAKDD-02) in May 2002. He is a recipient of the NSC Distinguished Research Award. He coauthored, with his students, their

works, which received ACM SIGMOD Research Student Award and the Long-Term Thesis Award. He is leading the Network Database Lab in his department. Dr. Chen is a fellow of the IEEE, and a member of the ACM.

Correspondence and offprint requests to: Ming-Syan Chen, Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. Email: mschen@cc.ee.ntu.edu.tw