Overview

Fundamentals of association rules in data mining and knowledge discovery

Shichao Zhang[1,2*] and Xindong Wu[3]

Association rule mining is one of the fundamental research topics in data mining and knowledge discovery that identifies interesting relationships between itemsets in datasets and predicts the associative and correlative behaviors for new data. Rooted in market basket analysis, there are a great number of techniques developed for association rule mining. They include frequent pattern discovery, interestingness, complex associations, and multiple data source mining. This paper introduces the up-to-date prevailing association rule mining methods and advocates the mining of complete association rules, including both positive and negative association rules. *WIREs Data Mining Knowl Discov* 2011 1 97–116 DOI: 10.1002/widm.10

INTRODUCTION

Knowledge Discovery in Databases (KDD) was initially named by Gregory Piatetsky-Shapiro in a workshop at the 1989 International Joint Conference on Artificial Intelligence in Detroit, USA, in August 1989. It was defined as the process of finding interesting, interpretable, useful, and novel knowledge. KDD has become a multidisciplinary subject today. The original KDD workshop series became an annual international conference in 1995, and a biannual academic magazine titled '*Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) Explorations*' was launched in 1999. The term of KDD was successively defined at two different times. In 1992, it was defined as a nontrivial extraction of implicit, previously unknown, and potentially useful information from data by Frawley et al.[1] In 1996, Fayyad et al.[2] defined KDD as a nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. The 1996 definition has been widely accepted by both academia and industry.

During the development of KDD, association rule mining has played a fundamental role, which was first proposed by Agrawal et al.[3] for market basket analysis. It has since emerged as a prominent research area in KDD. This is attributed to its simple representation, easy understanding, and being potentially useful in capturing, for example, customer buying behavior.

An association rule is of the form $X \rightarrow Y$. It is interesting if its support [$supp\ (X \cup Y)$] and confidence [$conf\ (X \rightarrow Y)$] are equal to or greater than the user-specified minimum support (ms) and minimum confidence (mc) thresholds, respectively. This simple representation is generally referred to as the support–confidence ($supp$–$conf$) framework. An association rule such as 'if milk then bread' means that if a customer purchases milk, she/he likely also buys bread. This explanation is easy to understand. In particular, the story concerning 'diapers → beer' has shown that association rules are useful, hidden, and difficult to mine. This explanation has made a big boost in association rule mining development. Thereby, it is not surprising that association rule mining almost meant KDD from 1993 to 2002 in many people's eyes.

Although an association rule looks really simple; in real applications, association rule mining is often challenging. The first challenge is the object to be mined - data sources. Data sources are often very large, multiple, and heterogeneous. Also, the data may be raw, rough, incomplete, with missing values, and dynamic. The second challenge is the mining method itself. It includes an interestingness measure, an exponential search space, a mining theory,

*Correspondence to: zhangsc@zjnu.cn

[1]Department of Computer Science, Zhejiang Normal University, Jinhua, Zhejiang, PR China

[2]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu, PR China

[3]Department of Computer Science, University of Vermont, Burlington, VT, USA

DOI: 10.1002/widm.10

and an evaluation criterion. The last challenge comes with the constraints such as accuracy requirements and time and space limits.

As an introduction to association analysis, this article focuses on the basic concepts, typical techniques, and some applications. The article is organized as follows: The section '*Association Rules*' introduces the basic concepts and the Apriori algorithm concerning association rule discovery. The section '*Complete Association Rule Analysis: Mining both Positive and Negative Rules*' describes a complete association analysis that identifies both positive and negative association rules of interest. In section '*Applications of Association Rules*', we list some of the main applications of association rules. Conclusion remarks are given in the fifth section.

ASSOCIATION RULES

This section introduces some representative work on association rule mining, including the *supp–conf* framework, the Apriori algorithm, and some research directions of association rule mining.

The Support–Confidence Framework

Let $I = \{i_1, i_2, \ldots, i_N\}$ be a set of literals or items. For example, milk, sugar, and bread for purchase in a store are items. Assume D is a set of transactions over I, called the transaction database, in which a transaction is a set of items, that is, a subset of I. A transaction has an associated unique identifier called Transaction IDentifier (TID).

A set of items is referred to 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 the itemset. Itemsets of some length k are referred to as k-itemsets.

Each itemset has an associated statistical measure called support, denoted as *supp* or p. The *supp* is either the proportion of transactions in the database that contains the itemset or the number of transactions that contain the itemset. Formally, for an itemset $X \subseteq I$, $p(X)$ is defined as the fraction of transactions in D containing X or

$$p(X) = \frac{1}{n} \sum_{i=1}^{n} 1(X \subseteq D_i)$$

where the database D is viewed as a vector of n records (or transactions) D_i such that each record is a set of items.

An itemset X in a transaction database D is called a large (or frequent) itemset if its support $p(X)$ is equal to, or greater than, a threshold of minimal support (*minsup*, *ms*), which is given by users or experts.

An association rule is an implication $X \rightarrow Y$ that describes the existence of a relationship between itemsets X and Y, where X, $Y \subset I$, and $X \cap Y = \Phi$ (itemsets X and Y must not intersect). Each association rule has two quality measurements: support and confidence. The confidence, denoted as *conf*, is the ratio of the number of transactions that include all items in the consequent as well as the antecedent (namely, the support) to the number of transactions that include all items in the antecedent. For $X \rightarrow Y$, they are defined as follows:

(1) The *supp* of $X \rightarrow Y$ is the *supp* of $XY/(X \cup Y)$, where XY means both itemsets X and Y occur at the same time, that is, *supp* = the frequency of occurring patterns or $p(XY)$.
(2) The *conf* of $X \rightarrow Y$ is the ratio $p(XY)/p(X)$, that is, *conf* = the strength of implication.

Association rules provide information of this type in the form of 'if–then' statements. These rules are computed from the data and, unlike the if–then rules in logic, association rules are probabilistic in nature. In addition to the antecedent (the 'if' part) and the consequent (the 'then' part), an association rule has the *supp* and *conf* measurements that express the degree of uncertainty about the rule. In association analysis, the antecedent and consequent are sets of items (called itemsets) that are disjoint (without any items in common).

Association rule mining seeks interesting associations and/or correlation relationships between frequent itemsets in datasets. Association rules show attribute-value conditions that occur frequently together in a given dataset. A typical and widely-used example of association rule mining is market basket analysis. The first effort on mining association rules is based on a *supp–conf* framework as follows.

The supp–conf framework (Agrawal et al.[3]). Let I be the set of items in database D, $X, Y \subseteq I$ be itemsets, $X \cap Y = \Phi$, $p(X) \neq 0$, and $p(Y) \neq 0$. Assume the minimal support (*minsup*, or *ms*) and minimal confidence (*minconf*, or *mc*) are given by users or experts. Then $X \rightarrow Y$ is a valid association rule if $p(XY) \geq ms$ and $conf(X \rightarrow Y) \geq mc$.

Accordingly, association rule mining can be broken down into two subproblems as follows.

(1) Generating all itemsets that have a *supp* greater than, or equal to, the user specified*ms*. That is, identifying all frequent itemsets.

(2) Generating all rules that have the *mc* in the following way: for a frequent itemset Z, any $X \subset Z$, and $Y = Z - X$, if the *conf* of a rule $X \rightarrow Y$ is greater than, or equal to, the *mc* (or $p(Z)/p(X) \geq mc$, then it can be extracted as a valid rule.

With the above decomposed subproblems, the *supp–conf* framework is a simple and easy-to-understand two-step process. The first step is to search for frequent itemsets and the second step generates association rules.

The Apriori Algorithm

The complexity of an association rule mining system is heavily dependent upon the identification of frequent itemsets. The most prevailing algorithm to perform this identification is the Apriori algorithm.

Agrawal and Srikant[4] observed an interesting downward closure property, called Apriori, among frequent k-itemsets: a k-itemset is frequent only if all of its subitemsets are frequent. Accordingly, the frequent 1-itemsets are searched in the first scan of the database, then the frequent 1-itemsets are used to generate candidate frequent 2-itemsets, and check against the database to obtain the frequent 2-itemsets. Generally, the frequent (k-1)-itemsets are used to generate candidate frequent k-itemsets, and check against the database to obtain the frequent k-itemsets. This process iterates until no more frequent k-itemsets can be generated for some k. This is the essence of the Apriori algorithm.[4] It is described as follows:

Algorithm 1. Apriori
Input: D: a database; *ms*: minimum support;
Output: F: a set of frequent itemsets of interest;

(1) **let** $F \leftarrow \{\}$;
(2) **let** $L1 \leftarrow \{\text{frequent 1-itemsets}\}$; $F \leftarrow F \cup L_1$;
(3) **for** $(k = 2; (L_{k\text{-}1} \neq \{\}); k^{++})$ **do begin**
//Generate all possible frequent k-itemsets of interest in D.
(3.1) let $\text{Tem}_k \leftarrow \{\{x_1, \ldots, x_{k\text{-}2}, x_{k\text{-}1}, x_k\} | \{x_1, \ldots, x_{k\text{-}2}, x_{k\text{-}1}\} \in L_{k\text{-}1} \wedge \{x_1, \ldots, x_{k\text{-}2}, x_k\} \in L_{k\text{-}1}\}$;
(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 for
(3.3) let $L_k \leftarrow \{c | c \in \text{Tem}_k \wedge (p(c) = (c.count/|D|) \geq ms)\}$;
(3.4) let $F \leftarrow F \cup L_k$;
end (3)
(4) output F;
(5) return.

TABLE 1 | A Transaction Database

TID	Item A	Item B	Item C	Item D	Item E
100	A		C	D	
200		B	C		E
300	A	B	C		E
400		B			E

The Apriori algorithm generates all frequent itemsets in a given database D. The initialization is done in Step (1). Step (2) generates $L1$ of all frequent 1-itemsets in D in the first pass of D.

Step (3) generates L_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, and the end condition of the loop is $L_{k\text{-}1} = \{\}$. For each pass of the database in Step (3), say pass k, there are four 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 $L_{k\text{-}1}$. Each itemset in Tem_k is counted in D by a loop in Step (3.2). Then L_k is generated in Step (3.3), which is the set of all potentially useful frequent k-itemsets in Tem_k, where all frequent k-itemsets in L_k meet *ms*. Finally, L_k is added to F in (3.4).

Step (4) outputs the frequent itemsets of potential interest in F. The procedure ends in Step (5).

An Example

Let $I = \{A, B, C, D, E\}$ and the transaction universe be $TID = \{100, 200, 300, 400\}$[a].

In Table 1, 100, 200, 300, and 400 are the unique identifiers of the four transactions: A = sugar, B = bread, C = coffee, D = milk, and E = cake.

Each row in the table can be taken as a transaction. We can identify frequent itemsets (the first step of the *supp–conf* framework) from these transactions using the Apriori algorithm (see, '*How Apriori Works*'), and association rules from the frequent itemsets (the second step of the *supp–conf* framework) using the *supp–conf* framework in '*Generating Association Rules*' (see below). Let

(1) $ms = 50\%$ (to be frequent, an itemset must occur in at least two transactions); and

TABLE 2 | Frequent 1-Itemsets and Their Frequencies in Table 1

Itemset	Frequency	$\geq ms$
{A}	2	Y
{B}	3	Y
{C}	3	Y
{E}	3	Y

TABLE 3 | Frequent 2-Itemsets and Their Frequencies in Table 1

Itemset	Frequency	$\geq ms$
{A,C}	2	Y
{B,C}	2	Y
{B,E}	3	Y
{C,E}	2	Y

TABLE 4 | Frequent 3-Itemsets and Their Frequencies in Table 1

Itemsets	Frequency	$\geq ms$
{B,C,E}	2	Y

(2) $mc = 60\%$ [to be a high-*conf* (or valid) rule, at least 60% of the time you find the antecedent of the rule in the transactions, you must also find the consequence of the rule there].

How Apriori Works

To illustrate the use of the Apriori algorithm (the first step of the *supp–conf* framework), we outline an example of the process of identifying frequent itemsets in the dataset in Table 1.

In the Apriori algorithm, in Step (1) F is assigned an empty set. Step (2) scans the database to generate frequent 1-itemsets. The 1-itemsets $\{A\}$, $\{B\}$, $\{C\}$, $\{D\}$, and $\{E\}$ are first generated as candidates at the first pass over the dataset, and A.count = 2, B.count = 3, C.count = 3, D.count = 1, and E.count = 3, where X.count = x means that the frequency of itemset X is x. Because $ms = 50\%$ and $dbsize = 4$, $\{A\}$, $\{B\}$, $\{C\}$, and $\{E\}$ are frequent itemsets, or $L_1 = \{A, B, C, E\}$. The frequent 1-itemsets and their frequencies are listed in Table 2.

L_1 is then added to F and $F = \{A, B, C, E\}$.

Step (3) is a loop. In the first iteration, candidate frequent 2-itemsets are first generated by the frequent 1-itemsets of L_1 in Step (3.1). That is, $Tem_2 = \{AB, AC, AE, BC, BE, CE\}$. And then the second pass begins over the dataset to examine the candidate 2-itemsets of Tem_2 in Steps (3.2) and (3.3). From Table 1, AB.count = 1, AC.count = 2, AE.count = 1, BC.count = 2, BE.count = 3, and CE.count = 2. Therefore, AC, BC, BE, and CE are frequent itemsets, or $L_2 = \{AC, BC, BE, CE\}$. The frequent 2-itemsets and their frequencies are listed in Table 3.

L_2 is then added to F and $F = \{A, B, C, E, AC, BC, BE, CE\}$.

Because $L_2 \neq \{\}$, Step (3) is repeated. In the second iteration, candidate frequent 3-itemsets are first generated by the frequent 2-itemsets of L_2 in Step (3.1). That is, $Tem_3 = \{BCE\}$[b]. And then the third pass begins over the dataset to examine the candidate 3-itemsets of Tem_3 in Steps (3.2) and (3.3). From Table 1, BCE.count = 2. Therefore, BCE is a frequent itemset or $L_3 = \{BCE\}$. The frequent 3-itemsets and their frequencies are listed in Table 4.

L_3 is then added to F and $F = \{$A, B, C, E, AC, BC, BE, CE, BCE$\}$.

Because $L_3 \neq \{\}$, Step (3) is repeated. In the third iteration, candidate frequent 4-itemsets are first generated by the frequent 3-itemsets of L_3 in Step (3.1). That is, $Tem_4 = \{\}$. Because $Tem_4 = \{\}$, $L_4 = \{\}$ and Step (3) is finished.

In Step (4), the frequent itemsets in F ($F = \{A, B, C, E, AC, BC, BE, CE, BCE\}$) are output and the algorithm is ended in Step (5).

Generating Association Rules

To show how to generate association rules from a given database (the second step of the *supp–conf* framework), we use the above frequent itemsets identified from the dataset in Table 1.

For simplifying the description, we detail how to generate association rules from the frequent itemset BCE in F, with $p(BCE) = 50\% = ms$.

Because $p(BCE)/p(BC) = 2/2 = 100\%$, which is greater than $mc = 60\%$, $BC \rightarrow E$ can be extracted as a valid rule. In the same way, because $p(BCE)/p(BE) = 2/3 = 66.7\%$, which is greater than mc, $BE \rightarrow C$ can be extracted as another valid rule; and because $p(BCE)/p(CE) = 2/2 = 100\%$ is greater than mc, $CE \rightarrow B$ can be extracted as a third valid rule. The association rules with 1-item consequences generated from BCE are listed in Table 5.

Also, because $p(BCE)/p(BE) = 2/3 = 66.7\%$ is greater than mc, $B \rightarrow CE$ can be extracted as a valid rule. In the same way, because $p(BCE)/p(C) = 2/3 = 66.7\%$ is greater than mc, $C \rightarrow BE$ can be extracted as a valid rule; and $p(BCE)/p(E) = 2/3 = 66.7\%$ is greater than mc, $E \rightarrow BC$ can be extracted as a valid rule. The association rules with

TABLE 5 | Association Rules with 1-Item Consequences from 3-Itemsets

Rule Number	Rule	Confidence (%)	Support (%)	$\geq mc$
Rule 1	$BC \rightarrow E$	100	50	Y
Rule 2	$BE \rightarrow C$	66.7	50	Y
Rule 3	$CE \rightarrow B$	100	50	Y

TABLE 6 | Association Rules with 2-Item Consequences from 3-Itemsets

Rule Number	Rule	Confidence (%)	Support (%)	$\geq mc$
Rule 4	$B \rightarrow CE$	66.7	50	Y
Rule 5	$C \rightarrow BE$	66.7	50	Y
Rule 6	$E \rightarrow BC$	66.7	50	Y

TABLE 7 | Association Rules from 2-Itemsets

Rule Number	Rule	Confidence (%)	Support (%)	$\geq mc$
Rule 7	$A \rightarrow C$	100	50	Y
Rule 8	$C \rightarrow A$	66.7	50	Y
Rule 9	$B \rightarrow C$	66.7	50	Y
Rule 10	$C \rightarrow B$	66.7	50	Y
Rule 11	$B \rightarrow E$	100	75	Y
Rule 12	$E \rightarrow B$	100	75	Y
Rule 13	$C \rightarrow E$	66.7	50	Y
Rule 14	$E \rightarrow C$	66.7	50	Y

2-item consequences generated from BCE are listed in Table 6.

For all frequent 2-itemsets in F, we can also generate all association rules as illustrated in Table 7.

According to the above analysis, the 14 association rules listed above can be extracted as valid rules for Table 1.

Research Directions in Association Rule Mining

As we have seen previously, association rules are useful in real world applications and have played a fundamental role in the development of data mining. When a given dataset is very large, association rule mining is challenging. This is because the Apriori algorithm used for identifying frequent itemsets involves a search with little heuristics about a space with an exponential amount of items and possible itemsets. This algorithm may suffer from large computational overhead when the number of frequent itemsets is very large. For example, suppose there are 1000 items in a given large database, the average number of items in each transaction is six. Then there are almost 10^{15} possible itemsets to be counted in the database. These have led to some research directions as follows:

Frequent pattern (itemset) mining, such as the FP-growth method[5] for frequent pattern (itemset) mining, frequent closed patterns,[6] and sampling techniques[7] for algorithm scale-up, has attracted much attention in data mining. Well-known mining methods include, for example, data structures for association rule mining,[8] hashing techniques,[9] partitioning,[10,11] sampling,[12] anytime mining,[13] parallel and distributed mining,[9,14–16] and integrating mining with relational database systems.[17]

Interestingness measures the strength of the relationship between itemsets X and Y. The prevailing measures are as follows:

(1) Interest factor[18,19]

$$\mathrm{I}(X \rightarrow Y) = \mathrm{I}(X, Y) = \frac{p(XY)}{p(X)p(Y)}$$

It is a nonnegative real number, with value 1 corresponding to the statistical independence.

(2) Pearson's correlation coefficient

$$\varphi(X \rightarrow Y) = \varphi(X, Y) = \frac{p(XY) - p(X)p(Y)}{\sqrt{p(X)p(Y)(1 - p(X))(1 - p(Y))}}$$

It ranges between -1 and 1. If X and Y are independent, then $\varphi(X \rightarrow Y) = 0$.

(3) Cosine similarity

$$\cos ine(X \rightarrow Y) = \cos ine(X, Y) = \frac{p(XY)}{\sqrt{p(X)p(Y)}} = \sqrt{conf(X \rightarrow Y)conf(Y \rightarrow X)}$$

(4) Odds ratio

$$\alpha(X \rightarrow Y) = \alpha(X; Y) = \log \frac{p(XY)p(\neg X \neg Y)}{p(X \neg Y)p(\neg XY)}$$

(5) Rule interest[20]

$$\mathrm{R}(X \rightarrow Y) = \mathrm{R}(X, Y) = |p(XY) - p(X)p(Y)|$$

(6) Conviction[18]

$$conviction(X \rightarrow Y) = conviction(X, Y) = \frac{p(X)p(\neg Y)}{p(X \neg Y)}$$

(7) Certainty factor[21]

$$CF(X \rightarrow Y) = CF(X, Y) = \frac{p(Y|X) - p(Y)}{1 - p(Y)} = \frac{p(XY) - p(X)p(Y)}{p(X)(1 - p(Y))}$$

Note that, if $p(Y) > p(Y|X)$, $CF(X, Y)$ is defined as

$$CF(X, Y) = \frac{p(Y|X) - p(Y)}{-p(Y)} = \frac{p(XY) - p(X)p(Y)}{-p(X)p(Y)}$$

(8) Laplace measure

$$laplace(X \rightarrow Y) = laplace(X, Y) = \frac{p(XY) + 1}{p(X) + 2}$$

Obviously, it is very similar to the confidence.

(9) J measure[22]

$$J(X \rightarrow Y) = J(X, Y) = p(X)\left[p(Y|X)\log\frac{p(Y|X)}{p(Y)} + (1 - p(X|Y))\log\frac{1 - p(Y|X)}{1 - p(Y)}\right]$$

Discovery of complex association rules, for example, quantitative association rules,[23] causal rules,[24] and multilevel and multidimensional association rules.[25–27]

Quantitative association rule mining is designed for analyzing quantitative data that are over categorical attributes. An item over a categorical attribute can be expressed as either an interval (a continuous set of attribute values) or a single value, called a quantitative item. For example, 'Salary $\in$ [50k, 70k]' is a quantitative item. If X is a quantitative item, X can be valued in a certain interval F. The *supp* of X is the sum of the *supps* of all values in F. A quantitative association rule is a relationship between X and Y of the form $X \rightarrow Y$, where X and Y are quantitative items. Quantitative association rule mining is based on the *supp–conf* framework, so do causal rule mining and multilevel and multidimensional association rule mining.

Automation of Mining Association Rules

Apriori-based mining algorithms are based on the assumption that users can specify the minimum support for their databases. That is, a frequent itemset (or an association rule) is interesting if its *supp* is larger than or equal to the minimum support. This creates a challenging issue: performances of these algorithms heavily depend on some user-specified thresholds. For example, if the minimum-support value is too big, nothing might be found in a database, whereas a small minimum support might lead to poor mining performance and many generations of uninteresting association rules. Therefore, users are unreasonably required to know details of the database to be mined to specify a suitable threshold. However, Han et al.[6] have pointed out that setting the minimum support is quite subtle, which can hinder the widespread applications of these algorithms; our own experience of mining transaction databases also tells us that the setting is by no means an easy task. In particular, even though a minimum support is explored under the supervision of an experienced miner, we cannot examine whether or not the results (mined with the hunted minimum support) are just what users want. This means that the minimum-support setting is a key issue in automatic association rule mining.

Current techniques for addressing the minimum-support issue are as follows: In proposals for marketing, Piatetsky-Shapiro and Steingold[28] proposed to identify only the top 10% or 20% of the prospects with the highest score. Han, et al.[6,25] designed a strategy to mine top-k frequent patterns for effectiveness and efficiency. In proposals for interesting itemset discovery, Cohen, et al.[29] developed a family of effective algorithms for finding interesting associations. In proposals for dealing with temporal data, Roddick and Rice[30] discussed independent thresholds and context-dependent thresholds for measuring time-varying interestingness of events. In proposals for exploring new strategies, Hipp and Guntzer[31] presented a new mining approach that postpones constraints from mining to evaluation. In proposals for identifying new patterns, Wang, et al.[32,33] designed a *conf*-driven mining strategy without minimum support. However, these approaches only attempt to avoid specifying the minimum support.

Different from traditional association rule mining methods, database-independent mining techniques have been developed,[34,35] in which users can specify a threshold of *supp* for a mining task without being required to know any of the database. This has provided a way of developing automatic association rule mining systems.

Association Analysis for Different Data Sources

For example, data sources may be multiple, heterogeneous, incomplete, and dynamic. Well-known mining methods include local pattern analysis,[36,37] selecting relevant databases toward multidatabase mining,[38] peculiarity discovery,[39] local mining for finding sequential patterns,[40] bridging the local and global analysis for noise cleansing,[41] classification from multiple sources,[42] distributed data mining,[43] and so on.

COMPLETE ASSOCIATION RULE ANALYSIS: MINING BOTH POSITIVE AND NEGATIVE RULES

Traditional association rule mining techniques are only designed to find the strong patterns that have high predictive accuracy or correlation, called positive association analysis. This mining sufficiently utilizes frequent itemsets for real data analysis applications. Wu et al.[21,44] studied negative association rule mining that utilizes infrequent itemsets as well in datasets. A negative association rule is an implication of the form $X \rightarrow \neg Y$ (or $\neg X \rightarrow Y$ or $\neg X \rightarrow \neg Y$). The rule $X \rightarrow \neg Y$ enables us to predict that Y is unlikely to occur when X occurs. In other words, the negative association rules do catch mutually exclusive correlations among items. We call negative association rule mining as negative association analysis and mining both positive and negative association rules as complete association rule (CAR) analysis.

This section first recalls what is a significant negative association rule in '*Negative Association Rules*'. A framework for complete association analysis is given in '*A Framework for Complete Association Analysis*'. Finally, some research directions are outlined in '*Research Directions in CAR Analysis*'.

Negative Association Rules

Although 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, 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 has gained an insight into the importance of negative association rule mining. On the contrary, the development of negative association rule mining will allow companies to hunt more business chances through using infrequent itemsets of interest than those which only take into account frequent itemsets. For capturing these significant infrequent itemsets, this subsection introduces some basic concepts concerning negative association rule mining.

An infrequent itemset is an itemset that does not meet the user-specified*ms*, whereas a frequent itemset is an itemset that meets the user-specified*ms*.

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

We call a rule of the form $X \rightarrow Y$ a positive rule, and rules of the other forms negative rules. For convenience, we often use only the form $X \rightarrow \neg Y$ to represent and describe negative association rules in this chapter.

Like positive rules, a negative rule $X \rightarrow \neg Y$ has also a measure of its strength, *conf*, defined as the ratio $p(X \neg Y)/p(X)$.

By extending the definition given in Ref 3, negative association rule discovery seeks rules of the form $X \rightarrow \neg Y$ with *supp* and *conf* greater than, or equal to, user-specified *ms* and minimum confidence thresholds, respectively, where

(1) X and Y are disjoint itemsets, that is, $XY = \Phi$;
(2) $p(X) \geq ms$, $p(Y) \geq ms$, and $p(XY) < ms$;
(3) $p(X \rightarrow \neg Y) = p(X \neg Y)$;
(4) $conf(X \rightarrow \neg Y) = p(X \neg Y)/p(X) \geq mc$.

In this article, a significant negative association rule $X \rightarrow \neg Y$ means that it satisfies the abovementioned four conditions. Accordingly, the infrequent itemset XY is statistically regarded as a significant itemset, or significant infrequent itemset. Below is an example of a negative association rule.

Example 1[c]. Suppose we have a market basket database from a grocery store, consisting of n baskets. Let us focus on the purchases of tea (denoted by t) and coffee (denoted by c).

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

Now consider $p(c) = 0.6$, $p(t) = 0.4$, $p(tc) = 0.05$, and $mc = 0.52$. The *conf* of $t \rightarrow c$ is $p(tc)/p(t) = 0.05/0.4 = 0.125 < mc = 0.52$, and $p(tc) = 0.05$ is low. This indicates that tc is an infrequent itemset and that $t \rightarrow c$ cannot be extracted as a rule in the *supp–conf* framework. However, $p(t \neg c) = p(t) - p(tc) = 0.4–0.05 = 0.35$ is high, and the *conf* of $t \rightarrow \neg c$ is the ratio

$p(t \neg c)/p(t) = 0.35/0.4 = 0.875 > mc$. Therefore $t \rightarrow \neg c$ is a valid rule from the database.

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.

Consider a transaction database $(TD) = \{(A, B, D); (B, C, D); (B, D); (B, C, D, E); (A, B, D, F)\}$, which has five transactions, separated by semicolons[d]. Each transaction contains several items, separated by commas. There are at least 818 possible negative association rules generated from the 49 infrequent itemsets in TD when minsupp = 0.4. This means there are essential differences between negative association rule mining and positive association rule mining.

Because negative association rules are hidden in infrequent itemsets (with lower frequency), traditional pruning techniques are inefficient for identifying infrequent itemsets of interest.[21] This means, we must exploit alternative strategies to (1) confront an exponential search space consisting of all possible itemsets, frequent and infrequent in a database; (2) detect which of the infrequent itemsets can generate negative association rules; (3) perceive which of the negative association rules are really useful to applications; and (4) measure the interestingness of both positive and negative association rules. These problems are very different from those being faced by discovering positive association rules. And it is rather difficult to identify negative association rules of interest in databases.

In this subsection, we have not introduced algorithms for identifying infrequent itemsets and negative association rules of interest. They will be presented in next subsection.

A Framework for Complete Association Analysis

As we have mentioned above, 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 efficiently discover complete associations of interest. Therefore, in this subsection, we first design a pruning strategy and the mining framework, and then a procedure for identifying frequent and infrequent itemsets of interest, and finally the algorithm of generating complete associations of interest.

A Pruning Strategy[e] and a Complete Association Mining Framework

According to the interest factor,[18,19] we use an interestingness function $R(X, Y) = |p(XY) - p(X)p(Y)|$[20] and a threshold minimum interestingness (mi). It means that a rule $X \rightarrow Y$ is of potential interest when $R(X, Y) \geq mi$, and XY is referred to as a potentially interesting itemset. Including this $R(X, Y)$ mechanism and the CF model,[21] we can formally define the function that Z is a frequent itemset of potential interest as follows:

$$fipi(Z) = (p(Z) \geq ms) \wedge (\exists X, Y \subset Z \wedge Z = XY) \wedge fipis(X, Y)$$

$$fipis(X, Y) = (X \cap Y = F) \wedge g(X, Y, mc, mi) = 1$$

where $g(X, Y, mc, mi) = s(X, Y) \vee s(Y, X)$, and

$$s(X, Y) = \begin{cases} 1, & if\ R(X, Y) \geq mi \wedge CF(X, Y) \geq mc \\ 0, & otherwise \end{cases}$$

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

If X is a frequent itemset and Y is an infrequent itemset with frequency 1 in a large database, then $X \rightarrow \neg Y$ certainly looks like a valid negative rule because $p(X) \geq ms$, $p(Y) \approx 0$, $p(X \neg Y) \approx p(X) \geq ms$, $conf(X \rightarrow \neg Y) = p(X \neg Y)/p(X) \approx 1 \geq mc$. This could indicate that the rule $X \rightarrow \neg Y$ 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 databases would mostly involve frequent itemsets only. This means that if $X \rightarrow \neg Y$ (or $\neg X \rightarrow Y$, or $\neg X \rightarrow \neg Y$) is a negative rule of interest, X and Y would be frequent itemsets. In other words, no matter whether association rules are positive or negative, we are only interested in relationships among frequent itemsets. To operationalize this insight, we can use the support measure p. If $p(X) \geq ms$ and $p(Y) \geq ms$, the rule $X \rightarrow \neg Y$ is of potential interest, and XY is referred to as a potentially interesting itemset.

Including the above insight, the $R(X, Y)$ mechanism and the CF model, Z is an infrequent itemset of

potential interest as follows:

$$iipi(Z) = (\exists X, Y \subset Z \wedge Z = XY) \wedge (p(XY) < ms) \wedge iipis(X, Y)$$

$$iipis(X, Y) = (p(X) \geq ms) \wedge (p(Y) \geq ms) \wedge (X \cap Y = F) \wedge h(X, Y, ms, mc, mi) = 1$$

where $h(X, Y, ms, mc, mi) = t(X, \neg Y) \vee t(\neg X, Y) \vee t(\neg X, \neg Y) \vee t(Y, \neg X) \vee t(\neg Y, X)$, and

$$t(X, Y) = \begin{cases} 1, & if\ R(X, Y) \geq mi \wedge CF(X, Y) \geq mc \wedge g(X, Y, mc, mi) \\ 0, & otherwise \end{cases}.$$

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 frequent itemset of potential interest (fipi) and infrequent itemset of potential interest (iipi) mechanisms for both positive and negative rule discovery, search is constrained to seek interesting rules on certain measures, and pruning is the removal of uninteresting branches that cannot lead to an interesting rule that satisfies those constraints.

On the basis of the measures *fipis* and *iipis*, a framework for identifying complete associations of interest is defined as follows:

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

It means that mining complete associations (both positive and negative association rules) of interest can be decomposed into the above two subproblems. We carry out these two in '*Searching For Frequent and Infrequent Itemsets of Interest*' and '*Identifying Complete Associations of Interest*', respectively. And their use will be illustrated with a dataset in '*Complete Association Analysis: An Illustration*'.

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.

Algorithm 2. All Itemsets Of Interest

Input: D: a database; *ms*: minimum support; *mc*: minimum conference; *mi*: minimum interest;

Output: PL: a set of frequent itemsets of interest;

NL: a set of infrequent itemsets of interest;

(1) //Apriori Algorithm
let $PL \leftarrow \{\}$;
(2) let $L_1 \leftarrow \{frequent1\text{-}itemsets\}$; $PL \leftarrow PL \cup L1$;
(3) for ($k = 2$; (Lk-1 $\neq \{\}$); k^{++}) do begin
//Generate all possible frequent k-itemsets of interest in D.
(3.1) let $Tem_k\{\{x_1, \ldots, x_{k\text{-}2}, x_{k\text{-}1}, x_k\} | \{x_1, \ldots, x_{k\text{-}2}, x_{k\text{-}1}\} \in L_{k\text{-}1} \wedge \{x_1, \ldots, x_{k\text{-}2}, x_k\} \in L_{k\text{-}1}\}$;
(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 for
(3.3) let $L_k \leftarrow \{c | c \in Tem_k \wedge (p(c) = (c.count/|D|) \geq ms)\}$;
(3.4) **let** $PL \leftarrow PL \cup L_k$;
end (3)
(4) let NL $\leftarrow \{\}$;
//Generate all possible infrequent k-itemsets of interest in D.
for any X and Y in $PL \wedge X$ÇY $= \Phi$ **do**
if $XY \notin PL$ **then**
let $NL \leftarrow NL \cup \{(X, Y)\}^f$
(5) //Prune all uninterested itemsets in PL
for each itemset Z in PL **do**
if NOT(*fipi*(Z)) **then**
let $PL \leftarrow PL$-{Z};
(6) //Prune all uninterested itemsets in NL
for each itemset (X, Y) in NL **do**
if NOT(*iipi*(XY)) **then**
let $NL \leftarrow NL - \{(X, Y)\}$;

(7) output PL and NL;

(8) return.

The procedure All Itemsets Of Interest 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 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, and the end condition of the loop is $L_{k\text{-}1} = \{\}$. For each pass of the database in Step (3), say, pass k, there are four 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 $L_{k\text{-}1}$. Each itemset in Tem_k is counted in D by a loop in Step (3.2). Then L_k is 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. Finally, L_k is added to PL in Step (3.4).

Step (4) generates the NL, that is, the set of all infrequent itemsets, whose supports do not meet ms. And NL is the set of all potentially useful infrequent itemsets in D.

Steps (5) and (6) select all frequent and infrequent itemsets of interest. In Step (5), if an itemset Z in PL does not satisfy $fipi(Z)$, then Z is an uninteresting frequent itemset, and is removed from PL. After all uninteresting frequent itemsets are removed from PL; in Step (6), if an itemset (X, Y) in NL does not satisfy $iipi(XY)$, then (X, Y) is an uninteresting infrequent itemset, and is removed from NL. All uninterested frequent itemsets are removed from NL.

Step (7) outputs the frequent and infrequent itemsets of potential interest in PL and NL. The procedure ends in Step (8).

Identifying Complete Associations of Interest

Let I be the set of items in a database TD, $i = XY \subset I$ be an itemset, $X \cap Y = \Phi$, $p(X) \neq 0$, $p(Y) \neq 0$, and ms, mc, and $mi > 0$ be given by the user. There are four possible rules between X and Y as follows:

(i) If $p(XY) \geq ms$, $R(X, Y) \geq mi$, and $CF(X, Y) \geq mc$, then $X \rightarrow Y$ is a positive rule of interest.

(ii) If $p(X \neg Y) \geq ms$, $p(X) \geq ms$, $p(Y) \geq ms$, $R(X, \neg Y) \geq mi$, and $CF(X, \neg Y) \geq mc$, then $A \rightarrow \neg B$ is a negative rule of interest.

(iii) If $p(\neg XY) \geq ms$, $p(X) \geq ms$, $p(Y) \geq ms$, $R(\neg X, Y) \geq mi$, and $CF(\neg X, Y) \geq mc$, then $\neg X \rightarrow Y$ is a negative rule of interest.

(iv) If $p(\neg X \neg Y) \geq ms$, $p(A) \geq ms$, $p(Y) \geq ms$, $R(\neg X, \neg Y) \geq mi$, and $CF(\neg X, \neg Y) \geq mc$, then $\neg X \rightarrow \neg Y$ is a negative rule of interest.

In the above, Case (1) defines positive association rules of interest, whereas others are negative association rules of interest (in Cases 2, 3, and 4), where $p(*) \geq ms$ 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 $CF(*,*) \geq mc$ specifies the *conf* constraint.

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

Algorithm 3. Complete Association

Input: D: a database; ms, mc, mi: threshold values;

Output: association rules;

Step (1)

call procedure All Itemsets Of Interest;

Step (2) // Generate positive association rules in PL.

for each frequent itemset Z in PL **do**

for each expression $XY = Z$ and $X \cap Y = \Phi$ **do begin**

if $fipis(X, Y)$ **then**

if $CF(X, Y) \geq mc$ **then**

output the rule $X \rightarrow Y$

with confidence $CF(X, Y)$ and support $p(A)$;

if $CF(X, Y) \geq mc$ **then**

output the rule $Y \rightarrow X$

with confidence $CF(X, Y)$ and support $p(A)$;

end for;

Step (3) // Generate all negative association rules in NL.

for each itemset (X, Y) in NL **do**

if $iipis$ (X, Y) **then begin**

if $CF(\neg X, Y) \geq mc$ **then**

output the rule $\neg X \rightarrow Y$

with confidence $CF(\neg X, Y)$ and support $p(\neg X|Y)$;
if $CF(Y, \neg X) \geq mc$ **then**
output the rule $Y \rightarrow \neg X$
with confidence $CF(Y, \neg X)$ and support $p(Y \neg X)$;
if $CF(X, \neg Y) \geq mc$ **then**
output the rule $X \rightarrow \neg Y$
with confidence $CF(X,\neg Y)$ and support $p(X|\neg Y)$;
if $CF(\neg Y, X) \geq mc$ **then**
output the rule $\neg Y \rightarrow X$
with confidence $CF(\neg Y, X)$ and support $p(\neg YX)$;
if $CF(\neg X, \neg Y) \geq mc$ **then**
output the rule $\neg X \rightarrow \neg Y$
with confidence $CF(\neg X, \neg Y)$ and support $p(\neg X|\neg Y)$;
if $CF(\neg Y, \neg X) \geq mc$ **then**
output the rule $\neg Y \rightarrow \neg X$
with confidence $CF(\neg Y, \neg X)$ and support $p(\neg Y \neg X)$;
end if
Step (4) **return**.

Complete Association generates not only all positive association rules in *PL* but also negative association rules in *NL*. Step (1) calls procedure All Itemsets Of Interest 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 *XY* of *Z* in *PL* if *fipis*(*X*, *Y*). If $CF(X, Y) \geq mc$, $X \rightarrow Y$ is extracted as a valid rule of interest, with *conf* $CF(X, Y)$ and support $p(XY)$. If $CF(Y, X) \geq mc$, $Y \rightarrow X$ is extracted as a valid rule of interest, with *conf* $CF(Y, X)$ and support $p(XY)$.

Step (3) generates negative association rules of interest for an infrequent itemset (*X*, *Y*) in *NL* if *iipis*(*X*, *Y*). If $CF(\neg X, Y) \geq mc$, $\neg X \rightarrow Y$ is extracted as a valid rule of interest. If $CF(Y, \neg X) \geq mc$, $Y \rightarrow \neg X$ is extracted as a valid rule of interest. If $CF(X, \neg Y) \geq mc$, $X \rightarrow \neg Y$ is extracted as a valid rule of interest. If $CF(\neg Y, X) \geq mc$, $\neg Y \rightarrow X$ is extracted as a valid rule of interest. If $CF(\neg X, \neg Y) \geq mc$, $\neg X \rightarrow \neg Y$ is extracted as a valid rule of interest. If $CF(\neg Y, \neg X) \geq mc$, $\neg Y \rightarrow \neg X$ is extracted as a valid rule of interest.

TABLE 8 | A Transaction Database TD

Transaction ID	Items
T1	*A, B, D*
T2	*A, B, C, D*
T3	*B, D*
T4	*C, D, E*
T5	*A, E*
T6	*B, D, F*
T7	*A, E, F*
T8	*C, F*
T9	*B, C, F*
T10	*A, B, C, D, F*

Complete Association Analysis: an Illustration

This subsection illustrates the use of the All Itemsets Of Interest and Complete Association algorithms with the data in Table 8[g] as follows.

Suppose we have a *TD* with 10 transactions in Table 8 from a grocery store. Let A = bread, B = coffee, C = tea, D = sugar, E = beer, F = butter, $ms = 0.3$, $mc = 0.6$, and $mi = 0.05$.

The All Itemsets Of Interest algorithm works as follows: From Steps (1)–(3), *PL* is generated in the same way as in the Apriori algorithm, where $PL = \{A, B, C, D, E, F, AB, AD, BC, BD, BF, CD, CF, ABD\}$. The frequent itemsets and their frequencies are listed in Table 8.

Step (4) is a loop of generating potentially useful infrequent itemsets *NL*. For simplification, we can carry out this loop as follows:

(1) For itemset *A*, because the intersection of any one of $\{B, C, D, E, F, BC, BD, BF, CD, CF\}$ in *PL* and *A* is empty, $\{(A, B); (A, C); (A, D); (A, E); (A, F); (A, BC); (A, BD); (A, BF); (A, CD); (A, CF)\}$ is a set of candidate infrequent itemsets. Because of *AB*, *AD*, and *ABD* in *PL*, $\{(A, C); (A, E); (A, F); (A, BC); (A, BF); (A, CD); (A, CF)\}$ is the set of potentially useful infrequent itemsets.

(2) For itemset *B*, because the intersection of any one of $\{C, D, E, F, AD, CD, CF\}$ in *PL* and *B* is empty, $\{(B, C); (B, D); (B, E); (B, F); (B, AD); (B, CF)\}$ is a set of candidate infrequent itemsets. Because of *BC*, *BD*, *BF*, *BAD*, and *ABD* in *PL*, $\{(B, E); (B, CD)\}$ is the set of potentially useful infrequent itemsets.

(3) For itemset *C*, because the intersection of any one of $\{D, E, F, AB, AD, BD, BF, ABD\}$ in *PL* and *C* is empty, $\{(C, D); (C, E); (C, F); (C,$

TABLE 9 | All Frequent Itemsets and Their Frequencies in *TD*

Itemset	Number of Transactions	Support	Itemset	Number of Transactions	Support
A	5	0.5	*AD*	3	0.3
B	6	0.6	*BC*	3	0.3
C	5	0.5	*BD*	5	0.5
D	6	0.6	*BF*	3	0.3
E	3	0.3	*CD*	3	0.3
F	5	0.5	*CF*	3	0.3
AB	3	0.3	*ABD*	3	0.3

AB); (*C*, *AD*); (*C*, *BD*); (*C*, *BF*); (*C*, *ABD*)} is a set of candidate infrequent itemsets. Because of *CD* and *CF* in *PL*, {(*C*, *E*); (*C*, *AB*); (*C*, *AD*); (*C*, *BD*); (*C*, *BF*); (*C*, *ABD*)} is the set of potentially useful infrequent itemsets.

(4) For itemset *D*, because the intersection of any one of {*E*, *F*, *AB*, *BC*, *BF*, *CF*} in *PL* and *D* is empty, {(*D*, *E*); (*D*, *F*); (*D*, *AB*); (*D*, *BC*); (*D*, *BF*); (*D*, *CF*)} is a set of candidate infrequent itemsets. Because of *ABD* in *PL*, {(*D*, *E*); (*D*, *F*); (*D*, *BC*); (*D*, *BF*); (*D*, *CF*)} is the set of potentially useful infrequent itemsets.

(5) For itemset *E*, because the intersection of any one of {*F*, *AB*, *AD*, *BC*, *BD*, *BF*, *CD*, *CF*, *ABD*} in *PL* and *E* is empty, {(*E*, *F*); (*E*, *AB*); (*E*, *AD*); (*E*, *BC*); (*E*, *BD*); (*E*, *BF*); (*E*, *CD*); (*E*, *CF*); (*E*, *ABD*)} is a set of candidate infrequent itemsets. Because of none of them in *PL*, all of them are potentially useful infrequent itemsets.

(6) For itemset *F*, because the intersection of any one of {*AB*, *AD*, *BC*, *BD*, *CD*, *ABD*} in *PL* and *F* is empty, {(*F*, *AB*); (*F*, *AD*); (*F*, *BC*); (*F*, *BD*); (*F*, *CD*); (*F*, *ABD*)} is a set of candidate infrequent itemsets. Because of none of them in *PL*, all of them are potentially useful infrequent itemsets.

(7) For itemset *AB*, because the intersection of any one of {*CD*, *CF*} in *PL* and *AB* is empty, {(*AB*, *CD*); (*AB*, *CF*)} is a set of candidate infrequent itemsets. Because of none of them in *PL*, all of them are potentially useful infrequent itemsets.

(8) For itemset *AD*, because the intersection of any one of {*BC*, *BF*, *CF*} in *PL* and *AD* is empty, {(*AD*, *BC*); (*AD*, *BF*); (*AD*, *CF*)} is a set of candidate infrequent itemsets. Because of none of them in *PL*, all of them are potentially useful infrequent itemsets.

(9) For itemset *BC*, because no one in *PL* is applicable to *BC* to generate candidate infrequent itemsets, there is no potentially useful infrequent itemset in this case.

(10) For itemset *BD*, because only *CF* in *PL* is applicable to *BD* to generate candidate infrequent itemsets and the union is not in *PL*, (*BD*, *CF*) is a potentially useful infrequent itemset.

(11) For itemset *BF*, because only *CD* in *PL* is applicable to *BF* to generate candidate infrequent itemsets and the union is not in *PL*, (*BF*, *CD*) is a potentially useful infrequent itemset.

(12) For itemset *CD*, because no one of in *PL* is applicable to *BC* to generate candidate infrequent itemsets, there is no potentially useful infrequent itemset in this case.

(13) For itemset *CF*, because only *ABD* in *PL* is applicable to *CF* to generate candidate infrequent itemsets and the union is not in *PL*, (*CF*, *ABD*) is a potentially useful infrequent itemset.

(14) For itemset *ABD*, no one in *PL* is applicable to *ABD* to generate candidate infrequent itemsets.

Therefore, we have

$$NL = \{(A, C, 2); (A, E, 2); (A, F, 2); (A, BC, 2); (A, BF, 1); (A, CD, 2); (A, CF, 1); (B, E, 0); (B, CD, 2); (C, E, 1); (C, AB, 2); (C, AD, 2); (C, BD, 2); (C, BF, 2); (C, ABD, 2); (D, E, 1); (D, F, 2); (D, BC, 2); (D, BF, 2); (D, CF, 1);$$

$(E, F, 1); (E, AB, 0); (E, AD, 0); (E, BC, 0);$
$(E, BD, 0); (E, BF, 0); (E, CD, 1);$
$(E, CF, 0); (E, ABD, 0);$
$(F, AB, 1); (F, AD, 1); (F, BC, 2); (F, BD, 2);$
$(F, CD, 1); (F, ABD, 1);$
$(AB, CD, 2); (AB, CF, 1);$
$(AD, BC, 2); (AD, BF, 1); (AD, CF, 1);$
$(BD, CF, 1);$
$(BF, CD, 1);$
$(CF, ABD, 1)\}$

There are 43 pairs of infrequent itemsets of the form (X, Y, x), which is only used to simplify the description. It means that X and Y are itemsets and x is the frequency of the itemset XY.

Step (5) is a loop of pruning uninteresting itemsets in PL. We illustrate this step with the frequent 2-itemset BF and 3-itemset ABD as follows:

(i) Considering BF and for $B \rightarrow F$, we have $p(BF) = 0.3 = ms$, and

$$R(B, F) = |p(BF) - p(B)p(F)| = |0.3 - 0.6 \times 0.5| = 0 < mi$$

$$CF(B, F) = \frac{p(BF) - p(B)p(F)}{p(B)(1 - p(F))} = \frac{0.3 - 0.6 \times 0.5}{0.6(1 - 0.5)} = 0 < mc$$

This means $s(B, F) = 0$ and the function *fipi* is false.

On the contrary, for $F \rightarrow B$, we have $s(F, B) = 0$. Therefore, $g(F, B, 0.6, 0.05) = 0$ and the function *fipi* is false and BF is uninteresting and is removed from PL.

(ii) Considering ABD and for $AB \rightarrow D$, we have $p(ABD) = 0.3 = ms$, and

$$R(AB, D) = |p(ABD) - p(AB)p(D)| = |0.3 - 0.3 \times 0.6| = 0.12 > mi$$

$$CF(AB, D) = \frac{p(ABD) - p(AB)p(D)}{p(AB)(1 - p(D))} = \frac{0.3 - 0.3 \times 0.6}{0.3(1 - 0.6)} = 1 > mc$$

This means, $s(AB, D) = 1$, $g(AB, D, 0.6, 0.05) = 1$ and the function *fipi* is true. Therefore, ABD is of interest because there is at least one pair (AB, D) satisfying all the conditions for a frequent itemset of potential interest. And ABD is not removed from PL.

Step (6) is also a loop of pruning uninteresting itemsets in NL. Like the above, we illustrate this step with pairs (A, C) and (E, BC) as follows:

(a) Considering (A, C) and for $A \rightarrow \neg C$, we have $p(AC) = 0.2 < ms$, $p(A) = 0.5 > ms$, $p(C) = 0.5 > ms$, $p(A \neg C) = 0.3 > ms$ and

$$R(A, \neg C) = |p(A\neg C) - p(A)p(\neg C)| = |0.3 - 0.5 \times 0.5| = 0.05 = mi$$

$$CF(A, \neg C) = \frac{p(A\neg C) - p(A)p(\neg C)}{p(A)(1 - p(\neg C))} = \frac{0.3 - 0.5 \times 0.5}{0.5(1 - 0.5)} = 0.2 < mc$$

This means $t(A, \neg C) = 0$.

For $\neg C \rightarrow A$, we have $p(AC) = 0.2 < ms$, $p(A) = 0.5 > ms$, $p(C) = 0.5 > ms$, $p(\neg CA) = 0.3 > ms$, and

$$R(\neg C, A) = |p(\neg CA) - p(\neg C)p(A)| = |0.3 - 0.5 \times 0.5| = 0.05 = mi$$

$$CF(\neg C, A) = \frac{p(\neg CA) - p(\neg C)p(A)}{p(\neg C)(1 - p(A))} = \frac{0.3 - 0.5 \times 0.5}{0.5(1 - 0.5)} = 0.2 < mc$$

This means $t(\neg C, A) = 0$.

For $\neg A \rightarrow C$, we have $p(AC) = 0.2 < ms$, $p(A) = 0.5 > ms$, $p(C) = 0.5 > ms$, $p(\neg AC) = 0.3 > ms$, and

$$R(\neg A, C) = |p(\neg AC) - p(\neg A)p(C)| = |0.3 - 0.5 \times 0.5| = 0.05 = mi$$

$$CF(\neg A, C) = \frac{p(\neg AC) - p(\neg A)p(C)}{p(\neg A)(1 - p(C))} = \frac{0.3 - 0.5 \times 0.5}{0.5(1 - 0.5)} = 0.2 < mc$$

This means $t(\neg A, C) = 0$.

For $C \rightarrow \neg A$, we have $p(AC) = 0.2 < ms$, $p(A) = 0.5 > ms$, $p(C) = 0.5 > ms$, $p(C \neg A) = 0.3 > ms$, and

$$R(C, \neg A) = |p(C\neg A) - p(C)p(\neg A)| = |0.3 - 0.5 \times 0.5| = 0.05 = mi$$

$$CF(C, \neg A) = \frac{p(C\neg A) - p(C)p(\neg A)}{p(C)(1 - p(\neg A))} = \frac{0.3 - 0.5 \times 0.5}{0.5(1 - 0.5)} = 0.2 < mc$$

This means $s(C, \neg A) = 0$.

For $\neg A \rightarrow \neg C$, $p(\neg A \neg C) = 0.2 < ms$, so does for $\neg C \rightarrow \neg A$.

From the above, $h(A, C, 0.6, 0.05) = 0$, AC is uninteresting and (A, C) is removed from NL.

(b) Considering (E, BC) and for $E \rightarrow \neg BC$, we have $p(EBC) = 0 < ms$, $p(E) = 0.3 = ms$, $p(BC) = 0.3 = ms$, p(BC) = 0.3 = ms, $p(E\neg BC) = 0.3 > ms$ and

$$R(E, \neg BC) = |p(E\neg BC) - p(E)p(\neg BC)| = |0.3 - 0.3 \times 0.3| = 0.21 > mi$$

$$CF(E, \neg BC) = \frac{p(E\neg BC) - p(E)p(\neg BC)}{p(E)(1 - p(\neg BC))} = \frac{0.3 - 0.3 \times 0.3}{0.3(1 - 0.3)} = 1 > mc$$

This means $s(E, \neg BC) = 1$, $h(E, BC, 0.6, 0.05) = 1$, and the function *iipi* is true. Therefore, (E, BC) is of interest because there is at a rule $E \rightarrow \neg BC$ satisfying all the conditions for an infrequent itemset of potential interest. And (E, BC) is not removed from NL.

In Step (7), PL and NL are output. The All Itemsets Of Interest algorithm is terminated in Step (8).

The Complete Association algorithm works as follows: Step (1) calls the All Itemsets Of Interest procedure to generate the PL and NL. We illustrate the use of the Complete Association algorithm with the same frequent itemset ABD and infrequent itemset (E, BC) as above.

Step (2) is a loop of generating positive association rules in PL. We only demonstrate the loop with the frequent itemset ABD as follows.

Because $g(F, B, 0.6, 0.05) = 1$ and the function *fipi* is true for ABD, we can generate some interesting association rules from frequent itemset ABD.

For $AB \rightarrow D$, we have $p(ABD) = 0.3 = ms$, and

$$R(AB, D) = |p(ABD) - p(AB)p(D)| = |0.3 - 0.3 \times 0.6| = 0.12 > mi$$

$$CF(AB, D) = \frac{p(ABD) - p(AB)p(D)}{p(AB)(1 - p(D))} = \frac{0.3 - 0.3 \times 0.6}{0.3(1 - 0.6)} = 1 > mc$$

This means $s(AB, D) = 1$, $g(AB, D, 0.6, 0.05) = 1$, and the function *fipi* is true. Therefore, $AB \rightarrow D$ is an association rule of interest.

For $AD \rightarrow B$, we have $p(ABD) = 0.3 = ms$, and

$$R(AD, B) = |p(ABD) - p(AD)p(B)| = |0.3 - 0.3 \times 0.6| = 0.12 > mi$$

$$CF(AD, B) = \frac{p(ABD) - p(AD)p(B)}{p(AD)(1 - p(B))} = \frac{0.3 - 0.3 \times 0.6}{0.3(1 - 0.6)} = 1 > mc$$

This means $s(AD, B) = 1$, $g(AD, B, 0.6, 0.05) = 1$, and the function *fipi* is true. Therefore, $AD \rightarrow B$ is an association rule of interest.

For $BD \rightarrow A$, we have $p(ABD) = 0.3 = ms$, and

$$R(BD, A) = |p(ABD) - p(BD)p(A)| = |0.3 - 0.5 \times 0.5| = 0.05 = mi$$

$$CF(BD, A) = \frac{p(ABD) - p(BD)p(A)}{p(BD)(1 - p(A))} = \frac{0.3 - 0.5 \times 0.5}{0.5(1 - 0.5)} = 0.2 < mc$$

This means $s(BD, A) = 0$. Therefore, $BD \rightarrow A$ is not of interest.

For $D \rightarrow AB$, we have $p(ABD) = 0.3 = ms$, and

$$R(D, AB) = |p(ABD) - p(D)p(AB)| = |0.3 - 0.6 \times 0.3| = 0.12 > mi$$

$$CF(D, AB) = \frac{p(ABD) - p(D)p(AB)}{p(D)(1 - p(AB))} = \frac{0.3 - 0.6 \times 0.3}{0.6(1 - 0.3)} = 0.2857 < mc$$

This means $s(D, AB, D) = 0$. Therefore, $D \rightarrow AB$ is not of interest.

For $B \rightarrow AD$, we have $p(ABD) = 0.3 = ms$, and

$$R(B, AD) = |p(ABD) - p(B)p(AD)| = |0.3 - 0.6 \times 0.3| = 0.12 > mi$$

$$CF(B, AD) = \frac{p(ABD) - p(B)p(AD)}{p(B)(1 - p(AD))} = \frac{0.3 - 0.6 \times 0.3}{0.6(1 - 0.3)} = 0.2857 < mc$$

This means $s(B, AD) = 0$. Therefore, $B \rightarrow AD$ is not of interest.

For $A \rightarrow BD$, we have $p(ABD) = 0.3 = ms$, and

$$R(A, BD) = |p(ABD) - p(A)p(BD)| = |0.3 - 0.5 \times 0.5| = 0.05 = mi$$

$$CF(A, BD) = \frac{p(ABD) - p(A)p(BD)}{p(A)(1 - p(BD))} = \frac{0.3 - 0.5 \times 0.5}{0.5(1 - 0.5)} = 0.2 < mc$$

This means $s(A, BD) = 0$. Therefore, $A \rightarrow BD$ is not of interest.

From the above, two positive association rules, $AB \rightarrow D$ with *conf* $CF(AB, D) = 1$ and support $p(ABD) = 0.3$, and $AD \rightarrow B$ with *conf* $CF(AD, B) = 1$ and support $p(ABD) = 0.3$, are output as valid rules.

Step (3) is also a loop that generates negative association rules in NL. We only demonstrate the loop with the infrequent itemset (E, BC) as follows.

Because $h(E, BC, 0.6, 0.05) = 1$ and the function *iipi* is true for (E, BC), we can generate some interesting negative association rules from infrequent itemset (E, BC).

For $E \rightarrow \neg BC$, we have $p(EBC) = 0 < ms$, $p(E) = 0.3 = ms$, $p(BC) = 0.3 = ms$, $p(E\neg BC) = 0.3 > ms$, and

$$R(E, \neg BC) = |p(E\neg BC) - p(E)p(\neg BC)| = |0.3 - 0.3 \times 0.3| = 0.21 > mi$$

$$CF(E, \neg BC) = \frac{p(E\neg BC) - p(E)p(\neg BC)}{p(E)(1 - p(\neg BC))} = \frac{0.3 - 0.3 \times 0.3}{0.3(1 - 0.3)} = 1 > mc$$

This means $s(E, \neg BC) = 1$, $h(E, BC, 0.6, 0.05) = 1$, and the function *iipi* is true. Therefore, $E \rightarrow \neg BC$ is a negative association rule of interest.

For $\neg E \rightarrow BC$, we have $p(EBC) = 0 < ms$, $p(E) = 0.3 = ms$, $p(BC) = 0.3 = ms$, $p(\neg EBC) = 0.3 > ms$, and

$$R(\neg E, BC) = |p(\neg EBC) - p(\neg E)p(BC)| = |0.3 - 0.7 \times 0.3| = 0.09 > mi$$

$$CF(\neg E, BC) = \frac{p(\neg EBC) - p(\neg E)p(BC)}{p(\neg E)(1 - p(BC))} = \frac{0.3 - 0.7 \times 0.3}{0.7(1 - 0.3)} = 0.1837 < mc$$

This means $s(\neg E, BC) = 0$. Therefore, $\neg E \rightarrow BC$ is not of interest.

For $\neg E \rightarrow \neg BC$, we have $p(EBC) = 0 < ms$, $p(E) = 0.3 = ms$, $p(BC) = 0.3 = ms$, $p(\neg E\neg BC) = 0.4 > ms$

$$R(\neg E, \neg BC) = |p(\neg E\neg BC) - p(\neg E)p(\neg BC)| = |0.4 - 0.7 \times 0.7| = 0.09 > mi$$

$$CF(\neg E, \neg BC) = \frac{p(\neg E\neg BC) - p(\neg E)p(\neg BC)}{-p(\neg E)p(\neg BC)} = \frac{0.4 - 0.7 \times 0.7}{-0.7 \times 0.7} = 0.1837 < mc$$

where $p(\neg BC) > p(\neg BC|\neg E)$. This means $s(\neg E, \neg BC) = 0$. Therefore, $\neg E \rightarrow \neg BC$ is not of interest.

For $BC \rightarrow \neg E$, we have $p(EBC) = 0 < ms$, $p(E) = 0.3 = ms$, $p(BC) = 0.3 = ms$, $p(BC\neg E) = 0.3 > ms$, and

$$R(BC, \neg E) = |p(BC\neg E) - p(BC)p(\neg E)| = |0.3 - 0.3 \times 0.7| = 0.09 > mi$$

$$CF(BC, \neg E) = \frac{p(BC\neg E) - p(BC)p(\neg E)}{p(BC)(1 - p(\neg E))} = \frac{0.3 - 0.3 \times 0.7}{0.3(1 - 0.7)} = 1 > mc$$

This means $s(BC, \neg E) = 1$, $h(BC, E, 0.6, 0.05) = 1$, and the function *iipi* is true. Therefore, $BC \rightarrow \neg E$ is a negative association rule of interest.

For $\neg BC \rightarrow E$, we have $p(\neg BCE) = 0 < ms$, $p(E) = 0.3 = ms$, $p(BC) = 0.3 = ms$, $p(\neg BCE) = 0.3 > ms$, and

$$R(\neg BC, E) = |p(\neg BCE) - p(\neg BC)p(E)| = |0.3 - 0.7 \times 0.3| = 0.09 > mi$$

$$CF(\neg BC, E) = \frac{p(\neg BCE) - p(\neg BC)p(E)}{p(\neg BC)(1 - p(E))} = \frac{0.3 - 0.7 \times 0.3}{0.7(1 - 0.3)} = 0.1837 < mc$$

This means $s(\neg BC, E) = 1, h(BC, E, 0.6, 0.05) = 1$, and the function *iipi* is true. Therefore, $\neg BC \rightarrow E$ is a negative association rule of interest.

For $\neg BC \rightarrow \neg E$, we have $p(\neg BC\neg E) = 0 < ms$, $p(E) = 0.3 = ms$, $p(BC) = 0.3 = ms$, $p(\neg BC\neg E) = 0.4 > ms$, and

$$R(\neg BC, \neg E) = |p(\neg BC\neg E) - p(\neg BC)p(\neg E)| = |0.4 - 0.7 \times 0.7| = 0.09 > mi$$

$$CF(\neg BC, \neg E) = \frac{p(\neg BC\neg E) - p(\neg BC)p(\neg E)}{-p(\neg BC)p(\neg E)} = \frac{0.4 - 0.7 \times 0.7}{-0.7 \times 0.7} = 0.1837 < mc$$

where $p(\neg E) > p(\neg E|\neg BC)$. This means $s(\neg BC, \neg E) = 0$. Therefore, $\neg BC \rightarrow \neg E$ is not of interest.

From the above, two negative association rules, $E \rightarrow \neg BC$ with *conf* $CF(E, BC) = 1$ and support $p(E\neg BC) = 0.3$, $BC \rightarrow \neg E$ with *conf* $CF(BC, \neg E) = 1$ and support $p(BC\neg E) = 0.3$, are output as valid rules.

The Complete Association algorithm is ended in Step (4).

Research Directions in CAR Analysis

Since the definition in Wu et al.,[44] mining positive and negative association rules, referred to CAR mining in this paper, have become an active research topic. There are three lines of main research efforts in CAR analysis, and we outline them as follows.

Algorithm Scale-Up

Research efforts in this direction include different frameworks,[45–48] new pruning strategies,[21,49] and confined rule mining.[50]

Indirect Associations

An itempair {a,b} is indirectly associated via an itemset (called a mediator) Y if the following conditions hold[51–53]:

(1) sup(a,b) ≥ minsupp (Itempair Support Condition), and

(2) There exists a nonempty itemset Y such that forall $y_i \in Y$:

(a) sup(a,y_i) ≥ minsupp, sup(b,y_i) ≥ minsupp (Mediator Support Condition).

(b) d(a,y_i) ≥ t_d, d(b, y_i) ≥ t_d where d(p, q) is a measure of the dependence between p and q (Dependence Condition).[52]

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 a lattice of frequent itemsets (FI) has been generated using an existing algorithm such as Apriori.[52] During each pass of candidate generation, it will find all FIs, $y_i \in I - \{a,b\}$, such that both $\{a\} \cup y_i \in FI$ and $\{b\} \cup y_i \in FI$. Indirect association mining has been implemented using (SAS Institute Inc., Cany, North Carolina, USA) and tested on various data sets including Web log, text, retail, and stock market data to demonstrate its utility, and can be combined with negative association analysis.

Applications

There are many successful cases in real applications, for example, identifying complex spatial relationships,[54] detecting adverse drug reactions,[55] discovering XML query patterns for caching,[56] exploring the relationship between urban land surface temperature and biophysical/social parameters,[57] hyperlink assessment,[58] and filtering Web recommendation lists.[59]

APPLICATIONS OF ASSOCIATION RULES

Association rules have been widely used in real applications. We have mentioned some of them in the previous sections. Here, we outline them from the following perspectives: (1) data mining and machine learning (e.g., associative classification and clustering), (2) search engines (e.g., cube computation and analysis), and (3) other applications.

Applications to Data Mining and Machine Learning

Association rules have been demonstrated to be useful in the areas of data mining and machine learning. One of the important applications is classification. A well-known application of association rules to classification is the Classification Based on Associations

algorithm.[38] The idea is to first identify the association between a frequent pattern and a class label, and then the discovered association rules are used for predicting unlabelled data. From published reports on associative classification, it can be more accurate than typical classification methods, such as C4.5. Other main reports include emerging patterns-based classifiers in Dong and Li[60] and Li et al.,[61] classification based on multiple association rules in Li et al.,[62] classification based on predictive association rules in Yin and Han,[63] and the classifier Refined Classification Based on Top-k rule groups (RCBT) in Cong et al.[64]

Another important application is clustering. It is mainly applied to high-dimensional data clustering. A well-established application CLustering In QUEst (CLIQUE) is given in Agrawal et al.,[65] which is an Apriori-based dimension-growth subspace clustering algorithm. It integrates density-based and grid-based clustering methods. The Apriori property is used to find clusterable subspaces, and dense units are identified. The algorithm then finds adjacent dense grid units in the selected subspaces using a depth first search. Clusters are formed by combining these units using a greedy growth scheme. An entropy-based subspace clustering algorithm for mining numerical data, called entropy-based subspace clustering (ENCLUS), was proposed by Cheng et al.[66] Beil et al.[67] proposed a method for frequent term-based text clustering. Wang et al.[68] proposed pCluster, a pattern similarity-based clustering method for microarray data analysis, and demonstrated its effectiveness and efficiency for finding subspace clusters in a high-dimensional space.

Applications to Search Engines

In real applications, data are simply large, and it is inefficient to perform a sequential scan on the whole database and examine objects one by one. To efficiently search large and complex databases, Yan et al.[69] proposed a discriminative frequent pattern-based approach to index structures and graphs, called gIndex. SeqIndex is one example using a frequent pattern-based approach to index sequences. Taking frequent patterns as features, new strategies to perform structural similarity search were developed in Grafil[34] and Partition-based graph Index and Search (PIS).[70]

Another application is the iceberg cube computation. The first algorithm to compute iceberg cubes is bottom-up computation (BUC) proposed by Beyer and Ramakrishnan.[71] It is based on the Apriori property. More examples include the cubegrade algorithm,[72] the LiveSet-Driven method,[73] and the ConSGapMiner technique.[74]

Also, association rule mining techniques have been successfully applied to discover patterns and knowledge from the Web.[75,76] It includes Web usage mining, Web structure mining, and Web content mining. An early application of association rule mining to Web data is the analysis of users' browse behaviors, called Web usage mining. It includes user grouping, page association, and sequential clicks through analysis. Web content mining identifies potentially useful information within Web pages, whereas Web structure mining discovers useful structure linkage among Web pages. Other applications include, for example, discovering XML query patterns for caching,[56] hyperlink assessment,[58] and filtering Web recommendation lists.[59]

Applications to Other Subjects

For trusty software development, association rule mining techniques have been applied to software bug mining[60] and software change history.[77–79] For example, Liu et al.[80] developed a method to classify the structured traces of program executions using software behavior graphs. It utilizes a frequent graph mining technique. Suspicious buggy regions are identified through the capture of the classification accuracy change, which is measured incrementally during program execution.

Other applications include identifying complex spatial relationships,[54] detecting adverse drug reactions,[55] and exploring the relationship between urban land surface temperature and biophysical/social parameters.[57]

CONCLUSIONS

Along with classification and clustering, both of which are mentioned in *Applications of Association Rules*, association analysis is among the three fundamental techniques in data mining. This paper has reviewed the traditional Apriori algorithm and mining both negative and positive association rules in detail with illustrative examples. We have also provided an account on the research directions and applications. Because of space constraints, we have not covered sequential association analysis and other advanced topics in this chapter.

NOTES

[a] It is adapted from (Zhang and Zhang 2002).

[b]From the definition of Tem_k, there is only one candidate in the second iteration.
[c]Some data are adapted from Refs 18, 21.
[d]It is adapted from Ref 21.
[e]The techniques are similar to that in Ref 21.
[f]For convenience of identifying negative rules, an infrequent itemset XY in NL is written to (X, Y).
[g]The data are slightly different from that in Wu et al.[21]

ACKNOWLEDGEMENTS

This work was supported in part by the Australian Research Council under grant DP0985456, the Nature Science Foundation (NSF) of China under grant 90718020, the China 973 Program under grant 2008CB317108, the Research Program of China Ministry of Personnel for Overseas-Return High-level Talents, the MOE Project of Key Research Institute of Humanities and Social Sciences at Universities (07JJD720044), and the Guangxi NSF (Key) grants.

REFERENCES

1. Frawley WJ, Piatetsky-Shapiro G, Matheus CJ. Knowledge discovery in databases: An overview, *AI Magazine*, 1992, 13:57–70.
2. Fayyad U, Piatetsky-Shapiro G, Smyth P. From data mining to knowledge discovery: an overview. *Adv Knowledge Discov Data Min* 1996, 1–34.
3. Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*. 1993, 207–216.
4. Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases. *Proceedings of the Twentieth International Conference on Very Large Databases*. 1994, 487–499.
5. Hon J, et al. Mining Frequent Patterns without Candidate Generation. *Proceedings 2000 ACM-SIGMOD International Conference on Management of Data (SIGMOD'00)*, Dallas, TX, May 2000, 1–12.
6. Han J, Wang J, Lu Y, Tzvetkov P. Mining top-K frequent closed patterns without minimum support. In: *Proceedings of ICDM*. 2002, 211–218.
7. Zhang C, Zhang S, Webb G. Identifying approximate itemsets of interest in large databases. *Appl Int* 2003, 18:91–104.
8. Yan X, Zhang C, Zhang S. On data structures for association rule discovery. *Appl Artif Intell* 2007, 21:57–79.
9. Park J, Chen M, Yu P. An effective hash-based algorithm for mining association rules. *Proceedings of ACM SIGMOD International Conference on Management of Data*, 1995, 175–186.
10. Savasere A, Omiecinski E, Navathe S. An efficient algorithm for mining association rules in large databases. *Proceedings of the 21nd International Conference on Very Large Databases*. 1995, 432–444.
11. Zhang S, Wu X. Large Scale Data Mining Based on Data Partitioning. *Applied Artificial Intelligence*, 2001, 15:129–139.
12. Toivonen H. Sampling large databases for association rules. *Proceedings of the 22nd International Conference on Very Large Databases*. 1996, 134–145.
13. Zhang S, Zhang C. Anytime mining for multiuser applications. *IEEE Trans Syst Man Cybern A Syst Hum* 2002, 32:515–521.
14. Agrawal R, Shafer J. Parallel mining of association rules. *IEEE Transactions on Knowledge and Data Engineering*, 1996, 8:962–969.
15. Cheung D, Han J, Ng V, Wong C. Maintenance of discovered association rules in large databases: an incremental updating technique. *Proceedings of the 12th IEEE International Conference on Data Engineering*. 1996, 106–114.
16. Zaki M, et al. New Algorithms for Fast Discovery of Association Rules. *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, 1997, 283–286.
17. Sarawagi S, Thomas S, Agrawal R: Integrating Mining with Relational Database Systems: Alternatives and Implications. *Proceedings of ACM SIGMOD International Conference on Management of Data*, 1998, 343–354.
18. Brin S, Motwani R, Silverstein C. Beyond market baskets: generalizing association rules to correlations. *Proceedings of the ACM SIGMOD Conference*. 1997, 265–276.
19. Silverstein C, Brin S, Motwani R. Beyond Market Baskets: Generalizing Association Rules to Dependence Rules. *Data Min Knowl Discov*, 1998, 2:39–68.

20. Piatetsky-Shapiro G. Discovery, Analysis, and Presentation of Strong Rules. *Knowledge Discovery in Databases*, 1991, 229–248.
21. Wu X, Zhang C, Zhang S. Efficient mining of both positive and negative association rules. *ACM Trans Inf Syst* 2004, 22:381–405.
22. Wang Ke, Tay W, Liu B. An Interestingness-Based Interval Merger for Numeric Association Rules. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, New York, USA, 1998, 121–127.
23. Srikant R, Agrawal R. Mining generalized association rules. *Proceedings of the 21nd International Conference on Very Large Databases*. 1995, 407–419.
24. Zhang S, Zhang C. Discovering causality in large databases. *Appl Artif Intell* 2002, 16:333–358.
25. Han J, Pei J, Yin Y, Mao R. Mining frequent patterns without candidate generation: a frequent-pattern tree approach. *Data Min Knowledge Discov* 2004, 8:53–87.
26. Han J, Kamber M. Data Mining: Concepts and Techniques. The Morgan Kaufmann Series in Data Management Systems, 2006.
27. Kamber M, Han J, Chiang J: Metarule-Guided Mining of Multi-Dimensional Association Rules Using Data Cubes. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 1997, 207–210.
28. Piatetsky-Shapiro G, Steingold S. Measuring lift quality in database marketing. *SIGKDD Explor* 2000, 2:76–80.
29. Cohen E, Datar M, Fujiwara S, Gionis A, Indyk P, Motwani R, Ullman JD, Yang C. Finding interesting associations without support pruning. *IEEE Trans Knowledge Data Eng* 2001, 13:64–78.
30. Roddick JF, Rice S. What's interesting about cricket? On thresholds and anticipation in discovered rules. *SIGKDD Explor* 2001, 3:1–5.
31. Hipp J, Guntzer U. Is pushing constraints deeply into the mining algorithms really what we want? *SIGKDD Explor* 2002, 4:50–55.
32. Wang K, He Y, Cheung D, Chin F. Mining confident rules without support requirement. In: *Proceedings of the 10th ACM International Conference on Information and Knowledge Management*. 2001, 89–96.
33. Wang K, He Y, Han J. Pushing support constraints into association rules mining. *IEEE Trans Knowledge Data Eng* 2003, 15:642–658.
34. Yan X, Zhang C, Zhang S. Armga: identifying interesting association rules with genetic algorithms. *Appl Artif Intell* 2005, 19:677–689.
35. Zhang S, Wu X, Zhang C, Lu J. Computing the minimum-support for mining frequent patterns. *Knowledge Inf Syst* 2008, 15:233–257.
36. Zhang S, Wu X, Zhang C, Multi-Database Mining. *IEEE Computational Intelligence Bulletin*, June 2003, 2:5–13.
37. Zhang S, Zaki M. Mining multiple data sources: local pattern analysis. *Data Min Knowledge Discov* 2006, 12:121–125.
38. Liu H, Lu H, Yao J. Identifying relevant databases for multi-database mining. *Proceeding of PAKDD*. 1998, 210–221.
39. Zhong N, Yao Y, Ohshima M. Peculiarity Oriented Multidatabase Mining. *IEEE Trans Knowl Data Eng*, 2003, 15:952–960.
40. Kum H, Chang J, Wang W. Sequential pattern mining in multi-databases via multiple alignment. *Data Min Knowledge Discov* 2006, 12:151–180.
41. Zhu X, Wu X, Chen Q. Bridging local and global data cleansing: identifying class noise in large, distributed data datasets. *Data Min Knowledge Discov* 2006, 12:275–308.
42. Ling C, Yang Q. Discovering classification from data of multiple sources. *Data Min Knowledge Discov* 2006, 12:181–201.
43. Zaki M. Parallel and distributed association mining: a survey. *IEEE Concurrency*. 1999.
44. Wu X, Zhang C, Zhang S. Mining Both Positive and Negative Association Rules. In: *Proceedings of the 19th International Conference on Machine Learning*, Sydney, Australia, July 2002, 658–665.
45. Goncalves E, Mendes I, Plastino A. Mining exceptions in databases. AI 2004: advances in artificial intelligence. *17th Australian Joint Conference on Artificial Intelligence*. 2004, 1076–1081.
46. Pedreshi D, Ruggieri S, Turini F. Discrimination-aware data mining. *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2008, 560–568.
47. Shimada K, Hirasawa K, Hu J. Class association rule mining with chi-squared test using genetic network programming. *IEEE International Conference on Systems, Man and Cybernetics*. (SMC06), 2006, 5338–5344.
48. Zhao L, Zaki MJ, Ramakrishnan N. BLOSOM: a framework for mining arbitrary Boolean expressions. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*. 2006, 827–832.
49. Wan Q, An A. An efficient approach to mining indirect associations. *J Intell Inf Sys* 2006, 27:135–158.
50. Antonie M, Zaiane O. Mining positive andnegative association rules: an approach for confined rules. *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*. 2004, 27–38.
51. Tan P-N, Kumar V, Kuno H. Using SAS for mining indirect associations in data. In *Proc of the Western Users of SAS Software Conference*. 2001.

52. Tan P, Kumar V, Srivastava J. Indirect association: Mining higher order dependencies in data. In *Principles of Data Mining and Knowledge Discovery*. Springer, Lyon, France, 2000, 632–637.
53. Tan P, Kumar V, Srivastava J. Selecting the right interestingness measure for association patterns. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 2002, 32–41.
54. Munro R, Chawla S, Sun P. Complex spatial relationships. *Third IEEE International Conference on Data Mining* (ICDM'03). 2003, 227.
55. Jin HW, Chen J, He H, Williams GJ, Kelman C, O'Keefe CM. Mining unexpected temporal associations: applications in detecting adverse drug reactions. *IEEE Trans Inf Technol Biomed* 2008, 12:488–500.
56. Chen L, Bhowmick SS, Chia LT. Mining positive and negative association rules from XML query patterns for caching. *DASFAA-05*. 2005, 736–747.
57. Rajasekar U, Weng Q. Application of association rule mining for exploring the relationship between urban land surface temperature and biophysical/social parameters. *Photogramm Eng Remote Sensing* 2009, 75:385–396.
58. Kazienko P and Pilarczyk M. Hyperlink assessment based on web usage mining. *Proceedings of the Seventeenth Conference on Hypertext and Hypermedia*. 2006, 85–88.
59. Kazienko P. Filtering of web recommendation lists using positive and negative usage patterns. *Knowledge-Based Intelligent Information and Engineering Systems*. 2007, 1016–1023.
60. Dong G, Li J. Efficient mining of emerging patterns: Discovering trends and differences. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 1999, 43–52.
61. Li J, Dong G, Ramamohanarao K. Instance-Based Classification by Emerging Patterns. *Principles of Data Mining and Knowledge Discovery (PKDD-00)*, 2000, 191–200.
62. Li J, Ramamohanarao K, Dong G. Combining the Strength of Pattern Frequency and Distance for Classification. *Knowledge Discovery and Data Mining (PAKDD-01)*, 2001, 455–466.
63. Yin X, Han J. CPAR: Classification based on Predictive Association Rules. *Proceedings of the Third SIAM International Conference on Data Mining*, San Francisco, CA, USA, May 1–3, 2003, Student Paper 5.
64. Cong G, Tan K, Tung A, Xu X. Mining Top-k Covering Rule Groups for Gene Expression Data. In: *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2005, 670–681.
65. Agrawal R, Gehrke J, Gunopulos D, Raghavan P. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In: *Proceedings ACM SIGMOD International Conference on Management of Data*, 1998, 94–105.
66. Cheng CH, Fu AW, Zhang Y. Entropy-based subspace clustering for mining numerical data. In: *Proceeding of International Conference on Knowledge Discovery and Data Mining (KDD'99)*, 1999, 84–93.
67. Beil F, Ester M, Xu X. Frequent term-based text clustering. In: *Proceeding of ACM SIGKDD International Conference on Knowledge Discovery in Databases (KDD'02)*, 2002, 436–442.
68. Wang H, Wang W, Yang J, Yu PS. Clustering by pattern similarity in large data sets. In: *Proceeding of ACM-SIGMOD International Conference on Management of Data*, 2002, 418–427.
69. Yan X, Zhang C, Zhang S. Identifying Software Component Association with Genetic Algorithm. *International Journal of Software Engineering and Knowledge Engineering*, 2004, 14:441–447.
70. Yan X, Zhang C, Zhang S. On Data Structures for Association Rule Discovery. *Applied Artificial Intelligence*, 2007, 21:57–79.
71. Beyer K, Ramakrishnan R. Bottom-up computation of sparse and iceberg cubes. In: *Proceeding of ACM-SIGMOD International Conference on Management of Data*, 1999, 359–370.
72. Imielinski T, Khachiyan L, Abdulghani A Cubegrades: generalizing association rules. *Data Min Knowl Discov*, 2002, 6:219–258.
73. Dong G, Han J, Lam J, Pei J, Wang K, Zou W. Mining constrained gradients in multi-dimensional databases. *IEEE Trans Knowl Data Eng*, 2004, 16:922–938.
74. Ji X, Bailey J, Dong G. Mining minimal distinguishing subsequence patterns with gap constraints. In: *Proceeding of International Conference on Data Mining (ICDM'05)*, 2005, 194–201.
75. Kosala R, Blockeel H. Web mining research: a survey. *ACM SIGKDD Explorations*, 2000, 2:1–15.
76. Srivastava J, Cooley R, Deshpande M, Tan PN. Web usage mining: discovery and applications of usage patterns from web data. *ACM SIGKDD Explorations*, 2000, 1:12–23.
77. Shirabad J, Lethbridge T, Matwin S. Mining the maintenance history of a legacy software system. *ICSM-2003*. 2003, 95–104.
78. Ying A, Murphy G, Ng R, Chu-carroll M. Predicting source code changes by mining change history. *IEEE Trans Software Eng* 2004, 30:574–586.
79. Zhao Q, Bhowmick S Mining history of changes to web access patterns. *PKDD-2004*. 2004, 521–523.
80. Liu C, Yan X, Yu H, Han J, Yu P. Mining behavior graphs for "backtrace" of noncrashing bugs. In: *Proceeding of the 2005 SIAM international conference on data mining* (SDM'05), Newport Beach: 2005, 286–297.

