Generating Concise Association Rules

Yue Xu
Faculty of Information Technology
Queensland University of Technology
Brisbane, QLD4001, Australia
yue.xu@qut.edu.au

Yuefeng Li
Faculty of Information Technology
Queensland University of Technology
Brisbane, QLD4001, Australia
y2.li@qut.edu.au

ABSTRACT

Association rule mining has made many achievements in the area of knowledge discovery. However, the quality of the extracted association rules is a big concern. One problem with the quality of the extracted association rules is the huge size of the extracted rule set. As a matter of fact, very often tens of thousands of association rules are extracted among which many are redundant thus useless. Mining non-redundant rules is a promising approach to solve this problem. The Min-max exact basis proposed by Pasquier et al [Pasquier05] has showed exciting results by generating only non-redundant rules. In this paper, we first propose a relaxing definition for redundancy under which the Min-max exact basis still contains redundant rules; then we propose a condensed representation called Reliable exact basis for exact association rules. The rules in the Reliable exact basis are not only non-redundant but also more succinct than the rules in Min-max exact basis. We prove that the redundancy eliminated by the Reliable exact basis does not reduce the belief to the Reliable exact basis. The size of the Reliable exact basis is much smaller than that of the Min-max exact basis. Moreover, we prove that all exact association rules can be deduced from the Reliable exact basis. Therefore the Reliable exact basis is a lossless representation of exact association rules. Experimental results show that the Reliable exact basis significantly reduces the number of non-redundant rules.

Categories and Subject Descriptors

H2.8 [**Database Management**]: Database application - Data mining

General Terms

Algorithms

Keywords

Redundant association rules, closed itemsets, generators

CIKM'07, November 6–8, 2007, Lisboa, Portugal.
Copyright 2007 ACM 978-1-59593-803-9/07/0011 ...$5.00.

1. INTRODUCTION

Association rule mining, firstly stated in [1], has become one of the most important data mining techniques. It aims to extract interesting correlations and associations among sets of items in large datasets. Traditionally, two phases are involved in mining association rules: extracting frequent itemsets and generating association rules from the frequent itemsets with the constraints of minimal support and minimal confidence. The rules whose confidence is larger than a user-specified minimum confidence threshold are considered interesting or useful. This is known as frequent itemset based approach. Various models have been proposed, most of them are focused on developing novel algorithms and data structures to aid efficient computation of such rules [2, 3, 6, 9, 13, 22], especially on improving the efficiency of generating frequent itemsets. However, the quality of the mined association rules hasn't drawn adequate attention. As a matter of fact, very often the resulting rule base can easily contain several thousands of rules among which are many redundancies and thus useless in practice. Many efforts have been made on solving this problem including defining various interest measures, incorporating constraints into mining process, or designing specific templates to mine for restricted rules [5, 6, 8, 12, 18]. Especially, from late 90s' a new trend emerges which adopts frequent closed itemsets to extract association rules. The notion of closed frequent itemset has its origins in the mathematical theory of Formal Concept Analysis (FCA) introduced in the early 80s'[7, 19]. FCA has over the years grown to a powerful theory for data analysis, information retrieval, and knowledge discovery. Its adaptation to the association rule mining has been studied since late of 90s' [14, 16, 20]. The use of frequent closed itemsets presents a clear promise to reduce the number of extracted rules and also provides a concise and lossless representation of association rules such as the non-redundant association rules discussed in [15, 21] and the representative association rules discussed in [10]. However, the redundancy still remains in the extracted rules using the closure based approaches.

In this paper, we propose a condensed association rule basis from which more concise non-redundant rules can be extracted. We adopt a similar definition of non-redundant association rules that have minimal antecedents and maximal consequents as defined by Pasquier et al. [15]. However, our definition relaxes the requirements to redundancy, thus more redundant rules can be eliminated. The certainty factor (CF) is an important and popular used measure of belief to inference rules [17]. We prove that the redundant rules eliminated by our approach have less or equal CF belief values than that of their corresponding non-redundant rules which are extracted, and thus the elimination of such rules will not reduce the belief to the extracted rules. Moreover, we prove that the Reliable exact basis is a lossless representation of exact association rules since all exact association rules can be deduced from the Reliable exact basis.

The paper is organized as follows. Section 2 briefly discusses some related work. The basic concepts of association rule mining are given in Section 3. In Section 4, we firstly propose a definition to redundancy and then introduce a concise association rule basis for extracting non-redundant rules. Experimental results are given in Section 5. Finally, Section 6 concludes the paper.

2. RELATED WORK

One approach on addressing the quality of association rules is to apply constraints to generate only those association rules that are interesting to users based on some constraints. [12] and [18] proposed some algorithms that incorporate item constraints to the process of generating frequent itemsets. Also some works have been done on measuring association rules with interestingness parameters [6]. These approaches focus on pruning the association rules to get more general or informative association rules based on interestingness parameters. The approach proposed in [4] integrates various constraints into mining process including consequent constraint and minimal improvement constraint as well. The consequent constraint is used to restrict rules with certain consequent specified by the user, while minimal improvement constraint is used to simplify the antecedents of rules based on item's contribution to the confidence and therefore prune association rules that have more specific antecedent but do not make more contribution to the confidence. Another approach is to use a taxonomy of items to extract generalized association rules [8], i.e., to generate rules between itemsets that belong to different abstract levels in the taxonomy, especially between high abstract levels aiming at reducing the number of extracted rules.

The approaches mentioned above aim at reducing the number of extracted rules and also improving the"usefulness" of the rules, but eliminating redundancy of rules is not a focus. The approaches proposed in [20] and [15] focus on extracting non-redundant rules. Both of them make use of the closure of the Galois connection [7] to extract non-redundant rules from frequent closed itemsets instead of from frequent itemsets. One difference between the two approaches is the definition of redundancy. The approach proposed in [20] extracts the rules with shorter antecedent and shorter consequent as well among rules which have the same confidence, while the method proposed in [15] defines that the non-redundant rules are those which have minimal antecedents and maximal consequents. Our definition to non-redundant rules is similar to that of [15]. However, the requirement to redundancy is relaxed, and the less requirement makes more rules to be considered redundant and thus eliminated. Most importantly, we prove that the elimination of such redundant rules does not reduce the belief to the extracted rules.

3. ASSOCIATION RULES

The classical definition of association rules was firstly given in [1]. Let $I = \{I_1, I_2, \ldots, I_m\}$ be a set of m distinct items, t be a transaction that contains a set of items such that $t \subseteq I$, T be a database containing different identifiable transactions. An association rule is an implication in the form of $X \Rightarrow Y$, where $X, Y \subset I$ are sets of items called itemsets, and $X \bigcap Y = \phi$. X is called antecedent and Y is called consequent, the rule means that X implies Y. Various metrics describe the utility of an association rule. The most common ones are the percentage of all transactions containing $X \cup Y$ which is called the support, and the percentage of transactions containing $X \cup Y$ among transactions containing X which is called confidence of the rule. Association rule mining is to find out association rules that satisfy the predefined minimum support and confidence from a given database. The problem is usually decomposed into two subproblems. One is to find those itemsets whose occurrences exceed a predefined threshold in the database, those itemsets are called frequent itemsets. The second subproblem is to generate association rules from those frequent itemsets with the constraints of minimal confidence and support. The frequent itemsets shown in Table 2 are generated from the simple dataset depicted in Table 1 which is the example dataset used in [15]. Table 3 displays all the 50 association rules extracted from these frequent itemsets with the minimum confidence and support being set to 2/6.

Table 1: A simple database

Transaction ID	Items
1	A C D
2	B C E
3	A B C E
4	B E
5	A B C E
6	B C E

Table 2: Frequent itemsets with supports

1-itemsets	2-itemsets	3-itemsets	4-itemsets
$\{A\}$ 1/2	$\{AB\}$ 1/3	$\{ABC\}$ 1/3	$\{ABCE\}$ 1/3
$\{B\}$ 5/6	$\{AC\}$ 1/2	$\{ABE\}$ 1/3	
$\{C\}$ 5/6	$\{AE\}$ 1/3	$\{ACE\}$ 1/3	
$\{E\}$ 5/6	$\{BC\}$ 2/3	$\{BCE\}$ 2/3	
	$\{BE\}$ 5/6		
	$\{CE\}$ 2/3		

3.1 Closed Itemsets and Generators

The definition of closed itemsets comes from the closure operation of the Galois connection [7]. Let I denote the set of items and T denote the set of transactions, 2^I and 2^T are the power set of I and T, respectively. $\forall i \in I$ and $\forall t \in T$, if item i appears in transaction t, then i and t has a binary relation δ denoted as $i\delta t$. The Galois connection of the binary relation is defined by the following mappings where $X \subseteq I$, $Y \subseteq T$:

$$\tau : 2^I \rightarrow 2^T, \tau(X) = \{t \in T | \forall i \in X, i\delta t\} \quad (1)$$

$$\gamma : 2^T \rightarrow 2^I, \gamma(Y) = \{i \in I | \forall t \in Y, i\delta t\} \quad (2)$$

$\tau(X)$ represents a set of transactions each of which contains X. $\tau(X)$ is called the transaction mapping of X. $\gamma(Y)$ represents a set of items which are the common items among the transactions in Y. $\gamma(Y)$ is called the item mapping of Y.

Table 3: Association rules with support and confidence

No.	Rules (supp, conf)	No.	Rules (supp, conf)
1	$A \Rightarrow B$ (1/3, 2/3)	26	$AB \Rightarrow C$ (1/3, 1)
2	$A \Rightarrow C$ (1/2, 1)	27	$AC \Rightarrow B$ (1/3, 2/3)
3	$A \Rightarrow E$ (1/3, 2/3)	28	$AC \Rightarrow E$ (1/3, 2/3)
4	$B \Rightarrow A$ (1/3, 2/5)	29	$AE \Rightarrow C$ (1/3,1)
5	$B \Rightarrow C$ (2/3, 4/5)	30	$AE \Rightarrow B$ (1/3,1)
6	$B \Rightarrow E$ (5/6, 1)	31	$BC \Rightarrow A$ (1/3,1/2)
7	$C \Rightarrow A$ (1/2, 3/5)	32	$BC \Rightarrow E$ (2/3,1)
8	$C \Rightarrow B$ (2/3, 4/5)	33	$BE \Rightarrow A$ (2/3, 2/5)
9	$C \Rightarrow E$ (2/3, 4/5)	34	$BE \Rightarrow C$ (2/3, 4/5)
10	$E \Rightarrow A$ (1/3, 2/5)	35	$CE \Rightarrow B$ (2/3,1)
11	$E \Rightarrow B$ (5/6, 1)	36	$CE \Rightarrow A$ (1/3,1/2)
12	$E \Rightarrow C$ (2/3, 4/5)	37	$A \Rightarrow BCE$ (1/3, 2/3)
13	$A \Rightarrow BE$ (1/3, 2/3)	38	$B \Rightarrow ACE$ (1/3, 2/5)
14	$A \Rightarrow CE$ (1/3, 2/3)	39	$C \Rightarrow ABE$ (1/3, 2/5)
15	$A \Rightarrow BC$ (1/3, 2/3)	40	$E \Rightarrow ABC$ (1/3, 2/5)
16	$B \Rightarrow AE$ (1/3, 2/5)	41	$AB \Rightarrow CE$ (1/3, 1)
17	$B \Rightarrow CE$ (2/3, 4/5)	42	$AC \Rightarrow BE$ (1/3, 2/3)
18	$B \Rightarrow AC$ (1/3, 2/5)	43	$AE \Rightarrow BC$ (1/3, 1)
19	$C \Rightarrow AB$ (1/3, 2/5)	44	$BC \Rightarrow AE$ (1/3, 1/2)
20	$C \Rightarrow BE$ (2/3, 4/5)	45	$BE \Rightarrow AC$ (1/3, 2/5)
21	$C \Rightarrow AE$ (1/3, 2/5)	46	$CE \Rightarrow AB$ (1/3, 1/2)
22	$E \Rightarrow AC$ (1/3, 2/5)	47	$ABC \Rightarrow E$ (1/3, 1)
23	$E \Rightarrow BC$ (2/3, 4/5)	48	$ABE \Rightarrow C$ (1/3, 1)
24	$E \Rightarrow AB$ (1/3, 2/5)	49	$ACE \Rightarrow B$ (1/3, 1)
25	$AB \Rightarrow E$ (1/3, 1)	50	$BCE \Rightarrow A$ (1/3, 1/2)

Table 4: Closed Itemsets and Minimal Generators

Closed itemsets	Minimal Generators
C	
AC	A
BE	B, E
BCE	BC, CE
ABCE	AB, AE

The composition $\gamma \circ \tau$ is called closure operator on itemsets. $\gamma \circ \tau(X)$, which is called the closure of X, gives the common items among the transactions each of which contains X.

Definition 1. (Frequent Closed Itemsets) Let X be a subset of I. X is a frequent closed itemset iff $\gamma \circ \tau(X) = X$ and X is a frequent itemset.

Definition 2. (Generators) An itemset $g \in 2^I$ is a generator of a closed itemset $c \in 2^I$ iff $c = \gamma \circ \tau(g)$ and $g \subset \gamma \circ \tau(g)$. g is said a minimal generator of the closed itemset set c if $\nexists g' \subset g$ such that $\gamma \circ \tau(g') = c$.

Definition 3. (Supports) The support of an itemset X, denoted as *supp*(X), is the size of $\tau(X)$, i.e., *supp*$(X) = |\tau(X)|$.

For the example dataset given in Table 1, the closed itemsets and their minimal generators (conf = 2/6) are given in Table 4. From Definition 2, we can get that $g \subset c$ is true for any generator g and its closed itemset c. The Galois connection satisfies the following properties[7].

PROPERTY 1. *Let X, $c \in 2^I$. If c is the closed itemset of X, then supp(X) = supp(c),*

PROPERTY 2. *Let X, X_1, $X_2 \in 2^I$ and Y, Y_1, $Y_2 \in 2^T$.*

1. $X_1 \subseteq X_2 \Longrightarrow \tau(X_1) \supseteq \tau(X_2)$

2. $Y_1 \subseteq Y_2 \Longrightarrow \gamma(Y_1) \supseteq \gamma(Y_2)$

3. $X \subseteq \gamma \circ \tau(X)$ *and* $Y \subseteq \tau \circ \gamma(Y)$

4. $\tau(X_1 \cup X_2) = \tau(X_1) \cap \tau(X_2)$

5. $\tau(X_1 \cap X_2) = \tau(X_1) \cup \tau(X_2)$

PROPERTY 3. *If $X \subseteq I$ is a minimal generator of a closed itemset, then all subsets of X are minimal generators.*

3.2 Certainty Factors

The basic principles of the certainty factor theory were first introduced in MYCIN [17], an expert system for the diagnosis and therapy of blood infections, to express how accurate and truthful a rule is and how reliable the antecedent of the rule is. The certainty factor theory is based on two functions: measure of belief $MB(X, Y)$ and measure of disbelief $MD(X, Y)$ for a rule $X \Rightarrow Y$. These functions indicate, respectively, the degree of the belief that the consequent Y would be increased if antecedent X was observed, and the degree of the disbelief that the consequent Y would be increased by observing the same antecedent X. In this paper, the following measurements [11] are adopted to measure belief and disbelief:

$$MB(X,Y) = \begin{cases} 1 & P(Y) = 1 \\ 0 & P(Y/X) \leq P(Y) \\ \frac{P(Y/X) - P(Y)}{1 - P(Y)} & otherwise \end{cases} \quad (3)$$

$$MD(X,Y) = \begin{cases} 1 & P(Y) = 0 \\ 0 & P(Y/X) \geq P(Y) \\ \frac{P(Y) - P(Y/X)}{P(Y)} & otherwise \end{cases} \quad (4)$$

where, in the context of association rules, $P(Y/X)$ and $P(Y)$ are the confidence of the rule and the support of the consequent, respectively. The values of $MB(X, Y)$ and $MD(X, Y)$ range between 0 and 1 measuring the strength of belief or disbelief in consequent Y given antecedent X. $MB(X, Y)$ weighs how much the antecedent X increases the possibility of Y occurring. If the antecedent is very weak to $P(Y)$, then $P(Y/X) - P(Y)$ is almost zero, and the uncertainty of Y remains about the same. On the other hand, if the antecedent completely support the consequent, then $P(Y/X) - P(Y)$ will equal 1 thus $MB(X, Y)$ will be 1. $MD(X, Y)$=1 indicates that the antecedent completely denies the consequent thus the disbelief in the rule reaches its highest value. The total strength of belief or disbelief in the association captured by the rule is measured by the certainty factor which is defined as follows:

$$CF(X,Y) = MB(X,Y) - MD(X,Y) \quad (5)$$

The value of a certainty factor is between 1 and -1. Negative values represent cases where the antecedent is against the consequent; positive values represent that the antecedent supports the consequent; while CF=0 means that the antecedent does not influence the belief to Y. Obviously, association rules with high CF values are more useful and important since they represent strong positive associations between antecedents and consequents. Indeed, the essential aim of association rule mining is to discover strong positive associations from large amount of data. Therefore, we propose in this paper that the certainty factors can be considered in determining useful association rules.

4. REPRESENTATIONS OF ASSOCIATION RULES

It is widely recognized that frequent itemset based association rule mining suffers from the generation of very large number of association rules. Recent studies have shown that all the information contained in frequent itemsets can be sufficiently captured by a set of closed itemsets with a much smaller size [15, 20]. Using closed itemsets and generators to extract association rules can greatly reduce the number of extracted rules without losing the information captured in the original rules. However, redundancy still exists in the extracted association rules using the closed itemsets based approach. In this section, we first discuss the redundancy in association rules and then describe a condensed association rule basis by which more concise non-redundant association rules can be derived.

4.1 Redundancy in Association Rules

The rules in Table 3 are considered useful based on the predefined minimum support and confidence. However, some of the rules actually do not contribute new information to the rule set. For example, rules 1, 13, 14, and 15 can be derived from rule 37 and all these rules have the same confidence. Therefore, removing rules 1, 13, 14, and 15 won't change the power of the rule set. All these rules have the same antecedent and identical confidence, while rule 37 has the longest consequent which is a superset of the consequent of other rules and thus brings more information than the other rules do. Therefore, rules 1, 13, 14, and 15 are considered redundant to rule 37. On the other hand, the consequent of rules 25, 32, and 47 is the same as the consequent of rule 6, which means that these rules can conclude the same result. However, rules 25, 32, and 47 have a larger antecedent to be satisfied than the antecedent that rule 6 has. That means, rules 25, 32, and 47 do not bring more information but require more in order to be fired. In this case, rules 25, 32, and 47 are considered redundant to rule 6. The following definition defines such kind of redundant rules.

Definition 4. (Redundant rules) Let $X \Rightarrow Y$ and $X' \Rightarrow Y'$ be two association rules which have the same confidence. $X \Rightarrow Y$ is said a redundant rule to $X' \Rightarrow Y'$ if $X' \subseteq X$ and $Y \subseteq Y'$.

Based on Definition 4, for an association rule $X \Rightarrow Y$, if there does not exist any other rule $X' \Rightarrow Y'$ such that the confidence of $X' \Rightarrow Y'$ is the same as the confidence of $X \Rightarrow Y$, $X' \subseteq X$ and $Y \subseteq Y'$, then $X \Rightarrow Y$ is non-redundant. Definition 4 is similar to Pasquier's definition of min-max association rules [15], but has less requirement to redundancy. In [15], a rule is redundant to another rule if the rule has longer antecedent, shorter consequent, identical confidence and identical support with the other rule, while Definition 4 requires only identical confidence. Theorem 1 below states that the CF value of a redundant rule defined by Definition 4 must be less than the CF value of its corresponding non-redundant rules no matter what their support values are.

Theorem 1. *Let $X \Rightarrow Y$ and $X' \Rightarrow Y'$ be two association rules. If $Y' \subseteq Y$, and $P(Y/X) = P(Y'/X')$ (i.e., they have the same confidence), then $CF(X,Y) \geq CF(X',Y')$.*

Proof. From Equation (5) we have
$CF(X,Y)$ - $CF(X',Y')$
$= MB(X,Y)$-$MB(X',Y')$+$MD(X',Y')$-$MD(X,Y)$

1. Assuming that $P(Y/X) \geq P(Y)$. In this case, $MD(X',Y')$-$MD(X,Y) = 0$. To prove the theorem, we need to prove that $MB(X,Y)$ - $MB(X',Y') \geq 0$. From Equation (3), we have:
   $MB(X,Y)$ - $MB(X',Y') = \frac{P(Y/X)-P(Y)}{1-P(Y)} - \frac{P(Y'/X')-P(Y')}{1-P(Y')}$
   $= \frac{(P(Y/X)-P(Y))(1-P(Y'))-(P(Y'/X')-P(Y'))(1-P(Y))}{(1-P(Y))(1-P(Y'))}$
   $= \frac{P(Y')-P(Y)-P(Y/X)(P(Y')-P(Y))}{(1-P(Y))(1-P(Y'))}$
   $= \frac{(P(Y')-P(Y))(1-P(Y/X))}{(1-P(Y))(1-P(Y'))}$
   From condition $Y' \subseteq Y$, we have $P(Y) \leq P(Y')$. Therefore, $P(Y') - P(Y) \geq 0$ which makes
   $\frac{(P(Y')-P(Y))(1-P(Y/X))}{(1-P(Y))(1-P(Y'))} \geq 0$.
   Hence, $MB(X,Y)$ - $MB(X',Y') \geq 0$

2. Assuming that $P(Y/X) \leq P(Y)$. In this case, $MB(X,Y)$-$MB(X',Y') = 0$. To prove the theorem, we need to prove that $MD(X',Y')$-$MD(X,Y) \geq 0$. From Equation (4), we have
   $MD(X',Y')$-$MD(X,Y) = \frac{P(Y')-P(Y'/X')}{P(Y')} - \frac{P(Y)-P(Y/X)}{P(Y)}$
   Similar to case (1), after expanding the above expression and eliminating identical dual terms, we have
   $MD(X',Y')$-$MD(X,Y) = \frac{P(Y/X)(P(Y')-P(Y))}{P(Y)P(Y')}$.
   Again, since $P(Y) \leq P(Y')$, we get
   $MD(X',Y')$-$MD(X,Y) \geq 0$.
   Combining the results of the above two cases, we have
   $CF(X,Y)$ - $CF(X',Y') \geq 0$, hence
   $CF(X,Y) \geq CF(X',Y')$

□

According to Theorem 1, the CF value of a redundant rule defined by Definition 4 is not higher than that of its corresponding non-redundant rule and thus the elimination of such redundant rules won't reduce the belief to the extracted non-redundant rules.

4.2 Concise Association Rule Basis

The use of frequent closed itemsets to generate association rules greatly reduces the number of extracted rules. However, redundancy still exits. Pasquier et al. [15] proposed two condensed association bases to represent non-redundant association rules, which are defined as follows:

Definition 5. (Min-max Approximate Basis) Let C be the set of frequent closed itemsets and G be the set of minimal generators of the frequent closed itemsets in C. The min-max approximate basis is:

$$MinMaxApprox = \{g \Rightarrow (c \backslash g) | c \in C, g \in G, \gamma \circ \tau(g) \subset c\}$$

Definition 6. (Min-max Exact Basis) Let C be the set of frequent closed itemsets. For each frequent closed itemset c, let G_c be the set of minimal generators of c. The min-max exact basis is:

$$MinMaxExact = \{g \Rightarrow (c \backslash g) | c \in C, g \in G_c, g \neq c\}$$

Table 5: Non-redundant Exact Rules Extracted From Min-max Exact Basis

No.	Non-redundant Rules	Supp	Conf
2	A⇒C	1/2	1
6	B⇒E	5/6	1
11	E⇒B	5/6	1
32	BC⇒E	2/3	1
35	CE⇒B	2/3	1
41	AB⇒CE	1/3	1
43	AE⇒BC	1/3	1

Table 6: Non-redundant Approximate Rules Extracted From Min-max Approximate Basis

No.	Non-redundant Rules	Supp	Conf
7	C⇒A	1/2	3/5
17	B⇒CE	2/3	4/5
20	C⇒BE	2/3	4/5
23	E⇒BC	2/3	4/5
37	A⇒BCE	1/3	2/3
38	B⇒ACE	1/3	2/5
39	C⇒ABE	1/3	2/5
40	E⇒ABC	1/3	2/5
44	BC⇒AE	1/3	1/2
46	CE⇒AB	1/3	1/2

The non-redundant association rules with confidence less than 1 can be derived from the Min-max approximate basis and the non-redundant association rules with confidence equal to 1 can be derived from the Min-max exact basis. Table 5 and Table 6 display the rules extracted from the min-max bases. These rules are considered non-redundant in terms of the redundancy definition given in [15]. However, under Definition 4, some rules extracted from the min-max exact basis are redundant such as rules 32 and 35 in Table 5. Rule 32 is redundant to rule 6 and rule 35 is redundant to rule 11. We propose a more concise exact association rule basis called Reliable exact basis as defined in Definition 7. Using the Reliable exact basis, more useless rules can be eliminated.

Definition 7. (Reliable Exact Basis) Let C be the set of frequent closed itemsets. For each frequent closed itemset c, let G_c be the set of minimal generators of c. The Reliable exact basis is:

$$ReliableExact = \{g \Rightarrow (c\backslash g) | c \in C, g \in G_c, \neg(g \supseteq ((c\backslash c') \cup g')),\ where\ \ c' \in C, c' \subset c, g' \in G_{c'}\}$$

The correctness of the above definition can be proved by the following theorems and properties.

Lemma 1. *Let C be the set of frequent closed itemsets, $c \in C$ be a frequent closed itemset, G_c be the set of c's minimal generators. If $\exists c' \in C$, $\exists g' \in G_{c'}$, $c \nsubseteq c'$, and $g \supseteq ((c\backslash c') \cup g')$, then $g \Rightarrow c\backslash g$ is redundant to $g' \Rightarrow c'\backslash g'$.*

Proof. Let $A = c\backslash c'$ so that $c \subseteq A \cup c'$ and $A \cap c' = \phi$. Therefore, we have $c\backslash((c\backslash c') \cup g') \subseteq (A \cup c')\backslash(A \cup g')$. Since $g' \subset c'$ and $A \cap c' = \phi$, then $A \cap g' = \phi$. So, $c\backslash((c\backslash c') \cup g') \subseteq (A \cup c')\backslash(A \cup g') = ((A \cup c')\backslash A)\backslash g') = c'\backslash g'$. That is, $c\backslash((c\backslash c') \cup g') \subseteq c'\backslash g'$. Because $g \supseteq ((c\backslash c') \cup g')$, we have $c\backslash g \subseteq c\backslash((c\backslash c') \cup g') \subseteq c'\backslash g'$, hence, $c\backslash g \subseteq c'\backslash g'$. Since $g \supseteq ((c\backslash c') \cup g')$, $c\backslash c' \neq \emptyset$, and $(c\backslash c') \cap g' = \emptyset$, thus we have $g \supset g'$. From $c\backslash g \subseteq c'\backslash g'$ and $g \supset g'$, we can conclude that $g \Rightarrow c\backslash g$ is redundant to $g' \Rightarrow c'\backslash g'$. □

According to Modus tolen inference rule, i.e., if the consequent of an implication is false, the antecedent of the rule must be false, from Lemma 1, we get the following corollary:

Corollary 1. *Let C be the set of frequent closed itemsets, $c \in C$ be a frequent itemset, G_c be the set of c's minimal generators. If $g \Rightarrow c\backslash g$ is a non-redundant rule, then $\forall c' \in C$, $\forall g' \in G_{c'}$, we have $\neg(g \supseteq ((c\backslash c') \cup g'))$.*

Theorem 2. *Let C be the set of frequent closed itemsets, $c \in C$ be a frequent closed itemset, G_c be the set of c's minimal generators. $g \Rightarrow c\backslash g$ is a non-redundant rule iff $\forall c' \in C$, $\forall g' \in G_{c'}$, $\neg(g \supseteq ((c\backslash c') \cup g'))$.*

Proof.

1. $\Longrightarrow$. The proof follows the conclusion of Corollary 1.

2. $\Longleftarrow$. From $\neg(g \supseteq ((c\backslash c') \cup g'))$, we get $g \subset (c\backslash c') \cup g'$, or $g \cap ((c\backslash c') \cup g') = \emptyset$, or $(g \cap ((c\backslash c') \cup g') \subset ((c\backslash c') \cup g')) \wedge (g \cap ((c\backslash c') \cup g') \subset g)$.

   (1). In the case that $g \subset (c\backslash c') \cup g'$ is true, assuming that $g \Rightarrow c\backslash g$ is redundant, then we get, $\exists c' \in C$, $\exists g' \in G_{c'}$ such that $g' \subseteq g$ and $c'\backslash g' \supseteq c\backslash g$.
   From $c'\backslash g' \supseteq c\backslash g$ and $g' \subseteq c'$, we have $c' \supseteq c'\backslash g' \supseteq c\backslash g$, i.e., $c' \supseteq c\backslash g$. Since $g \subset c$, obviously we have $c = (c\backslash g) \cup g$ and $(c\backslash g) \cap g = \phi$; also $(c\backslash c') \cup c' \supseteq c$ and $(c\backslash c') \cap c' = \phi$ are true. Therefore, we have $(c\backslash c') \cup c' \supseteq c = (c\backslash g) \cup g$, i.e.:

   $$(c\backslash c') \cup c' \supseteq (c\backslash g) \cup g \qquad (a)$$

   Because $c' \supseteq c\backslash g$, $(c\backslash c') \cap c' = \phi$ and $(c\backslash g) \cap g = \phi$, after c' being removed from the left side of (a) and $c\backslash g$ being removed from the right side of (a), the formula (a) becomes $c\backslash c' \subseteq g$. From $g' \subseteq g$, we get $(c\backslash c') \cup g' \subseteq g \cup g' = g$, i.e., $(c\backslash c') \cup g' \subseteq g$ which contradicts to $(c\backslash c') \cup g' \supset g$.
   Therefore, the assumption is false, i.e., $g \Rightarrow c\backslash g$ is non-redundant.

   (2). In the case that $g \cap ((c\backslash c') \cup g') = \emptyset$ is true, $g \cap g' = \emptyset$, thus $g \supset g'$ is always false. Therefore, $g \Rightarrow c\backslash g$ can't be redundant to $g' \Rightarrow c'\backslash g'$.

   (3). In the case that $(g \cap ((c\backslash c') \cup g') \subset ((c\backslash c') \cup g')) \wedge (g \cap ((c\backslash c') \cup g') \subset g)$ is true, there must exist some x such that $x \in c\backslash c'$ and $x \notin g$ or $x \in g'$ and $x \notin g$. The former will make $(c\backslash g) \subset (c'\backslash g')$ false and the latter will make $g \supset g'$ false. Therefore, $g \Rightarrow c\backslash g$ will never be redundant to $g' \Rightarrow c'\backslash g'$

□

The following property states that the generator of a closed itemset won't be larger than or equal to the generator of its super closed itemset. Therefore, the rules generated from a closed itemset won't be redundant to the rules generated

Table 7: Non-redundant Exact Rules Extracted From Reliable Exact Basis

No.	Non-redundant Rules	Supp	Conf
2	A⇒C	1/2	1
6	B⇒E	5/6	1
11	E⇒B	5/6	1
41	AB⇒CE	1/3	1
43	AE⇒BC	1/3	1

from its super closed itemset. Thus when calculating non-redundant exact rules from a closed itemset c using the Reliable Exact Basis, only sub closed itemsets of c need to be checked. This property is reflected in the definition of Reliable Exact Basis where only subsets $c' \subset c$ are checked.

PROPERTY 4. *Let g and g' be minimal generators of c and c', respectively, c and c' be closed itemsets, then $c \subset c' \Rightarrow \neg(g \supseteq g')$.*

PROOF. Assume that $g \supseteq g'$. From Property 2-(1) and Property 2-(2), we get
$g \ \supseteq g' \Rightarrow c \supseteq c'$.
Negating both sides of the above implication by using Modus tolen inference rule, we have $\neg(c \supseteq c') \Rightarrow \neg(g \supseteq g')$. That is, $(c \subset c') \vee (c \cap c' = \emptyset) \vee ((c \cap c' \neq \emptyset) \wedge (c \not\subset c')) \Rightarrow \neg(g \supseteq g')$. Because $(c \subset c')$, $c \cap c' = \emptyset$, and $(c \cap c' \neq \emptyset) \wedge (c \not\subset c')$ are exclusive events, they can't be true simultaneously. Therefore we have $(c \subset c') \Rightarrow \neg(g \supseteq g')$. □

The generic representation resulting from coupling the Reliable Exact Basis with the Min-max Approximate Basis defines a more concise set of association rules which are non-redundant, sound and lossless. The algorithm to extract non-redundant exact rules based on the Reliable Exact Basis is given below (some notations are explained in Table 8):

```
ALGORITHM 1. ReliableExactRule(Closure)
Input: Closure: a set of frequent closed itemsets
Output: A set of non-redundant exact rules.
exactRules := $\emptyset$
for each $c \in$ Closure
  for each $g \in G.c$
    if $\forall c' \in$ Closure such that $c' \subset c$ and $\forall g' \in G.c'$
      we have $\neg(g \supseteq ((c \backslash c') \cup g'))$
    then exactRules := exactRules $\cup \{g \Rightarrow (c \backslash g)\}$
  end
end
Return exactRules
```

For the example mentioned in Section 3, the non-redundant exact rules extracted from the Reliable exact basis are given in Table 7. Even rules 32 and 35 have a larger antecedent than the antecedent of rule 6 and rule 11, respectively, they are still considered non-redundant using the Min-max exact basis proposed in [15] because their supports are different. However, the two rules, 32 and 35, are considered redundant to rules 6 and 11 respectively based on our Reliable exact basis and thus eliminated.

4.3 Deriving All Exact Rules from the Reliable Exact Basis

Table 8: Some Notations (X is an itemset)

Notation	Meaning
$X.t$	transaction mapping of X.
$G._X$	set of minimal generators of X
$X.supp$	support of X
$X.conf$	confidence of X
$X.closure$	closed itemset of X
$minconf$	user specified minimal confidence
$minsupp$	user specified minimal support threshold

Pasquier et al. have proved that all exact association rules can be deduced from the Min-max exact basis [15]. The algorithm to deduce all exact rules from the Min-max exact basis is described below in Algorithm 2. In this section, we provide a modified algorithm that can deduce all exact rules from the Reliable exact basis. This means that the Reliable exact basis is also a lossless representation of exact association rules.

```
ALGORITHM 2. DeduceFromMinMax(MinMaxExact)
Input: MinMaxExact: Min-max exact basis
Output: AllExact: A set of all exact association rules
AllExact := $\emptyset$
for each rule $(r_1 : a_1 \Rightarrow c_1, r_1.supp) \in MinMaxExact$
  for each subset $c_2 \subset c_1$
    AllExact := AllExact $\cup \{(r_2 : a_1 \Rightarrow c_2, r_1.supp)\}$
    if $(r_3 : a_1 \cup c_2 \Rightarrow c_1 \backslash c_2, r_1.supp) \notin$ AllExact
      then AllExact := AllExact $\cup \{r_3\}$
  end
end
return AllExact
```

As discussed in Section 4.2, the Min-max exact basis is a super set of the Reliable exact basis. Under the redundancy definition given in Definition 4, the Min-max exact basis can still contain redundancy. Based on the definitions 6 and 7, the Min-max exact basis can be described as:

$$\begin{aligned}
&MinMaxExact = \{g \Rightarrow (c\backslash g) | c \in C, g \in G_c, g \neq c\} \\
&= \{g \Rightarrow (c\backslash g) | c \in C, g \in G_c, \\
&\qquad \neg(g \supseteq ((c\backslash c') \cup g'))\ for\ all\ c' \in C, c' \subset c, g' \in G_{c'} \\
&\qquad or\ \ g \supseteq ((c\backslash c') \cup g')\ for\ some\ c' \in C, c' \subset c, g' \in G_{c'}\} \\
&= \{g \Rightarrow (c\backslash g) | c \in C, g \in G_c, \\
&\qquad \neg(g \supseteq ((c\backslash c') \cup g'))\ for\ all\ c' \in C, c' \subset c, g' \in G_{c'}\} \cup \\
&\quad \{g \Rightarrow (c\backslash g) | c \in C, g \in G_c, \\
&\qquad g \supseteq ((c\backslash c') \cup g')\ for\ some\ c' \in C, c' \subset c, g' \in G_{c'}\} \\
&= ReliableExact \cup \\
&\quad \{g \Rightarrow (c\backslash g) | c \in C, g \in G_c, \\
&\qquad g \supseteq ((c\backslash c') \cup g')\ for\ some\ c' \in C, c' \subset c, g' \in G_{c'}\}
\end{aligned}$$

Let

$$\begin{aligned}
&NonReliableExact = \{g \Rightarrow (c\backslash g) | c \in C, g \in G_c, \\
&\quad g \supseteq ((c\backslash c') \cup g')\ for\ some\ c' \in C, c' \subset c, g' \in G_{c'}\} \qquad (6)
\end{aligned}$$

We have

$$MinMaxExact = ReliableExact \cup NonReliableExact \quad (7)$$

The equation 7 shows that the Reliable exact basis is a subset of the Min-max exact basis. Given *ReliableExact*, if we can deduce *NonReliableExact* from *ReliableExact*, then by applying the algorithm *DeriveRuleMinMax* to *ReliableExact* and *NonReliableExact*, we would be able to deduce all exact association rules. The following theorem allow us to generate all NonReliable rules from *ReliableExact*.

Theorem 3. *Let C be the set of frequent closed itemsets. For rules $r_1 : g_1 \Rightarrow c_1 \backslash g_1$ and $r_2 : g_2 \Rightarrow c_2 \backslash g_2$ where c_1, $c_2 \in C$, $g_1 \in G_{c_1}$ and $g_2 \in Gc_2$, r_1 is a NonReliable exact rule iff $c_1 \supset c_2$ and $(g_1 \supseteq (c_1 \backslash c_2) \cup g_2)$.*

Proof.

1. $\Longrightarrow$
   For rules $r_1 : g_1 \Rightarrow c_1 \backslash g_1$ and $r_2 : g_2 \Rightarrow c_2 \backslash g_2$, if $c_1 \supset c_2$, and $g_1 \supseteq (c_1 \backslash c_2) \cup g_2$, then $\neg(g_1 \supseteq (c_1 \backslash c_2) \cup g_2)$ must be false. According to the definition of Reliable exact basis, $r_1 \notin \mathit{ReliableExact}$ must be true. Therefore, $r_1 \in NonReliableExact$ is true.

2. $\Longleftarrow$
   Assuming that $r_1 : g_1 \Rightarrow c_1 \backslash g_1 \in NonReliableExact$. From Equation (6), we immediately get, $g_1 \supseteq ((c_1 \backslash c_2) \cup g_2)$ for some $c_2 \in C, c_2 \subset c_1,$, and $g2 \in G_{c_2}$.

□

According to Theorem 3, for $r_2 : g_2 \Rightarrow c_2 \backslash g_2$ where c_2 is a closed itemset and g_2 is a generator (i.e., r_2 is a rule in $MinMaxExact$), if we can find a super set $c_1 \supset c_2$, and $(g_1 \supseteq (c_1 \backslash c_2) \cup g_2)$, then we can deduce: $r_1 : g_1 \Rightarrow c_1 \backslash g_1$ is a $NonReliable$ basis rule. This means, from a rule in $MinMaxExact$, we might deduce a $NonReliable$ basis rule. Algorithm 3 given below is a modification of Algorithm 2 which takes *ReliableExat* as the initial value for $MinMaxExact$ and generates all $NonReliable$ basis rules from it so that $MinMaxExact$ will be completed progressively during the course. Theorem 3 ensures that we can deduce all $NonReliable$ basis rules. On completion of executing Algorithm 3, $MinMaxExact$ will contains all *ReliableExat* basis rules and all $NonReliable$ basis rules as well.

For Algorithm 3, initially, $MinMaxExact$ is assigned with *ReliableExact*. Steps 3-7 generate exact association rules from a basis rule in current $MinMaxExact$, which performs the same task as what the algorithm 2 does. Steps 8 to 13 deduce possible $NonReliable$ basis rules and add them into the current $MinMaxExact$.

Algorithm 3. **DeduceFromReliable***(ReliableExact)*
Input: *ReliableExact: reliable exact basis*
Output: *AllExact: A set of all exact association rules*

```
1.  AllExact := $\emptyset$, MinMaxExact := ReliableExact
2.  for each rule $(r_1 : a_1 \Rightarrow c_1, r_1.supp) \in MinMaxExact$
3.      for each subset $c_2 \subset c_1$
4.          AllExact := AllExact $\cup \{(r_2 : a_1 \Rightarrow c_2, r_1.supp)\}$
5.          if $(r_3 : a_1 \cup c_2 \Rightarrow c_1 \backslash c_2, r_1.supp) \notin$ AllExact
6.              then AllExact := AllExact $\cup \{r_3\}$
7.      end
8.      for each super set $c_3 \supset (c_1 \cup a_1)$ and
                    $c_3$ is a closed itemset
9.          for each $a_3 \in G_{c_3}$
10.             if $a_3 \supseteq ((c_3 \backslash (c_1 \cup a_1)) \cup a_1)$
11.                 then $MinMaxExact := MinMaxExact \cup$
                                          $\{a_3 \Rightarrow (c_3 \backslash a_3)\}$
12.         end
13.     end
14. end
15. return AllExact
```

5. EXPERIMENTAL RESULTS

We have conducted experiments to evaluate the effectiveness of the proposed Reliable exact basis. This section presents the experimental results.

5.1 Datasets

We used the following three datasets from UCI KDD Archive (http://kdd.ics.uci.edu/). The Mushrooms dataset contains 8,124 records each of which describes the characteristics of one mushroom object. Each mushroom object has 23 attributes some of which are multiple value attributes. After converting the multiple value attributes to binary ones, the number of attributes of each object becomes 126. The Annealing dataset contains 898 annealing instances (objects), each has 38 attributes. After converting multiple value attributes to binary ones, each object has 276 attributes. The Flare2 dataset contains 1,066 solar flare instances each of which represents captured features for one active region on the sun. Each flare instance has 50 attributes after the multiple value attributes are converted to binary attributes. The experiment is to find the associations among attributes for the three datasets.

5.2 Results

The purpose of the experiment is to evaluate how effective the proposed Reliable exact basis eliminates the redundancy. We have proved in Section 4.2 that *ReliableExact* basis is more concise than $MinMaxExact$ basis and thus should contain less number of basis rules (i.e., the non-redundant rules produced by the basis). We have also proved in Section 4.3 that, from *ReliableExact* basis, we can deduce all exact association rules. In this experiment, firstly we confirmed that both $MinMaxExact$ basis and *ReliableExact* basis can deduce all exact rules. For example, when $Minsupp$ is 0.3, both bases produce 2,142 exact rules for the Mushroom dataset as showed in Table 9. Secondly, we tested the reduction ratio between the size of the $MinMaxExact$ basis and the size of the *ReliableExact* basis for different $Minsupp$ settings. For all tests, the $minconf$ was set to 0.5. Table 9, Table 10, and Table 11 present the test results for the three datasets, respectively.

We surprisingly found that the reduction ratio is very high. For instance, when $Minsupp$ was set to 0.5, the $MinMax$ basis contains 44 basis rules for the Annealing dataset as showed in Table 10, while the *Reliable* basis contains only 7 basis rules, the reduction ratio is 84% which is a very significant reduction. In this case, 104 exact rules can be deduced either from $MinMax$ basis or from *Reliable* basis. After having taken a close check of the rules in each of the bases, we found that there indeed exist a great amount of redundancy in the $MinMax$ basis for each of the tests we have conducted. For example, in the case of $Minsupp = 0.5$ for the Annealing dataset, for the rule *carbon-00* $\Rightarrow$ *product-type-C* in *Reliable* Basis, we found that the following 13 rules in $MinMax$ basis are redundant to *carbon-00* $\Rightarrow$ *product-type-C*:

carbon-00,hardness-00 $\Rightarrow$ *product-type-C*
carbon-00,strength-000 $\Rightarrow$ *product-type-C*
carbon-00,bore-0000 $\Rightarrow$ *product-type-C*
carbon-00,class-3 $\Rightarrow$ *product-type-C*
carbon-00,hardness-00,strength-000 $\Rightarrow$ *product-type-C*
carbon-00,hardness-00,bore-0000 $\Rightarrow$ *product-type-C*

Table 9: Number of exact rules (Mushroom dataset)

Minsupp	Exact Rules	MinMax Basis	Reliable Basis	Reduction Ratio
0.3	2,142	453	117	74%
0.4	493	145	51	65%
0.5	161	44	18	60%
0.6	46	20	12	40%
0.7	27	12	6	50%

Table 10: Number of exact rules (Annealing dataset)

Minsupp	Exact Rules	MinMax Basis	Reliable Basis	Reduction Ratio
0.3	650	194	44	77%
0.4	265	89	23	74%
0.5	104	44	7	84%
0.6	44	28	6	78%

carbon-00,hardness-00,class-3 ⇒ *product-type-C*
carbon-00,strength-000,bore-0000 ⇒ *product-type-C*
carbon-00,strength-000,class-3 ⇒ *product-type-C*
carbon-00,bore-0000,class-3 ⇒ *product-type-C*
carbon-00,hardness-00,strength-000,bore-0000 ⇒ *product-type-C*
carbon-00,hardness-00,bore-0000,class-3 ⇒ *product-type-C*
carbon-00,strength-000,bore-0000,class-3 ⇒ *product-type-C*

The 13 rules listed above have the same consequent but a larger antecedent than that of the rule *carbon-00* ⇒ *product-type-C*, their support values are different, but they have exactly the same confidence and the same CF value. In real world problem solving, if we know that *carbon-00* is true, by applying the rule *carbon-00* ⇒ *product-type-C*, we can conclude that *product-type-C* is true. We don't have to know *hardness-00*, *strength-000*, or *bore-0000*, etc. in order to reach this conclusion. That means, all the 13 rules are useless if we have the rule *carbon-00* ⇒ *product-type-C* at hand. By eliminating the redundant rules, the size of the *Reliable* basis is much smaller than that of the *MinMax* basis, but the capacity of solving problems remains the same. This reduction provides a great potential to improve the effectiveness of using the extracted association rules.

6. CONCLUSION

One challenge problem with association rule mining is the redundancy in association rules which greatly affects the use of the extracted rules in solving real world problems. The work presented in this paper aims at improving the quality of association rules by eliminating redundancy. In this paper, we proposed a relaxed definition of redundancy and a concise representation of association rules. We theoretically proved and experimentally confirmed that the proposed Reliable exact basis can dramatically eliminate considerable amount of redundancy. Based on certainty factor theory, we also proved that the elimination of the redundancy using the proposed Reliable exact basis does not affect the belief to the extracted rules. Similar to the Min-max exact basis, the Reliable exact basis is not only a concise but also a lossless representation of exact association rules. From the Reliable exact basis, all exact association rules can be deduced. The work presented here can be extended to approximate rules. The Reliable exact basis is a concise representation for exact rules only. We have found that there is similar redundancy existing in the approximate basis. Our next work is to extend the Reliable exact basis to further eliminate redundancy from approximate rules.

Table 11: Number of exact rules (Flare2 dataset)

Minsupp	Exact Rules	MinMax Basis	Reliable Basis	Reduction Ratio
0.3	957	241	41	83%
0.4	364	154	47	70%
0.5	383	107	20	82%
0.6	230	90	11	88%

7. ACKNOWLEDGMENTS

This paper was partially supported by Grant DP0556455 from the Australian Research Council.

8. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on very large data bases*, pages 487–499, 1994.

[3] R. J. Bayardo. Efficiently mining long patterns from databases. In *Proceedings of the 1998 ACM SIGMOD Conference*, pages 85–93, 1998.

[4] R. J. Bayardo, R. Agrawal, and D. Gunopulos. Constraint-based rule mining in large, dense databases. *Data Mining and Knowledge Discovery*, 4:217–240, 2000.

[5] M. J. A. Berry and G. S. Linoff. *Data Mining Techniques for Marketing, Sales and Customer Support*. John Wiley and Sons, 1997.

[6] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the 1997 ACM SIGMOD Conference*, pages 255–264, 1997.

[7] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag, 1999.

[8] J. Han and Y. Fu. Mining multiple-level association rules in large databases. *IEEE Transactions on Knowledge and Data Engineering*, 11:798–804, 5 2000.

[9] J. Han and J. Pei. Mining frequent patterns by pattern-growth: methodology and implications. *ACM SIGKDD Explorations Newsletter*, 2(2):14–20, 2000.

[10] M. Kryszkiewicz, H. Rybinski, and M. Gajek. Dataless transitions between concise representations of frequent patterns. *Journal of Intelligent Information Systems*, 22(1):41–70, 2004.

[11] K. Ng and B. Abramson. Uncertainty management in expert systems. *IEEE Expert*, 5(2):29–47, 1990.

[12] R. T. Ng, V. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning otimizations of constrained association rules. In *Proceedings of the SIGMOD conference*, pages 13–24, 1998.

[13] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th ICDT conference*, pages 398–416, 1999.

[14] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rultes using closed itemset lattices. *Information Systems*, 24(1):25–46, 1999.

[15] N. Pasquier, R. Taouil, Y. Bastide, G. Stumme, and L. Lakhal. Generating a condensed representation for association rules. *Journal of Intelligent Information Systems*, 24(1):29–60, 2005.

[16] J. Pei, J. Han, and R. Mao. Closet: An efficient algorithm for mining frequent closed itemsets. In *Proceedings of the DMKDWorkshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[17] E. H. Shortliffe and B. G. Buchanan. A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23(3/4):351–379, 1975.

[18] R. Srikant, Q. Vu, and R. Agrawal. Mining association rules with item constraints. In *Proceedings of the KDD Conference*, pages 67–73, 1997.

[19] R. Wille. *Restructuring lattices theory: An approach based on hierarchies of concepts*. I. Rival (editor), Ordered sets. Dordrecht-Boston, 1982.

[20] M. J. Zaki. Generating non-redundent association rules. In *Proceedings of the KDD Conference*, pages 34–43, 2000.

[21] M. J. Zaki. Mining non-redundant association rules. *Data Mining and Knowledge Discovery*, 9:223–248, 2004.

[22] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. In *Proceedings of the 3rd KDD conference*, pages 283–286, 1997.