International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 3– No.9, August 2012 – www.ijais.org

Mining Non- Redundant Frequent Pattern in Taxonomy Datasets using Concept Lattices

R. Vijaya Prakash
Dept. of Informatics
Kakatiya University
Warangal, AP, India

A.Govardhan
Dept. of Comp. Sci. & Eng.
JNT University
Hyderabad, AP, India

SSVN. Sarma
Dept. of Comp. Sci. & Eng.
Vaagdevi College of Eng.
Warangal, AP, India

ABSTRACT
In general frequent itemsets are generated from large data sets by applying various association rule mining algorithms, these produce many redundant frequent itemsets. In this paper we proposed a new framework for Non-redundant frequent itemset generation using closed frequent itemsets without lose of information on Taxonomy Datasets using concept lattices.

General Terms
Frequent Pattern, Association Rules, Lattices.

Keywords
Non Redundant, Frequent Patterns, Concept Lattice, Association Rules, Itemset.

1. INTRODUCTION
Data mining has attracted a great deal of attention in the information industry and in society as a whole in recent years, due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from market analysis, fraud detection, and customer retention, to production control and science exploration.

Frequent pattern mining is an important area of Data mining research. The frequent patterns are patterns (such as itemsets, subsequences, or substructures) that appear in a data set frequently. For example, a set of items, such as milk and bread that appear frequently together in a transaction data set is a *frequent itemset*. A subsequence, such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a *frequent sequential pattern*. A *substructure* can refer to different structural forms, such as subgraphs, subtrees, or sublattices, which may be combined with itemsets or subsequences. If a substructure occurs frequently, it is called a *frequent structured pattern*. Finding such frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data. Moreover, it helps in data classification, clustering, and other data mining tasks as well.

The process of discovering interesting and unexpected rules from large data sets is known as association rule mining. This refers to a very general model that allows relationships to be found between items of a database. An association rule is an *implication* or *if-then-rule* which is supported by data. The association rules problem was first formulated in [1][5] and was called the *market-basket* problem. The initial problem was the following: given a set of items and a large collection of sales records, which consist in a transaction data and the items bought in the transaction, the task is to find relationships between the items contained in the different transactions. A typical association rule resulting from such a study could be "90 percent of all customers who buy bread and butter also buy milk" – which reveals a very important information. Therefore this analysis can provide new insights into customer behavior and can lead to higher profits through better customer relations, customer retention and better product placements.

1.1 Previous Work
Mining of association rules is a field of data mining that has received a lot of attention in recent years. The main association rule mining algorithm, Apriori[1], not only influenced the association rule mining community, but it affected other data mining fields as well. Apriori and all its variants like Partition[6], Pincer-Search[4], Incremental[10], Border algorithm[11] etc. take too much computer time to compute all the frequent itemsets. These algorithms produce redundant association rules. Some Algorithms [2][4] generate maximal frequent itemsets. Maximal itemsets cannot be used for rule generation, since support of subsets is required for confidence computation, this requires one more scan to gather the supports of all subsets, and this still have the problem of redundant association rules. Further, for all these methods, it is not possible to find association rules in dense datasets which may easily have frequent itemsets.

There has been some work in pruning discovered association rules by forming rule covers [8]. However, the problem of constructing a generating set has not been studied previously. The recent work in [3] addresses the problem of mining the most interesting rules. They do not address the issue of rule redundancy, The Pasiquir [12][13] have used closed itemset for association rule mining. However they mainly concentrate on the discovery of frequent closed itemset and do not report any experiments on non-redundant association rule mining.

In this paper, an attempt has been made to compute frequent itemsets by using closed frequent itemsets to remove the redundant itemsets. We use the recently proposed CHARM algorithm [9] for mining all closed frequent itemsets, in a fraction of the time it takes to mine all frequent itemsets using the Apriori [1] method. Our framework builds upon and adapts the work in [7]. However our characterization of the generating set is different, and we also present an experimental verification.

2. ASSOCIATION RULE MINING
Association Rule Mining aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or

other data repositories [5]. The major aim of ARM is to find the set of all subsets of items or attributes that frequently occur in many database records or transactions, and additionally, to extract rules on how a subset of items influences the presence of another subset. ARM algorithms discover high-level prediction rules in the form: IF the conditions of the values of the predicting attributes are true, THEN predict values for some goal attributes.

In general, the association rule is an expression of the form X=>Y, where X is antecedent and Y is consequent. Association rule shows how many times Y has occurred if X has already occurred depending on the support and confidence value.

Support: It is the probability of item or item sets in the given transactional data base:
support(X) = n(X) / n where n is the total number of transactions in the database and n(X) is the
number of transactions that contains the item set X.
Therefore, support (X→Y) = p(XUY).

Frequent itemset: Let A be a set of items, T be the transaction database and *minsup* be the user specified minimum support. An itemset X in A (i.e., X is a subset of A) is said to be a *frequent itemset* in T with respect to *minsup* if $\text{support}(X)_T >$ *minsup*

The problem of mining association rules can be decomposed into two sub-problems:

- Find all itemsset whose support is greater than the user-specified minimum support, *minsup*. Such itemsets are called *frequent itemsets*.
- Use the frequent itemsets to generate the desired rules. The general idea is that if, say ABCD and AB are frequent itemsets, then we can determine if the rule AB=>CD holds by checking the following inequality

support({A,B,C,D}) / support({A,B}) > *minconf*, where the rule holds with confidence *minconf*

To demonstrate the use of the support-confidence framework, we illustrate the process of mining association rules by the following example.

Example 1. Assume that we have a transaction database in a supermarket, as shown in Table 1. There are six transactions in the database with their transaction identifiers (TIDs) ranging from 100 to 600. The universal itemset I ={A, B, C, D, E}, where A, B, C, D and E can be any items in the supermarket.

Table 1. An example transaction database

TID	Items
100	ABDE
200	BCE
300	ABDE
400	ABCE
500	ABCDE
600	BCD

There are totally 2^5(=32) itemsets. {A}, {B}, {C}, {D}, and {E} are all 1-itemsets, {AC} is a 2-itemset, and so on. All Frequent Itemset with min support =50% is

Table 2. Frequent Itemset

Itemsets	Support
B	100%
E, BE	83%
A, C, D, AB, AE, BC, BD, ABE	66%
AD, CE, DE, ABD, ADE, BDE, BCE, ABDE	50%

ABDE, BCE are maximal-by-inclusion frequent itemsets i.e., they are not a subset of any other frequent itemset.

2.1 Generating confident rules

This step is relatively straightforward; rules of the form $X \xrightarrow{P} Y$, Where X, Y are generated frequent itemset and p≥*minconf.* The following table shows the generated confidence rules.

Table 3: Frequent Itemsets

Association Rules	Confidence
A→B, A→E, A→BE, C→B, D→B, E→B	100%
AB→E, AD→B, AD→E, AE→B, CE→B,	100%
DE→A, DE→B, AD→BE, DE→AD, ABD→E,	100%
ADE→B, BDE→A	100%
B→E	83.33%
E→AB, BE→A, E→A	80%
B→AE	66.67

From the above generated frequent itemset ABE can generate 6 possible rules those are $A \xrightarrow{1.0}$ BE, B $\xrightarrow{0.67}$ AE, E $\xrightarrow{0.80}$ AB, , $AB \xrightarrow{1.0}$ E, $AE \xrightarrow{1.0}$ B and BE $\xrightarrow{1.0}$ A

3. CLOSED FREQUENT ITEMSETS

In this section we develop the concept of closed frequent itemsets, and show that this set is necessary and sufficient to capture all the information about frequent itemsets, and has smaller cardinality than the set of all frequent itemsets.

3.1 Partial Order and Lattices

We first introduce some lattice theory concepts, Let P be a set. A *partial order* on P is a binary relation ≤, such that for all x, y, z ε P, the relation is: 1) Reflexive, 2) Anti-Symmetric, 3) Transitive. The set P with the relation ≤ is called an *ordered set*, and it is denoted as a pair (P, ≤). Let (P, ≤) be an ordered set, and let S be a subset of P. An element u ε P is an *upper bound* of S if s ≤ u for all s ε S. An element l ε P is a lower bound of S if s ≥ l for all s ε S. The least upper bound is called **join** of S, and is denoted by ∨S, and the greatest lower bound is called the **meet** of S, and is denoted by ∧S. If S={x,y}, we write x∨y for join and x∧y for meet.

An ordered set (L, ≤) is a lattice, if for any two elements x and y in L the join x∨y and meet x∧y always exist. L is called *complete lattice* if ∨S, ∧S exist for all S⊆L. Any finite lattice is complete, L is called a *join semilattice,* if only join lattice exists. L is called *meet semilattice* if only a meet exists.

Let *P* denote the power set of S. The ordered set (*P*(S),⊆) is a complete lattice, where the meet is intersection and join is union. For example the partial orders (*P*(I), ⊆) is set of all possible itemsets, (*P*(T), ⊆) are the set of all possible transaction are both complete lattices.

The set of all frequent itemsets is only meet semilattice(Fig – 1). For any two itemsets, only their meet is guaranted to be frequent, while their join may or may not be frequent. This follows from the principle of association rule mining that is, if an itemset is frequent, then all its subsets are also frequent. For example AB$\wedge$AD = AB$\cap$AD = A is frequent. For the join, AB$\vee$AD = AB $\cup$AD = ABD is frequent, AB$\cup$CE = ABCE is not frequent.

3.2 Closed Itemsets

Let the binary relation £$\subseteq$ I x T be the input dataset for frequent itemset mining. Let X$\subseteq$I and Y$\subseteq$T, the mappings

t:I → T, t(X) = {y ε T | ∀x ε X, x£y}
i:T → I, t(Y) = {x ε I | ∀y ε Y, x£y}

(P(I), $\subseteq$) and (P(T), $\subseteq$) are power sets of I and T. We denote a X, t(X) pair as X x t(X) and a i(Y), Y pair as i(Y) x Y. The mapping t(X) is the set of all transactions which contains the itemset X, similarly i(Y) is the itemset that is contained in all transactions in Y. For example t(ABE) = 1345, and i(245) = BCE. In terms of individual elements t(X) = $\bigcap_{x \in X} t(x)$ and i(Y) = $\bigcap_{y \in Y} i(y)$. For example t(ABE) = t(A) $\cap$ t(B) $\cap$ t(E) = 1345 $\cap$ 123456 $\cap$ 12345 = 1345.

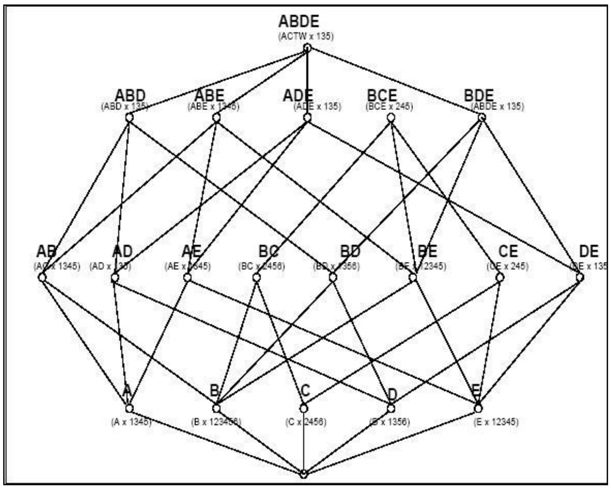

Fig 1. Meet Semi Lattice of Frequent Itemset

Let S be a set. A function c:P(S) → P(S) is a closure on S if, for all X, Y $\subseteq$S, c satisfies the following properties 1) Extension X $\subseteq$ c(X). 2) Monotonicity: if X $\subseteq$ Y, then c(X) $\subseteq$ c(Y). 3) Idem potency: c(c(X)) = c(X). A subset X of S is called closed if c(X) = X

Let X$\subseteq$I and Y$\subseteq$T. Let c_{it}(X) denote the composition of the two mappings i o t(X) = i(t(X). Dually, let c_{ti}(Y) = t o i(Y) = t(i(Y)). Then c_{it}:P(I)→P(I) and c_{ti}:P(T)→P(T) are both closure operators on the itemsets and transaction set.

We define a *closed itemset* as an itemset X that is that same as its closure, i.e X = c_{it}(X). For example the itemset ABE is closed. A *closed transaction set* is a transaction set Y = c_{ti}(Y). For example, the transaction set 1345 is closed. The mappings c_{it} and c_{ti}, being closure operators, satisfy the three properties of extension, monotonicity, and idem potency. We also call the application of i o t or t o i a *round-trip,* starting with an itemset X. For example, let X = AB, then the extension property say that X is a subset of its closure, since c_{it}(AB) = i(t(AB)) = i(1345) = ABE. Since AB ≠ c_{it}(AB) = ABE, we conclude that AB is not closed. On the other hand, the idempotency property say that once we map an itemset to the transaction set that contains it, and then map that transaction set back to the set of items common to all transaction ids in the transaction set, we obtain a closed itemset. After this no matter how many such round-trips we make we cannot extend a closed itemset. For example, after one round-trip for AB we obtain the closed itemset ABE. If we perform another round-trip on ABE, we get c_{it}(ABE) = i(t(ABE)) = i(1345) = ABE.

For any closed itemset X, there exists a closed transaction set given by Y , with the property that Y = t(X) and X = i(Y) i.e for any closed transaction set there exists a closed itemset. We can see that X is closed by the fact that X = i(Y), then plugging Y = t(X), we get X = i(Y) = i(t(X)) = c_{it}(X), thus X is closed. Dually, Y is closed. For example, we have seen above that for the closed itemset ABE the associated closed tidset is 1345. Such a closed itemset and closed tidset pair X x Y is called a *concept*.

A concept X_1 x Y_1 is a *subconcept* of X_2 x Y_2, denoted as X1 x Y1 ≤ X2 x Y2, iff X1 $\subseteq$ X2 (iff Y2 $\subseteq$ Y1). Let β(£) denote the set of all possible concepts in the database. Then the ordered set (β(£),≤) is a complete lattice, called the *Galois* lattice. For example, Figure 2 shows the Galois lattice for our example database, which has a total of 10 concepts. The least element is the concept C x 123456 and the greatest element is the concept ABCDE x 5. Notice that the mappings between the closed pairs of itemsets and transaction sets are anti-isomorphic, i.e., concepts with large cardinality itemsets have small transaction sets, and vice versa.

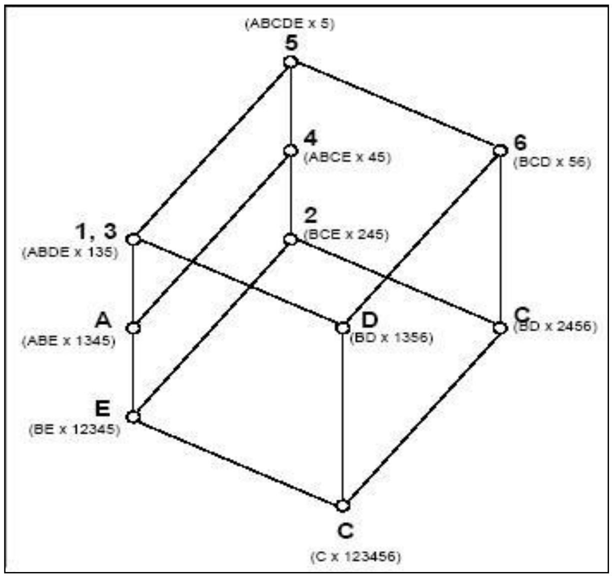

Fig 2: Galois Lattice

The concept generated by a single item *x* ε I is called an *item concept*, and is given as $C_i(x)$ = $c_{it}(x)$ x t(*x*). Similarly, the concept generated by a single transaction *y* ε T is called a *tid concept*, and is given as $C_t(y)$ = i(*y*) x $c_{ti}(y)$. For example, the item concept C_i(A) = i(t(A)) x t(A) = i(1345) x 1345 = ABE x 1345. Further, the tid concept C_t(2) = i(2) x t(i(2)) = BCE x t(BCE) = BCE x 245.

In Figure 2 if we re-label each concept with the item concept or tid concept that it is equivalent to, then we obtain a lattice with *minimal labeling*, with item or tid labels, as shown in the figure in bold letters. Such a re-labeling reduces clutter in the lattice diagram, which provides an excellent way of visualizing the structure of the patterns and relationships that exist between items. We shall see its benefit in the next section when we talk about high confidence rules extraction.

It is easy to reconstruct the concepts from the minimal labeling. For example, consider the tid concept $C_t(2)$ = X x Y. To obtain the closed itemset X, we append all item labels reachable below it. Conversely, to obtain the closed transaction set Y we append all labels reachable above $C_t(2)$. We see that E, C and B are all the labels reachable by a path below it. Thus X = BCE forms the closed itemset. We also see that 4 and 5 are the only labels reachable above $C_t(2)$. Thus Y = 245, giving the concept BCE x 245, which matches the concept shown in the Figure 1.

3.3 Frequent Closed Itemsets vs. Frequent Itemsets

We begin this section by defining the join and meet operation on the concept lattice [6]. The set of all concepts in the database relation £, given by (β(£), ≤) is a complete lattice with join and meet given by

Join: $(X_1 \text{ x } Y_1) \vee (X_2 \text{ x } Y_2) = c_{it}(X_1 \cup X_2) \text{ x } (Y_1 \cup Y_2)$
meet: $(X_1 \text{ x } Y_1) \wedge (X_2 \text{ x } Y_2) = c_{it}(X_1 \cap X_2) \text{ x } (Y_1 \cap Y_2)$

For the join and meet of multiple concepts, we simply take the unions and joins over all of them. For example, consider the join of two concepts, (ABCE x 45) ∨ (BCD x 56) = c_{it}(ABCE ∪ BCD) x (45 ∩ 56) = ABCDE x 5. On the other hand their meet is given as, (ABCE x 45) ∧ (BCD x 56) = (ABCE ∩ BCD) x c_{ti}(45 ∪ 56) = BC x c_{ti}(456) = BC x 2456. Similarly, we can perform multiple concept joins or meets; for example, (BD x 1356) ∨ (BC x 2456) ∨ (BCE x 245) = c_{it}(BD ∪ BC ∪ BCE) x (1356 ∩ 2456 ∩ 245) = c_{it}(BCDE) x 5 = ABCDE x 5.

We define the support of a closed itemset X as the cardinality of the closed tidset Y = t(X), i.e, supp(X) = |Y| = |t(X)|. A closed itemset or a concept is *frequent* if its support is at least *minsup*. Table 6 shows all the frequent concepts with *minsup* = 50% (i.e., with tidset cardinality at least 3). The frequent concepts form a meet-semilattice, where the meet is guaranteed to exist, while the join may not.

All frequent itemsets can be determined by the join operation on the frequent item concepts. For example, since join of item concepts C and D, C_i(C) ∨ C_i(D), doesn't exist, CD is not frequent. On the other hand, C_i(A) ∨ Ci(D) = ABDE x 135, thus AD is frequent. Furthermore, the support of AD is given by the cardinality of the resulting concept's tidset, i.e., supp(AD) = |t(AD)| = |135| = 3.

Theorem: For any itemset X, *its support is equal to the support of its closure, i.e., supp*(X) = supp(c_{it}(X)).

This theorem states that all frequent itemsets are uniquely determined by the frequent closed itemsets or frequent concepts. Furthermore, the set of frequent closed itemsets is bounded above by the set of frequent itemsets, and is typically much smaller, especially for dense datasets. For very sparse datasets, in the worst case, the two sets may be equal. To illustrate the benefits of closed itemset mining, contrast Figure 2, showing the set of all frequent itemsets, with Table 6, showing the set of all closed frequent itemsets. We see that while there are only 7 closed frequent itemsets, in contrast there are 19 frequent itemsets. This example clearly illustrates the benefits of mining the closed frequent itemsets.

Table 4: Frequent Concepts

Tidset	Frequent Concepts
245	BDE
135	ABDE
1345	ABE
12345	BE
1356	BD
2456	BC
123456	B

4. RULE GENERATION

In the last section, we showed that the support of an itemset X equals the support of its closure c_{it}(X). Thus it suffices to consider rules only among the frequent concepts. In other words the rule $X1 \xrightarrow{p} X2$ is exactly the same as the rule $cit(X1) \xrightarrow{p} cit(X2)$, where p is the confidence.
From the concept lattice is it is sufficient to consider rules among adjacent concepts, since other rules can be inferred by transitivity, that is:
Transitivity: *Let* X_1, X_2, X_3 *be frequent closed itemsets, with* $X_1 \subseteq X_2 \subseteq X_3$. *If* $X1 \xrightarrow{p} X2$ *and* $X2 \xrightarrow{p} X3$ *then* $X1 \xrightarrow{p} X3$.
In this paper, we consider two cases of association rules, those with 100% confidence, i.e., with p = 1:0, and those with p < 1:0.

4.1 Rules with 100% confidence

An association rule $X1 \xrightarrow{1.0} X2$ has confidence p = 1:0 if and only if $t(X_1) \subseteq t(X_2)$. i.e all 100% confidence rules are those that are directed from a super-concept (X_1 x $t(X_1)$) to a sub-concept (X_2 x t(X2)). Since it is precisely in these cases that t(X1) ⊆ t(X2) (or X1 ⊆ X2). For example, consider the item concepts Ci(E) = BE x12345 and Ci(B) = B x 123456. The rule $E \xrightarrow{1.0} B$ is a 100% confidence rule. Note that if we take the itemset closure on both sides of the rule, we obtain $BE \xrightarrow{1.0} B$, i.e., a rule between closed itemsets, but since the antecedent and consequent are not disjoint in this case, we prefer to write the rule as $E \xrightarrow{1.0} B$ although both rules are exactly the same.

Table 5: Rules with 100% confidence

Association Rules	Confidence
DE→A, DE→AB, BDE→A	100%
DE→A, DE→AB, BDE→A	100%
A→E, A→BE, AB→E	100%
E→B	100%
D→B	100%
C→B	100%

In the above table, we prefer the rule that is most general. For example, consider the rules $\text{DE} \xrightarrow{1.0} \text{A}$, $DE \xrightarrow{1.0} AB$ and $BDE \xrightarrow{1.0} A$. We prefer the rule $DE \xrightarrow{1.0} A$ A since the latter two are obtained by adding one (or more) items to either the antecedent or consequent of $DE \xrightarrow{1.0} A$. In other words $DE \xrightarrow{1.0} A$ is more general than the latter two rules. In fact, we can say that the addition of C to either the antecedent or the consequent has no effect on the support or confidence of the rule. In this case we also call the other two rules are redundant.

International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 3– No.9, August 2012 – www.ijais.org

Let R_i *stand for a 100% confidence rule* $X_1^i \xrightarrow{p} X_2^i$ *, and let* R = {R_1,R_2, , , R_n} *be a set of rules such that* $I_1 = c_{it}(X_1^i \cup X_2^i)$ *, and* $I_2 = c_{it}(X_2^i)$ *for all rules* R_i*. Then all the rules are equivalent to the 100% confidence rule* $I_1 \xrightarrow{1.0} I_2$ *, and thus are* redundant.

We find that for the first rule that c_{it}(DE∪A) = c_{it}(ADE) = ABDE. Similarly for the other two rules c_{it}(DE∪AB) = c_{it}(ABDE) = ABDE, and c_{it}(BDE∪A) = c_{it}(ABDE) = ABDE. Thus for these three rules we get the closed itemset I_1=ABDE. By the same process we obtain I_2=ABE. All three rules correspond to the edge between the tid concept C_t(1.3) and the item concept C_i(A). Finally $DE \xrightarrow{1.0} A$ is the most general rule and so other are redundant.

A set of such general rules constitutes a *generating set*, i.e., a rule set, from which all other 100% confidence rules can inferred. Note that in this paper we do not address the question of eliminating self redundancy within this generating set, i.e., there may still exist rules in the generating set that can be derived from other rules in the set. In other words we do not claim anything about the minimality of the generating set.

Table 6: generating set with 100% confidence

Frequent Itemset	Confidence
DE→A	100%
A→E	100%
E→B	100%
D→B	100%
C→B	100%

Table 8 shows the generating set, which includes the 5 most general rules $DE \xrightarrow{1.0} A$, $A \xrightarrow{1.0} E$, $E \xrightarrow{1.0} B$, $D \xrightarrow{1.0} B$, $C \xrightarrow{1.0} B$. All other 100% confidence rules can be derived from this generating set by application of simple inference rules. For example, we can obtain the rule $A \xrightarrow{1.0} B$ by transitivity from the two rules $A \xrightarrow{1.0} E$ and $E \xrightarrow{1.0} B$. The rule $CE \xrightarrow{1.0} B$ can be obtained by augmentation of the two rules $E \xrightarrow{1.0} B$ and $C \xrightarrow{1.0} B$, etc. One can easily verify that all the 19 100% confidence rules produced by using frequent itemsets, as shown in Table 3, can be generated from this set of 5 rules, produced using the closed frequent itemsets

4.2 Rules with confidence less than 100%

We now turn to the problem of finding a generating set for frequent itemset with confidence less than 100%. But in this the rules go from sub-concepts to super-concepts.

Let R_i *stand for a 100% confidence rule* $X_1^i \xrightarrow{p} X_2^i$ *, and let* R = {R_1,R_2, , , R_n} *be a set of rules such that* $I_1 = c_{it}(X_1^i)$ *, and* $I_2 = c_{it}(X_1^i \cup X_2^i)$ *for all rules* R_i*. Then all the rules are equivalent to the 100% confidence rule* $I_1 \xrightarrow{p} I_2$*, and thus are* redundant.

The three rules E→A, E→AC and BE→A can be applied to the above theorem, then we get I_1=c_{it}(E) = c_{it}(BE) = BE, and I_2 = c_{it}(E∪A) = c_{it}(E∪AC) = c_{it}(BE∪A) = ABE. The support of the rule is |t(I_1∪I_2)| = |t(ABE)| = 4, and the confidence is given as |t(I_1∪I_2)|/|t(I_1)| = 4/5 = 0.8. similarly we get $B \xrightarrow{0.83} E$ Finally $E \xrightarrow{0.8} A$, $B \xrightarrow{0.83} E$ are the most general rule for less than 100% confidence, the other two are redundant.

Table 7: Generating set with <100% confidence

Frequent Itemset	Confidence
E→A	83.33%
B→E	8.%

By combining the generating set for rules with p = 1.0, shown in Table 8 and the generating set for rules with 1.0 > p ≥ 0.8, shown in Table 9, we obtain a generating set for all association rules with *minsup* =50%, and *minconf* = 80%: { $DE \xrightarrow{1.0} A$, $A \xrightarrow{1.0} E$, $E \xrightarrow{1.0} B$, $D \xrightarrow{1.0} B$, $C \xrightarrow{1.0} B$, $E \xrightarrow{0.8} A$, $B \xrightarrow{0.83} E$ }.
It can be easily verified that all the association rules shown in Table 3, for our example database from Table 1, can be derived from this set. Using the closed itemset approach we produce 7 rules versus the 22 rules produced in traditional association mining. To see the contrast further, consider the set of all possible association rules we can mine. With *minsup* = 50%, the least value of confidence can be 50%. There are 60 possible association rules versus only 13 in the generating set.

5. EXPERIMENTS

All Experiments were performed on a 400MHz Pentium PC with 500MB of memory, running on windows XP and XLMiner. The Experiments are applied on two datasets Chess and Mushroom Datasets obtained from UCI Machine Learning Repository. The Database Characteristics are the Mushroom dataset contains 8125 rows and 23 attributes and census-income contains 32562 rows 15 attributes, the proposed model applied on these data sets to identify the no of frequent itemsets, no of closed frequent itemsets and no. of non redundant frequent itemsets. This is explained in the following tables and figures

Table 7: No. of Frequent, Closed and Redundant Rules

Dataset	Minsupp=80%			Minsupp=70%		
	F	FCI	NR	F	FCI	NR
Mushroom	25	8	6	29	10	7
Census-income	12	10	8	24	18	15

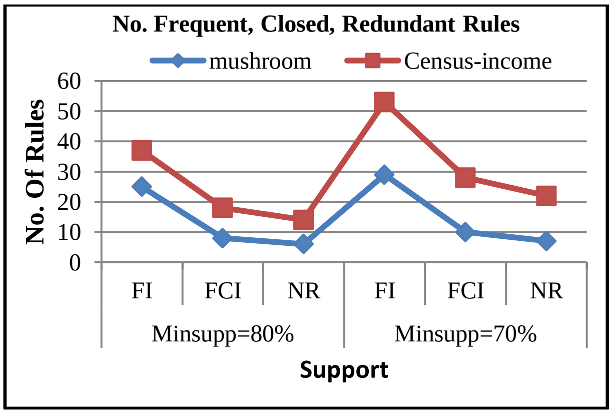

Fig. 3: Rules for Frequent, Closed, Redundant Rules

6. CONCLUSIONS

The traditional Association Rules produces too many rules, in which most of them are redundant. In the given proposed, a framework based on closed frequent itemsets to reduce the rule set, and obtain the strong Non Redundant association rules based on the concept lattice. We can extend this concept on Multilevel Association rules, in which we may get the many redundant rules at different levels, such a redundant rules can be avoided to get quality association rules.

7. REFERENCES

[1] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Inkeri Verkamo, A. 1996. "*Fast discovery of association rules. In Advances in Knowledge Discovery and Data Mining*", U. Fayyad et al. (Eds.), Menlo Park, CA: AAAI Press, pp. 307–328.

[2] Bayardo, R.J. 1998. *"Efficiently mining long patterns from databases*". In ACM SIGMOD Conf. Management of Data.

[3] Brin, S., Motwani, R., Ullman, J., and Tsur, S. 1997. *"Dynamic itemset counting and implication rules for market basket data".* In ACM SIGMOD Conf. Management of Data.

[4] Lin, D.-I. and Kedem, Z.M. *"Pincer-search: A new algorithm for discovering the maximum frequent set".* In IEEE Transaction on Knowledge and Data Engineering, Vol 14, Issue 3, 2002

[5] Agrawal, R., Srikant, R: *"Fast Algorithms for Mining Association Rules*". Proc. Of the VLDB Conference (1994) 487–489, Santiago (Chile)

[6] Savasere, A., Omiecinski, E., and Navathe, S. 1995. *"An efficient algorithm for mining association rules in large databases*". In 21st VLDB Conf.

[7] D.G. Kourie, Sergei O ,B. W. Watson ,DVD Merwe "*An incremental algorithm to construct a lattice of set intersections*". Science of Computer Programming, Vol 74, Issue 3,2009, P 128-142.

[8] Toivonen, H., Klemettinen, M., Ronkainen, P., H¨at¨onen, K., and Mannila, H. 1995. *"Pruning and grouping discovered association rules*". In MLnet Wkshp. on Statistics, Machine Learning, and Discovery in Databases.

[9] Zaki, M.J., Hsiao, C.-J., "*Efficient algorithms for mining closed itemsets and their lattice structure*", IEEE Transactions on Knowledge and Data Engineering, Volume: 17 , Issue: 4 2005 , Page(s): 462 – 478

[10] D.W. Cheung, J. Han, V.Ng and C.Y. Wong, "*Maintenance of Discovered Association Rules in Large Databases, An Incremental updating Techniques"* In Proc, Intl. Conf. on Data Engineering (ICDE'96), Pages 106 – 114.

[11] Yonatan Aumann, Ronen Feldman, Orly Lipshtat, "*Borders: An Efficient Algorithm for Association Generation in Dynamic Databases"* Journal of Intelligent Information System, 12, 61 – 73 (1999).

[12] Lei Wen, **"**An efficient algorithm for mining frequent closed itemset**",** *Fifth World Congress on Intelligent Control and Automation, (WCICA 2004). Page(s): 4296 - 4299 Vol.5*

[13] Pasiquir, Bastide, Y. Stemme G, & Lakhal, *"Generating a Condensed Representation for Association Rule"* Journal of Intellegent system 24(1), 29-60, 2005

AUTHOR'S PROFILE

R. Vijaya Prakash is presently working as associate Professor in Department of Computer Science & Engineering,SR Engineering College, Warangal, Pursuing Ph.D from Kakatiya University, Warangal. He is completed his MCA from Kakatiya University and M.Tech(IT) from Punjabi University, Patiala.

Pr**of. A. Govardhan** is working as professor in Department of Computer Science & Engineering, JNT niversity, Hyderabad. He Received his B.Tech in 1992 from Osmania University, M.Tech in 1994 from Jawahar Lal Nehru University, Delhi and Ph.D from JNT University in 2003. He is very eminent person in the field of teaching he guided 4 Ph.D scholars.

Prof. SSVN. Sarma is a Professor in Mathematics, He Worked in Department of Mathematics and Department of Informatics, Kakatiya University over period of 30 years. Presently he is working as Professor in Dept. of Computer Science and Engineering,Vaagdevi College of Engineering ,Warangal. His research interests include artificial Intelligence, Computer Networks, Data Mining, Software Engineering