INFORMATION QUALITY IMPROVEMENT THROUGH ASSOCIATION RULE MINING ALGORITHMS DFCI, DFAPRIORI-CLOSE, EARA, PBAARA, SBAARA.

[1]E.Ramaraj [2]R.Gokulakrishnan [3]K.Rameshkumar

[1]Director, Computer Centre, Alagappa University, Karaikudi. e_mail: dr_ramaraj@yahoo.co.in
[2]Lecturer, Department of IT, J.J.College of Arts and Science, Pudukkottai. e_mail: rajgokul2002@yahoo.com
[3]FullTime Ph.D Scholar, Department of CSE, Alagappa University, Karaikudi, e_mail: rameshkumar_phd@yahoo.co.in

ABSTRACT

This paper concentrates on the difficulty of the value of the set of revealed association rules. This problem is important since real-life databases capitulate most of the time several thousands of rules with high confidence and propose new algorithms based on closed sets to reduce the mining to bases for exact and estimated rules. Once frequent closed itemsets which constitute a generating set for both frequent itemsets and association rules have been discovered. Proposed algorithms for efficiently generating bases for association rules. A basis is a set of non-redundant rules from which all association rules can be derived, thus it captures all useful information. Moreover, its size is significantly reduced compared with the set of all possible rules because redundant and thus useless rules are discarded. New approach has a twofold advantage on one hand, the user is provided with a smaller set of resulting rules, easier to handle, and information of improved quality. On the other hand, execution times are reduced compared with the discovering of all association rules. Chess dataset used for experiments.

Keywords: *Apriori algorithm, closure operator, itemsets, Association rules, Chess Dataset.*

1 INTRODUCTION

The approach presented in this paper belongs to the second trend since it aims to extract not all possible rules but a sub-set called basis or cover for association rules. When computing such a basis, redundant rules are discarded since they do not relevant knowledge. Such a pruning operation is a key-step during rule extraction, and significantly reduces the resulting set. An association rule is to exhibit relationships between data items or attributes and compute the precision of each relationship in the database. Usual precision measures are support and confidence that point the proportion of database transactions or objects upholding each rule out. When an association rule has support and confidence exceeding some user-defined minimum thresholds, the rule is considered as relevant and the extracted knowledge would likely be used for supporting decision making. Various approaches have been proposed for an increased efficiency of rule discovery. Among approaches addressing the described issue, some main trends can be distinguished, filtering rules and Boolean operators for selecting rules given items. A similar approach expanded with a measure of usefulness of extracted rules, called improvement, is proposed an SQL-like operator called Mine Rule, allowing the specification of general extraction criteria, is proposed. The quoted approaches operate Apriori, i.e. once huge amount of rules are extracted, querying facilities make it possible to handle rule subsets selected according to the user preferences. In contrast, the second trend addresses the problem with Apriori vision, by attempting to minimize the number of exhibited rules. Information about taxonomies is used to define criterion of interest which apply for pruning redundant rules.

This paper states the foundations of new approach since it makes it possible to generate the bases from frequent closed itemsets by avoiding handling of large sets of rules. Previous achieves frequent closed itemsets from frequent itemsets without accessing the dataset, called Apriori-Close, extends the Apriori algorithm by discovering simultaneously frequent itemsets and

frequent closed itemsets without additional execution time. Then, using the frequent closed itemsets and the pseudo-closed itemsets defined in lattice theory, Here, define the exact association rules with a maximum confidence. Rules in this basis are non-redundant exact rules with minimal antecedent and maximal consequent. Besides, using the frequent closed itemsets, This paper define the proper basis and the structural basis for approximate association rules. The proper basis is a small set containing the most informative and useful approximate rules: the non-redundant informative rules. The structural basis can be viewed as an abstract of all approximate rules that hold and can be useful when the proper basis is large. We propose three algorithms intended for yielding these three bases. Using the set of frequent closed itemsets, generating the evoked bases is performed without any access to the dataset.

An algorithm discovering closed and pseudo-closed. However, this algorithm does not consider the support of itemsets and, since it works only in main memory, it cannot be applied when the number of objects exceeds some hundreds and the number of items some tens. The association rule framework is defined. Fitting in this groundwork, efficient algorithms that discover frequent closed itemsets for association rules are defined: the Close algorithm [24] for correlated data and the A-Close algorithm [23] for weakly correlated data. The work presented in this paper shows that frequent closed itemsets constitute a generating set for frequent itemsets and association rules, extends the Apriori algorithm and algorithms for discovering maximal frequent Itemsets to generate frequent closed itemsets, adapts the results for exact and partial Implications to the context of association rules. This adaptation is based on the generating set, presents new algorithms for generating bases for exact and approximate association Rules using frequent closed itemset and the algorithms proposed are efficient for both improving the usefulness of extracted association rules and decreasing the execution time of the association rule extraction. As shown by experiments, the proposed process for extracting bases does not require any overhead compared with the traditional approaches for discovering association rules.

This paper addresses the concept of basis for both exact and approximate association rules. New algorithms for discovering frequent and frequent closed itemsets are described and the following section presents algorithms computing the bases for association rules from the frequent closed itemsets. Finally, as a conclusion, suggest more research.

2 FREQUENT ITEM SETS AND ASSOCIATION RULE CONSTRUCTION

This paper present the association rule framework based on the closure operators and connection, primarily introduced. A data mining context1 is defined as D = (O, I,R), where O and I are finite sets of objects and items respectively. R subset of O × I is a binary relation between objects and items. Each couple (o, i) Є R denotes the fact that the object o Є O is related to the item i Є I. Depending on the target system, a data mining context can be a relation, a class, or the result of an SQL/OQL query. Datamining context D consisting of objects identified by their OID.

A. Let I be a subset of items from D. The support count of the itemset I in D is: supp(I)= ||g(I)|| / ||O|| I is said to be frequent if the support of I in D is at least minsupp.

The set L of *frequent itemsets* in D is:

$L = \{(I \subseteq I) \text{ supp } (I) \geq \text{minsupp}\}$

B. An *Association rule* is an implication between two itemsets, with the form $I_1 \rightarrow I_2$ where I_1; $I_1 \subseteq I$, I_1; $I_2 \neq \theta$ and $I_1 \cap I_2 = \theta$. I_1 and I_2 are called respectively the antecedent and the consequent of the rule. The support supp(r) and confidence conf(r) of an association rule r : $I_1 \rightarrow I_2$ are defined as follows:

$supp(r) = \|\|g(I_1 \cup I_2)\|\| \ / \ \|\|\theta\|\|$,
$con\,f(r) = \ supp(I_1 \cup \ I_2) / supp(I_2)$

Association rules holding in the context are those that have support and confidence greater than or equal to the minsupp and minconf thresholds respectively. Definition of the set AR of association rules holding in D given minsupp and minconf thresholds as follows:

$AR = \{r: I_1 \rightarrow I_2 \rightarrow I_1 \mid I_1 \subset I_2 \subseteq I \wedge supp(I_2) \geq \text{minsupp} \wedge conf(r) \geq \text{minconf}\}$

If conf(r)=1 then r is called an exact association rule or implication rule, otherwise r is called approximate association rule. Exact and approximate association rules extracted from D for minsupp = 2/5 and minconf = 1/2 .

Frequent closed itemsets constitute a generating set for frequent itemsets and association rules. Then, we characterize the association rules and the proper and structural bases for approximate association rules as

defined and extended in this paper to the context of association rules.

A. The operators h = f o g in 2^I and h^1 = g o f in 2^O are *closure operators*, here we use the notation f o g (I) = f(g(I)) and gof(o) = g(f(o)). Given the set (f, g), the following properties hold for all I, I_1, $I_2 \subseteq$ I and O, O_1, $O_2 \subseteq$ O.

Extension

$I \subseteq h(I)$ $O \subseteq h^1(O)$

Idempotency

$h(h(I))=h(I)$ $h^1(h^1(O)) = h^1(O)$

C. An *itemset* I ⊆ I in D is a closed itemset if h(I) = I. A *closed itemset* I is said to be frequent if the support of I in D is at least *min-sup*. The smallest *closed itemset* containing an itemset I is h(I), the closure of I. The set FC of *frequent closed itemsets* in D is defined as follows:

FC = {I ⊆ I' | I = h(I) ^ supp(I) ≥ minsupp}

A *frequent closed itemset* is a maximal set of items common to a set of objects, for which support is at least *minsupp*. The frequent closed itemsets in the context {5} for *minsupp=2/5* are presented. The *itemset* {2,3,5} is a *frequent closed itemset* since it is the maximal set of items common to the objects {2, 3, 5}. The *itemset* {2,3} is not a *frequent closed itemset* since it is not a maximal set of items common to some objects: all objects in relation with the *items* {2} and {3} (objects) are also in relation with the *item* {5}.

Hereafter, we demonstrate that the set of frequent closed itemsets with their support is the smallest collection from which frequent itemsets with their support and association rules can be

Monotonicity

$I_1 \subseteq I_2 \rightarrow h(I_1) \subseteq h(I_2)$

$O_1 \subseteq O_2 \rightarrow h^1(O_1) \subseteq h^1(O_2)$

generated (it is a generating set). The support of an *itemset I* is equal to the support of the smallest closed itemset containing *I: supp(I) = supp(h(I)).*

The set of maximal frequent itemsets M = {I Є L | ∃ I' Є L where I⊂ I' } is identical to the set of maximal frequent closed itemsets

MC = {I Є FC | ∃ I' Є FC where I ⊂ I'}.

Frequent closed itemset	Support
{θ}	5/5
{3}	4/5
{1,3}	3/5
{2,5}	4/5
{2,3,5}	3/5
{1,2,3,5}	2/5

Table 1: Frequent Closed Itemsets Extracted from D for minsupp

C. The set FC of frequent closed itemsets with their support is a *generating set* for all frequent itemsets and their support, and for all association rules holding in the dataset, their support and their confidence. All frequent itemsets can be derived from the maximal frequent closed itemsets. The support of each frequent itemset can be derived from the support of frequent closed itemsets. Then, the set of frequent closed itemsets FC is a generating set for both the set of frequent itemsets L and the set of association rules.

2.1 Exact Association Rules

Let FP be the set of frequent pseudo-closed itemsets in P.

The set P = {r : $I_1 \rightarrow h(I_1) - I_1 \in I_1 \in FP \wedge I_1 \neq \theta$ is a basis for all exact association rules holding in the dataset. Minimal with respect to the number of rules since there can be no complete set with fewer rules than there are frequent pseudo-closed itemsets. A frequent pseudo-closed *itemset I* is a frequent non-closed itemset that includes the closures of all frequent pseudo-closed itemsets included in I. The set FP of frequent pseudo closed itemsets and the association rules extracted for minsupp=2/5 and minconf =1/2. The itemset {1,2} is not a frequent pseudo closed itemset since the closures of {1} and {2} respectively {1,3} and {2,5} are not included in {1,2}.

{1,2,3,5} is not a frequent pseudo-closed itemset since it is closed.

Frequent pseudo-closed temset	Support
{1}	3/5
{2}	4/5
{5}	4/5

Table 2: Frequent Pseudo-Closed Itemsets

Exact rule	Support
{1}→ {3}	3/5
{2}→ {5}	4/5
{5}→ {2}	4/5

Table 3: extracted from minsupp

Furthermore, FC is the smallest generating set for L and AR. Hence, even if frequent itemsets can be derived from the maximal frequent itemsets, passes over the dataset are still needed to compute the frequent itemset supports.

2.2 Approximate Association Rules

Let FC be the set of frequent closed itemsets in {4}. The set

PB = {r : $I_1 \rightarrow I_2 \rightarrow I_1 \mid I_1, I_2 \in FC \wedge I_1 \neq \theta \wedge I_1 \subset I_2 \wedge conf(r) \geq$ mincon *f*} is a basis for all approximate association rules holding in the dataset. Association rules in PBA are proper approximate association rules. The proper basis for approximate association rules extracted from {4} for minsupp=2/5 and minconf =1/2 are presented.

Approximate rule	Support	Confidence
{2,3,5}→{1}	2/5	2/3
{1,3}→{2,5}	2/5	2/3
{ 2,5}→{1,3}	2/5	2/4
{ 2,5}→{3}	3/5	3/4
{3}→{1,2,5}	2/5	2/4
{3}→{2,5}	3/5	3/4
{3}→{1}	3/5	3/4

Table 4: Proper Basis Extracted from {4} for minsupp = 2/5 and minconf = 1/2.

2.3 Structural Basis for Approximate Association Rules

Let FC be the set of frequent closed itemsets in D. here, define G_{FC} = (V,E) as the undirected graph associated with FC where the set of vertices V and the set of edges E. Maximal Confidence Spanning Forest F_{FC}: Let F_{FC} = (V,E) be the maximal confidence spanning forest associated with FC. F_{FC} is obtained from the undirected graph G_{FC} = (V,E) by suppressing transitive edges and cycles. Cycles are removed by deleting some edges that enter the last vertex I (maximal vertex with respect to the inclusion) of the cycle. Among all edges entering in I, those with confidence less than the maximal confidence value associated with an edge with the form (I^1, I) Є E are deleted. If more than one edge have the maximal confidence value, the first one in lexicographic order is kept.

Let SB be the set of association rules represented by edges in FFC except rules from the vertex {∅}. The set SB = {r : $I_1 \rightarrow I_2 - I_1 \mid I_1, I_2 \in V \wedge I_1 \subset I_2 \wedge I_1 \neq \emptyset \wedge (I_1, I_2) \in E'$} is a basis for all approximate association rules holding in the dataset (I is the consequent of at most one approximate association rule in SB).The structural basis for approximate association rules

extracted from {4} for minsupp=2/5 and minconf =1/2 is presented.

Approximate rule	Support	Confidence
{1,3}→{2,5}	2/5	2/3
{2,5}→{3}	3/5	3/4
{3}→{1}	3/5	3/4

Table 5: Structural Basis Extracted from {4} for minsupp = 2/5 and minconf = 1/2.

3 FREQUENT AND FREQUENT CLOSED ITEMSETS

We propose a new algorithm to achieve frequent closed itemsets from frequent itemsets without accessing the dataset. This algorithm discovers frequent closed itemsets while for instance an algorithm for discovering maximal frequent itemsets is used. also, we present an extension of the Apriori algorithm called Apriori-Close for discovering frequent and frequent closed itemsets without additional computation time. Like in the Apriori algorithm, we assume in the following that items are sorted in lexicographic order and that k is the size of the largest frequent itemsets. Based on theorem k is also the size of the largest frequent closed itemsets.

Many efficient algorithms for mining frequent itemsets and their support have been proposed. Efficient algorithms for discovering the maximal frequent itemsets and then achieve all frequent itemsets .All these algorithms give as result the set $L = {}_{i=1}U^{i=k}$ where Li contains all frequent i-itemsets (itemsets of size i). Based on Proposition 1 and theorem, the frequent closed itemsets and their support can be computed from the frequent itemsets and their support without any dataset access.

The pseudo-code to determine frequent closed itemsets among frequent itemsets. The input of the algorithm are sets Li, $1 \leq I \leq k$, containing all frequent itemsets in the dataset. It recursively generates the sets FCi, $0 \leq I \leq k$, of frequent closed i-itemsets from FC_k to FC_0.

Algorithm1:DFCIA (Derived Frequent Closed Itemsets from Frequent Itemsets Algorithm)

1. $FC_k \leftarrow L_k$;
2. *for* (i←k-1; i ≠ 0; i - -) *do begin*
3. $FC_i \leftarrow \{\}$; //FCi Set of frequent closed i-itemsets and their support
4. *forall* itemsets l ∈ L_i do begin
5. isclosed ← true; // Isclosed -Variable indicating if the considered itemset is closed or not
6. *forall* itemsets l' ∈ L_{i+1} *do begin //* Li Set of frequent i-itemsets and their support
7. *if* (l ⊂ l') and (l.support=l'.support)
8. *then* isclosed← false;
9. *end*
10. *if* (isclosed = true) then FCi← FCi U {l};
11. *end*
12. *end*
13. $FC_0 \leftarrow \{\theta\}$;
14. *forall* itemsets l ЄL_1 *do begin*
15. *if* (l.support = ||O||) then $FC_0 \leftarrow \{\}$;
16. *end*

First, the set FC_k is initialized with the set of largest frequent itemsets L_k. Then, the algorithm iteratively determines which i-itemsets in Li are closed from L_{k-1} to L_1., for each frequent itemset l in Li, we verify that l has the same support as a frequent (i+1)-itemset l' in L_{i+1} in which it is included. If so, we have l' subset of h(l) and then l ≠h(l): l is not closed. Otherwise, l is a frequent closed itemset and is inserted in FCi (step 9). During the last phase, the algorithm determines if the empty itemset is closed by first initializing FC_0 with the empty itemset and then considering all frequent 1-itemsets in L1. If a 1-itemset l has a support equal to the number of objects in the

context, meaning that l is common to all objects, then the itemset cannot be closed (supp({θ}) = ||O|| = supp(l)) and is removed from FC_0. Thus, at the end of the algorithm, each set FCi contains all frequent closed i-itemsets. Since all maximal frequent itemsets are maximal frequent closed itemsets, the computation of the set FCk containing the largest frequent closed itemsets is correct. The correctness of the computation of sets FCi for I < k relies on Proposition 1. This proposition enables to determine if a frequent i-itemset l is closed by comparing its support and the supports of the frequent (i+1)-itemsets in which l is included. If one of them has the same support as l, then l cannot be closed.

3.1 Apriori-Close Algorithm

In this section, we present an extension of the Apriori algorithm computing simultaneously frequent and frequent closed itemsets. The pseudo-code is given in Algorithm 2. The algorithm iteratively generates the sets Li of frequent i-itemsets from L_1 to L_k. Besides, during the ith iteration, all frequent closed (i-1)-itemsets in FC_{i-1} are determined. The set FC_k is determined during the last step of the algorithm.

First, the variable *k* is initialized to *0*. Then, the set L_1 of frequent 1-itemsets is initialized with the list of items in the context and one pass is performed to compute their support. The set FC_0 is initialized with the empty itemset and the supports of Li Set of frequent i-itemsets, their support and marker isclosed indicating if closed or not .FCi Set of frequent closed i-itemsets and their support.

Algorithm2: DFApriori-Close (Discovering Frequent and Frequent Closed Itemsets with Apriori-Close)

1. k← 0;
2. itemsets in L_1 ←{1-itemsets};
3. L_1 ←Support-Count(L_1); // Li- Set of frequent i-itemsets and their support
4. FC0← {θ};
5. *forall* itemsets l ∈ L_1 *do begin*
6. *if* (l.support < minsupp) *then* L_1←L_1 – {l};
7. *else if* (l.support = ||O||) *then* FC_0←{};
8. *end*
9. *for* (i ←1; Li ≠{}; i^{++}) *do begin*
10. *forall* itemsets l ∈ Li *do* l'.isclosed←true; // Isclosed- indicating if closed or not
11. L_{i+1} ←Apriori-Gen(Li);
12. *forall* itemsets l ∈ Li+1 do begin
13. *forall* i-subsets l' of l do begin
14. *if* (l' ∉ Li) *then* Li+1 Li+1 n {l};
15. *end*
16. *end*
17. Li+1 Support-Count(Li+1);
18. *forall* itemsets l ∈ L_{i+1} *do begin*
19. *if* (l.support < minsupp) *then* Li+1 Li+1 n {l};
20. *else do begin*
21. *forall* i-subsets l'∈ Li of l *do begin*
22. *if* (l.support = l'.support) *then* l'.isclosed ←false;
23. *end*
24. *end*
25. *end*
26. FCi ←{l ∈ Li | l.isclosed = true}; // FCi- Set of frequent closed i-itemsets and their support
27. k ←i;
28. *end*
29. FC_k← L_k;

itemsets in L1 are considered (steps 5 to 8). All infrequent 1-itemsets are removed from L_1 and if a frequent 1-itemset has a support equal to the number of objects in the context then the empty itemset is removed from FC_0 (step 7). During each of the following iterations, frequent

itemsets of size i+1, k > i ≥1, and frequent closed itemsets of size i are computed as follows. For all frequent i-itemsets in Li, the marker isclosed is initialized to true. A set Li+1 of possible frequent (i+1)-itemsets is created by applying the Apriori-Gen function to the set Li .For each of these possible frequent (i+1)-itemsets, we check that all its subsets of size i exist in Li. One pass is performed to compute the supports of the remaining itemsets in Li+1. Then, for each (i+1)-itemsets l ∈ Li+1, if l is infrequent then it is discarded from L_1+1. Otherwise for all i-subsets l0 of l, we verify that supports of l0 and l are equal; if so, then l0 cannot be a closed itemset and its marker isclosed is set to false (steps 20 to 24). Then, all frequent i-itemsets in Li for which marker isclosed is true are inserted in the set FCi of frequent closed i-itemsets and the variable k is set to the value of i . Finally, the set FCk is initialized with the frequent k-itemsets in Lk.

A. Apriori-Gen function: The Apriori-Gen function [2] applies to a set Li of frequent i-itemsets. It returns a set Li+1 of potential frequent (i+1)-itemsets. A new itemset in Li+1 is created by joining two itemsets in Li sharing common first i-1 items.

B. Support-Count function: The Support-Count function takes a set Li of i-itemsets as argument. It efficiently computes the supports of all itemsets l ∈ Li. Only one dataset pass is required: for each object o read, the supports of all itemsets l ∈ Li that are included in the set of items associated with o, i.e. l ⊆ f({o}), are incremented. The subsets of f({o}) are quickly found using the Subset function.

C. Correctness

Since the support of a frequent closed itemset l is different from the support of all its

Supersets the computation of sets FCi for i < k is correct. Hence, a frequent i-itemset l'∈ Li is determined closed or not by comparing its support with the supports of all frequent (i + 1)-itemsets l ∈ Li+1 for which l' ⊂ l. The correctness of the computation of the set FC_k containing the largest frequent closed itemsets.

4. Generating - Association Rules

The pseudo-code generating for exact association rules is given in Algorithm. The algorithm takes as input the sets Li, 1 ≤ I ≤ k, containing the frequent itemsets and their support, and the sets FCi, 0 ≤ i ≤ k, containing the frequent closed itemsets and their support. It first computes the frequent pseudo-closed itemsets iteratively and then uses them to generate the association rules.

First, the set EA is initialized to the empty set. If the empty itemset is not a closed itemset ,it is then necessarily a pseudo-closed itemset, it is inserted in FP_0. Otherwise FP_0 is empty. Then, the algorithm recursively determines which i-itemsets in Li are pseudo closed from L_1 to Lk. At each iteration, the set FPi is initialized with the list of frequent i-itemsets that are not closed (step 5) and each frequent i-itemsets l in FPi is considered as follows. The variable pseudo is set to true. We verify for each frequent pseudo-closed itemset p previously discovered (i.e. in FPj with j < i) if p is contained in l. In that case and if the closure of p is not included in l, then l is not pseudo-closed and is removed from FPi. Otherwise, the closure of l ,the smallest frequent closed itemset containing l is determined. Once all frequent pseudo-closed itemsets p and their closure are computed, all rules with the form r : p ⇒ p.closure - p) are generated. The algorithm results in the set EA containing all rules in the basis for exact association rules.

A.Correctness:

Since the itemset ∅ has no subset, if it is not a closed itemset then it is by definition a pseudo-closed itemset and the computation of the set FP_0 is correct. The correctness of the computation of frequent pseudo-closed i-itemsets in FPi for 1 ≤ i ≤ k relies on Definition 7. All frequent i-itemsets l in Li that are not closed, i.e. not in FCi, are considered. Those l containing the closures of all frequent pseudo-closed itemsets that are subsets of l are inserted in FPi. According to Definition 7, these i-itemsets are all frequent pseudo-closed i-itemsets and the sets FPi are correct.

The association rules generated in the last phase of the algorithm are all rules with a frequent pseudo-closed itemset in the antecedent. Then, the resulting set EA corresponds to exact association.

4.1 Generating Proper Basis for Approximate Association Rules

The pseudo-code generating the proper basis for approximate association rules is presented and Notations are given. The algorithm takes as

input the sets FCi,1 ≤ I ≤ k, containing the frequent closed non-empty itemsets and their support. The output of the algorithm is the proper basis for approximate association rules PBA.

Algorithm 3 : EARA (Exact Association Rule based Algorithm)

1) EAR ← {}; // EAR → Exact Association rules.
2) if (FC_0 = {}) then FP_0← {∅};
3) else FP_0 ←{};
4) for (i← 1; i ≤ k; i++) do begin
5) FPi ← Li \ FCi; // Li-Set of frequent i-itemsets and their support.
6) forall itemsets l ∈ FPi do begin
7) pseudo true;
8) forall itemsets p ∈ FPj with j < i do begin
9) if (p ⊂ l) and (p.closure ⊄ *I*)
10) then do begin
11) pseudo false;
12) FPi← FPi \ {l}; // FPi- Set of frequent pseudo-closed i-itemsets, their closure and their support
13) endif
14) end
15) if (pseudo = true) then l.closure Min ⊆ (fc Є $FC_{j>i}$ j | l ⊆ c});
16) end
17) end
18) forall sets FPi where FPi ≠{} do begin
19) forall pseudo-closed itemsets p Є FPi do begin
20) EAR ← EAR U {r : p ⇒(p.closure − p),p.support};
21) end
22) end

The set PB (Proper basis for approximate association rules) is first initialized to the empty set (step 1). Then, the algorithm iteratively considers all frequent closed itemsets l ∈ FCi for i ≤ k. It determines which frequent closed itemsets l' ∈ $FC_{j<i}$ are subsets of l and generates association rules with the form l' → l → l' that have sufficient confidence as follows. During the ith iteration, each itemset l in FCi is considered (. For each set FCj , 1 ≤ j < i, a set S_j containing all frequent closed j-itemsets in FCj that are subsets of l is created. Then, for each of these subsets l' ∈ Sj, compute the confidence of the proper approximate association rule r : l'→ l − l'. If the confidence of *r* is sufficient then *r* is inserted in *PBA*. At the end of the algorithm, the set *PBA* contains all rules of the proper basis for approximate association rules.

Subset function takes a set X of itemsets and an itemset y as arguments. It determines all itemsets x ∈ X that is subsets of y. In algorithm implementation, frequent and frequent closed itemsets are stored in a prefix-tree structure in order to improve efficiency of the subset search.

Correctness The correctness of the algorithm relies on the fact that we examine all proper approximate association rules holding in the dataset. For each frequent closed itemset, the algorithm computes, among its subsets, all other frequent closed itemsets. Then, the generation of all rules between two frequent closed itemsets having succient confidence is ensured. These rules are all proper approximate association rules holding in the dataset, and the resulting set PB is the proper basis for approximate association rules.

4.2 Generating Structural Basis for Approximate Association Rules

The pseudo-code generating the structural basis for approximate association rules is given in Algorithm. The algorithm takes as input the sets FCi(Set of frequent closed i-itemsets and their support), 1 ≤ i ≤ k, of frequent closed non-empty itemsets and their support. It generates the structural basis for approximate association rules SB represented by the maximal confidence spanning forest F_{FC} associated with
FC =$_{i=1}U^{i=k}$ FCi (without the empty itemset).

The set SB (Structural basis for approximate association rules) is first initialized to the empty

set. Then, the algorithm iteratively considers all frequent closed itemsets l ∈ FCi fori ≤ k. It determines which frequent closed itemsets l' ∈ $FC_{j<i}$ are covered by l, i.e. are direct predecessors of l, and then generates the maximal confidence association rules with the form l → l' → l. During the i^{th} iteration, each itemset l in FCi is considered as follows. The set *CR(*Set of candidate approximate association rules*.)* of candidate association rules with l in the consequent is initialized to the empty set. For 1 ≤ j < i, sets Sj containing all frequent closed j-itemsets in FCj that are subsets of l are created.

Algorithm 4: PBAARA (Proper Basis Approximate Association Rules Algorithm)

1) *PBA* ← {} //PBA-proper Basis for approximate Association rule
2) *for* (i ← 2; i ≤ k; i++) do begin
3) *forall* itemsets l ∈ FCi do begin
4) *for* (j← I - 1; j > 0; j- -) do begin
5) Sj Subsets(FCj , l); // Sj Set of j-itemsets that are subsets of the considered itemset
6) *forall* itemsets l'∈ Sj do begin
7) conf(r) ← l.support / l'.support;
8) *if* (conf(r) ≥ mincon*f*)
9) *then* PBA ← PBA U {r : l' → l – l', l.support, con*f*(r)};
10) *end*
11) *end*
12) *end*
13) *end*

All these subsets are considered in decreasing order of their sizes. For each of these subsets, the confidence of the proper approximate association rule *r : l' ∈ l – l'* is computed. If the confidence of r is sufficient, r is inserted and all subsets are removed from $S_{n<j}$. This because rules with the form *l''→ l – l''* with l'' ∈ $S_{n<j}$ are transitive proper approximate rules. Finally, the candidate proper approximate rules with l in the consequent are pruned. The maximum confidence value maxcon*f* of rules determined and the first rule with such a confidence is inserted. At the end of the algorithm, the set contains all rules in the structural basis for approximate association rules.

- **Correctness:** The algorithm considers all association rules *l'→l – l'* with *confidence* ≥ *minconf* between two frequent closed itemsets l and l' where l covers l'. These rules are all proper non transitive approximate association rules that hold and can be represented by the edges of the graph GFC without transitive edges. Moreover, among all rules with the form $X \rightarrow l - X$ (generated from l), Here, keep only the first one with confidence equal to the maximal confidence of rules

$$X \rightarrow l - X.$$

Algorithm 5: SBAARA (Structural Basis for Approximate Association Rules Algorithm)

1) *SBAA* ←{};
2) *for* (i ←2; i ≤ k; i++) *do begin*
3) *forall* itemsets l ∈ FCi *do begin*
4) CR ←{};
5) *for* (j← i– 1; j > 0; j- -) *do begin*
6) Sj Subsets(FCj ,l);
7) *end*
8) *for* (j ← i– 1; j > 0; j- -) *do begin*
9) *forall* itemsets l' ∈ Sj *do begin*
10) con*f*(r) ← l.support / l'.support;
11) *if* (conf(r) ≥ mincon*f*)
12) *then* CR← CR U {r : l' → l – l', l.support, conf(r)};

13) *for* (n← j–1; n > 0; n- -) *do begin*
14) S_n← S_n – Subsets(S_n, l');
15) *end*
16) *endif*
17) *end*
18) *end*
19) *if* (CR ≠{}) then
20) maxconf ← $Max_{r \in CR}$(conf(r));
21) *find first* {r ∈CR | con*f*(r) = maxcon*f*};
22) SBAA ← SBAA U {r};
23) *endif*
24) *end*
25) *end*

5 Experimental Results- Relative Performance of Apriori and Apriori-Close

Experiments were performed on a Pentium IV PC with a 1 GHz clock rate, 700 MBytes of RAM, running the Linux operating system. Algorithms were implemented in JAVA. Characteristics of the datasets used are given below. These datasets are the market basket data and the Chess Dataset. In all experiments, we attempted to choose significant minimum support and confidence threshold values: we observed threshold values used in other papers for experiments on similar data types and examined rules extracted in the bases.

Name	Number of objects	Average size of objects	Number of items
Chess	100,000	10	1,000
Mushrooms	8,416	23	127
Market basket	10,000	20	386

Table 6: Datasets.

We conducted experiments to compare response times obtained with Apriori and Apriori-Close on the four datasets. Results for the chess and market basket datasets are. We can observe that execution times are identical for the two algorithms: adding the frequent closed itemset derivation to the frequent itemset discovery does not induce additional computation time.

Minsupp	Apriori	Apriori-Close
2.0%	1.99s	1.97s
1.0%	3.47s	3.46s
0.5%	9.62s	9.70s
0.25%	15.02s	14.92s

Table 7: Chess dataset

Minsupp	Apriori	Apriori-Close
90%	0.28s	0.28s
70%	0.73s	0.73s
50%	2.40s	2.70s
30%	18.22s	17.93s

Table 8: Market basket data

Execution Times of Apriori and Apriori-Close.

Total number of approximate association rules, their number in the proper basis and in the structural basis for approximate rules, and the number of non-transitive rules in the proper basis for approximate rules. For example in the context D, rules {3}→{1} and {1,3}→{2,5} are extracted, as well as the rule {3}→{1,2,5} which is clearly transitive. Since by construction, it's confidence-retrieved by multiplying the confidence of the two former -is less than theirs, this rule is the less interesting.

Reducing the extraction to non-transitive rules in the proper basis for approximate rules can also be interesting. Such rules are generated by a variant of Algorithm with the pruning strategy and candidate rules in CR are inserted in SB. Datasets the average relative size of bases compared with the sets of all rules obtained. In the case of weakly correlated data, no exact rule is generated and the proper basis for approximate rules contains all approximate rules that hold. The reason is that, in such data, all frequent

itemsets are frequent closed itemsets. In the case of correlated data, the number of extracted rules in bases is much smaller than the total number of rules that hold. Dataset the execution times of the computation of all rules. Execution times of the derivation of the exact rules and the proper basis for non-transitive approximate rules are not presented since they are identical.

6 Conclusions

In this paper, Five algorithms *DFCI, DFApriori-Close, EARA, PBAARA, SBAARA* for efficiently generating bases for association rules. A basis is a set of non-redundant rules from which all association rules can be derived, thus it captures all useful information. Moreover, its size is significantly reduced compared with the set of all possible rules because redundant, and thus useless, rules are discarded. Our approach has a twofold advantage: on one hand, the user is provided with a smaller set of resulting rules, easier to handle, and information of improved quality. On the other hand, execution times are reduced compared with the discovering of all association rules. Such results are proved (in the groundwork of lattice theory) and illustrated by experiments, achieved from real-life datasets. Integrating reduction methods Templates can directly be used for extracting from the bases all association rules matching some user specified patterns. Information in taxonomies associated with the dataset can also be integrated in the process as proposed for extracting bases for generalized multi-level association rules. Integrating item constraints and statistical measures in the generation of bases requires further work. Functional and approximate dependencies Algorithms presented in this paper can be adapted to generate bases for functional and approximate dependencies. The functional dependencies constituted of minimal non-trivial functional dependencies. Hence, the number of rules is minimal; moreover these rules have minimal antecedent and maximal consequent. Furthermore, the proper and structural bases for approximate rules are also smaller than the basis for approximate dependencies defined. Adapting our algorithms to the discovery of functional and approximate dependencies is an ongoing research.

References

[1] R.Agrawal, T.Imielinski, and A.Swami. Mining association rules between sets of items in large databases. Proc. of the ACM SIGMOD Conference, pages 207-216, May 1993.

[2] R.Agrawal and R.Srikant. Fast algorithms for mining association rules. Proc. of the 20th VLDB Conference, pages 478-499, June 1994.

[3] E.Baralis and G.Psaila. Designing templates for mining association rules. Journal of Intelligent Information Systems, 7-32, July 1997.

[4] R.J.Bayardo. Efficiently mining long patterns from databases. Proc.of the ACM SIGMOD Conference, pages 85-93, June 1998.

[5] R.J.Bayardo, R.Agrawal, and D.Gunopulos. Constraint-based rule mining in large, Dense databases. Proc. of the 15th ICDE Conference, pages 188-197, March 1999.

[6] G.Birkhoff . Lattices theory. In Colloquium Publications XXV. American Mathematical Society, 1967. Third edition.

[7] S.Brin, R.Motwani, and C.Silverstein. Beyond market baskets: Generalizing association rules to correlation. Proc. of the ACM SIGMOD Conference, pages 265-276, May 1997.

[8] S.Brin, R.Motwani, J.D.Ullman, and S.Tsur. Dynamic itemset counting and implication rules for market basket data. Proc. of the ACM SIGMOD Conference, pages 255-264, May 1997.

[9] P.Burmeister. Formal concept analysis with ConImp: Introduction to the basic features.Technical report, Technische Hochschule Darmstadt, Germany, 1998.

[10] J.Demetrovics, L. Libkin, and I. B. Muchnik. Functional dependencies in relational databases: A lattice point of view. Discrete Applied Mathematics, 40:155-185, 1992.

[11] V.Duquenne and J.-L.Guigues. Famille minimale d'implication informatives resultant d'un tableau de donnees binaires. Mathematiques et Sciences Humaines,24(95):5-18, 1986.

[12] B.Ganter and K.Reuter. Finding all closed sets: A general approach. In Order, pages 283-290.Kluwer Academic Publishers, 1991.

[13] B.Ganter and R.Wille. Formal Concept Analysis: Mathematical Foundations. Springer, 1998.

[14] J.Han and Y.Fu. Discovery of multiple-level association rules from large databases. Proc. Of the 21st VLDB Conference, pages 420-431, September 1995.

[15] Y.Huhtala, J.Karkkanen, P.Porkka, and H.Toivonen. Efficient discovery of functional and approximate dependencies using partitions. Proc. of the 14th ICDE Conference, pages 392-401, Febuary 1998.

[16] M.Klemettinen, H.Mannila, P.Ronkainen, H.Toivonen, and A.I.Verkamo. Finding interesting rules from large sets of discovered association rules. Proc. of the 3rd CIKM Conference, pages 401-407, November 1994.

[17] D.Lin and Z.M.Kedem. Pincer-search: A new algorithm for discovering the maximum frequent set. Proc. of the 6th EDBT Conference, pages 105-119, March 1998.

[18] B.Liu,W.Hsu and S.Chen. Using general impressions to analyse discovered Classification rules. Proc. of the 3rd KDD Conference, pages 31-36, August 1997.

[19] M.Luxenburger. Implications partielles dans un contexte. Mathematiques, Informatique Et Sciences Humaines, 29(113):35-55, 1991.

[20] H.Mannila and K.J.Raha. Algorithms for inferring functional dependencies from relations. Data & Knowledge Engineering, 12(1):83-99, Febuary 1994.

[21] R. Meo, G. Psaila, and S. Ceri. A new SQL-like operator for mining association rules. Proc.of the 22nd VLDB Conference, pages 122-133, September 1996.

[22] R.T.Ng, V.S.Lakshmanan, J.Han, and A.Pang. Exploratory mining and pruning optimizations of constrained association rules. Proc. of the ACM SIGMOD Conference, pages 13-24, June 1998.

[23] N.Pasquier, Y.Bastide, R.Taouil, and L.Lakhal. Discovering frequent closed itemsets For association rules. Proc. of the 7th ICDT Conference, pages 398-416, January 1999.

[24] N.Pasquier, Y.Bastide, R.Taouil, and L.Lakhal. Efficient mining of association rules Using closed itemset lattices. Journal of Information Systems, 24(1):25-46, 1999.

[25] G.Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. Knowledge Discovery in Databases, pages 229-248, 1991.

[26] A.Savasere, E.Omiecinski, and S.Navathe. An efficient algorithm for mining association rules in large databases. Proc. of the 21st VLDB Conference, pages 432-444, Sept 1995.

[27] A.Silberschatz and A.Tuzhilin. What makes patterns interesting in knowledge discovery systems. IEEE Transactions on -Knowledge and Data Engineering, 8(6):970-974, December 1996.

[28] R.Srikant and R.Agrawal. Mining generalized association rules. Proc. of the 21st VLDB Conference, pages 407-419, September 1995.

[29] R.Srikant, Q.Vu, and R.Agrawal. Mining association rules with item constraints. Proc. Of the 3rd KDD Conference, pages 67-73, August 1997.

[30] H.Toivonen. Sampling large databases for association rules. Proc.of the 22nd VLDB Conference, pages 134-145, September 1996.

[31] R.Wille. Concept lattices and conceptual knowledge systems. Computers and Mathematics with Applications, 23:493-515, 1992.

[32] M.J.Zaki and M.Ogihara. Theoretical foundations of association rules. 3rd SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, June 1998.

[33] M.J.Zaki, S.Parthasarathy, M.Ogihara, and W.Li. New algorithms for fast discovery of association rules. Proc. of the 3rd KDD Conference, pages 283-286, August 1997.

Authors:

Dr.E.Ramaraj is presently working as a Director, computer centre at Alagappa University, Karaikudi. He has 20 years of teaching experience and 5 years of research experience. He has presented research papers in more than 20 national and international conferences and published more than 30 papers in national and international journals. His research areas include Data mining and Network security.

R.Gokulakrishnan is a Ph.D candidate in the Department of Computer Science and Engineering, Alagappa University, Karaikudi, Tamilnadu. He received his Master degree in Computer Applications in the year 1996. He has more than 4 years of industrial experience. Currently he is working as a Lecturer in Information Technology at J.J.College of Arts and Science, Pudukkottai, Tamilnadu. He has participated and presented the research papers in various National and International level conferences. His research interests include Data mining and Information system. University Grants Commission, New Delhi, recommended his candidature for International Commonwealth Scholarship in the year 2007.

K.Rameshkumar is presently doing Ph.D Fulltime at Dept of CSE, Alagappa university, Karaikudi. He has more than 3 years of industrial experience. He has participated and presented the research papers in various national and International level conferences. His research interests are Data mining, E-security and Quantum computing.