Int. J. Advance. Soft Comput. Appl., Vol. 1, No. 2, November 2009
ISSN 2074-8523; Copyright © ICSRS Publication, 2009
www.i-csrs.org

nVApriori : A novel approach to avoid irrelevant rules in association rule mining using n-cross validation technique

Eswara thevar Ramaraj and Krishnamoorthy Ramesh kumar

Computer Centre, Alagappa University, Karaikudi, Tamilnadu, India
e-mail: dr_ramaraj@yahoo.co.in
Dept. CSE, Alagappa University, Karaikudi, Tamilnadu, India
e-mail:rameshkumar_phd@yahoo.co.in

Abstract

Association rule mining finds interesting associations or correlations in a large pool of transactions. Apriori based algorithms are two step algorithms for mining association rules from large datasets. They find the frequent item sets from transactions as the first step and then construct the association rules. Though these algorithms generate multiple rules, most of the rules become irrelevant to the transactions. The exercise becomes costly in terms of memory usage and decision making is also not precise. This research addresses this drawback by developing ways to reduce irrelevant rules. This paper proposes the n-cross validation technique to filter such irrelevant rules. The proposed algorithm is called nVApriori (n-cross Validation based Apriori) algorithm. The proposed nVApriori algorithm uses a partition based approach to support the association rule validations. The proposed nVApriori algorithm has been tested with two synthetic datasets and two real datasets. The performance analysis is compared with Apriori, most frequent rule mining algorithm and non redundant rule mining algorithm to study the efficiency. This proposed work aims at reducing a large number of irrelevant rules and produces a new set of rules having high levels of confidence.

Keywords: *Data Mining, Association rule, nVApriori, frequent itemset mining*

1 Introduction

Association rule mining aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in transaction databases or other data repositories [1], [8]. Association rules are widely used in various areas like telecommunication networks, market and risk management and inventory control. Problems are usually split into two, the first finds those itemsets whose occurrences exceed a predefined support threshold in the database called frequent or large itemsets. In the next step it generates association rules from the itemsets by applying the constraints of minimal confidence. For eg. If L_k is the itemset where $L_k = \{i_1, i_2, \ldots, i_k\}$, association rules with these temsets are generated in the following way: the confidence constraint is applied first rule $\{i_1, i_2, \ldots, i_{k-1}\} \Rightarrow \{i_k\}$, to check the rules relevance before accepting it as a rule. Most researches focus on finding candidate itemsets and frequent itemsets. itemsets which exceed the support threshold in frequent itemsets are called candidate itemsets.

In many cases, the algorithms generate an extremely large number of association rules, often in thousands or even millions [15]. Most of the proposed algorithms focus on the problem of mining frequent itemsets with frequency greater than a support threshold. Normally the subjective and objective measures are used to mine strong and interesting rules. Objective measures based on the statistical strengths or properties of the discovered patterns and subjective measures which are derived from the user's beliefs or expectations of their particular problem domain [25]. However, it is difficult for the users to determine the threshold value. If the threshold is set too large, then there are no frequent itemsets in the output. If the threshold is set too small, then there are too many frequent itemsets in the output. Finding a suitable threshold is a difficult task [22]. Many techniques for association rule mining require a suitable metric to capture the dependencies among variables in a data set. For example, measures such as subjective and objective are often used to determine the interestingness of association patterns. However, many such measures provide conflicting information about the interestingness of a pattern, and the best metric to use for a given application domain is rarely known [23]. It is nearly impossible for the end users to comprehend or validate such large number of complex association rules, thereby limiting the usefulness of the data mining results.

Several strategies have been proposed to reduce the number of association rules, such as generating only "interesting" rules [15], generating only "non redundant" rules[20], or generating only those rules satisfying certain other criteria such as coverage, leverage, lift or strength..

The proposed algorithm uses a partition based technique to extract the frequent itemset. It is created with no overlapping partitions. Frequent itemsets and rules form the first partition. The next two partitions are used to validate rules derived from first partition. Association rule mining extracts the rules from complete

transactions and mines rules from other partitions. This algorithm creates new set of partition to mine rules. The process is repeated for a user specified times for cross validation. This algorithm is called as nVApriori (n-cross Validation based Apriori) algorithm. The proposed nVApriori algorithm mines the frequent itemsets using Apriori like candidate generation. The proposed nVApriori algorithm is implemented and tested with synthetic datasets and real datasets. It shows a better performance compared to Apriori, most interesting rule mining algorithm and non redundant algorithm.

This paper proposes the n-cross validation based Apriori algorithm for mining interesting rules. The organization of the paper is as follows. Section 2 provides the definition of association rule mining and extensions of the Apriori algorithm. The existing algorithm is detailed in section 3. Section 4 describes the n-cross validation technique. Section 5 discusses the proposed nVApriori algorithm. Section 6 exemplifies the proposed algorithm with samples. Section 7 presents the results and discussion of proposed nVApriori algorithm and a comparative study between earlier algorithms. Section 8 concludes the paper and discussed the future enhancement.

2 Association rule mining

2.1 Definition

This paper uses the standard definition of association rules [1], [11], [12]. Let D be a set of n transactions such that D=$\{T_1, T_2, T_3,..,T_n\}$, Where $T_i \subseteq I$ and I is a set of items, $I = \{i_1, i_2, i_3, .. ,i_m\}$. A subset of I containing k items is called a k-itemset. Let x and y be two itemsets such that $x \subset I$, $y \subset I$, and $x \cap y = \phi$. An association rule is an implication denoted by $x \Rightarrow y$ where x is called antecedent and y is called the consequent.

This section proceeds to define association rule metrics. Given an itemset x, support $Supp(x)$ is defined as the fraction of transactions $T_i \in D$ such that $x \subseteq T_i$. Consider P(x) the probability of appearance of x in D, and P($y|x$) the conditional probability of appearance of y given x. P(x) can estimated as P(x)= $Supp(x)$. The support of a rule $x \Rightarrow y$ is defined as $Supp(x \Rightarrow y) = Supp(x \cup y)$. An association rule $x \Rightarrow y$ has a measure of reliability called the confidence, defined as $Conf(x \Rightarrow y) = \frac{Supp(x \Rightarrow y)}{Supp(x)}$. Confidence can be used to estimate $P\left(\frac{y}{x}\right) = \frac{P(x \cup y)}{P(x)} = Conf(x \Rightarrow y)$. This paper uses third metric called lift, defined as $Lift(x \Rightarrow y)\frac{P(x \cup y)}{P(x)P(y)} = \frac{Conf(x \Rightarrow y)}{Supp(y)}$. Lift quantifies the relationship

between x and y. In general, a lift value greater than 1 provides strong evidence that x and y depend on each other. A lift value below 1 state x depends on the absence of y or vice versa. A lift value close to 1 indicates x and y are independent.

The problem of mining association rules is defined as finding the set of all rules $\{x\} \Rightarrow \{y\}$ such that $Supp(x \Rightarrow y) \geq \alpha$, $Conf(x \Rightarrow y) \geq \beta$ and $Lift(x \Rightarrow y) \geq \gamma$, given support threshold, confidence threshold and lift threshold. An itemset X (means $x \cup y$) such that $Supp(X) \geq \alpha$ is called frequent. An association rule $x \Rightarrow y$ such that $Supp(X) \geq \alpha$, $Conf(x \Rightarrow y) \geq \beta$ and $Lift(x \Rightarrow y) \geq \gamma$ is called valid association rule.

2.2 Apriori and its extensions

Since there are usually a large number of distinct single items in a typical transaction database, and their combinations may form a very huge number of itemsets, it is challenging to develop scalable methods for mining frequent itemsets in a large transaction database. Apriori [1] observed an interesting downward closure property, among frequent k-itemsets: A k-itemset is frequent only if all of its sub-itemsets are frequent. This implies that frequent itemsets can be mined by first scanning the database to find the frequent 1-itemsets, then using the frequent 1-itemsets to generate candidate frequent 2-itemsets, and check against the database to obtain the frequent 2-itemsets. This process iterates until no more frequent k-itemsets can be generated for some k. This is the essence of the Apriori algorithm [1] and its alternative [9].

From the day Apriori algorithm was proposed, there have been extensive studies on the improvements or extensions of Apriori, e.g., hashing technique [13], partitioning technique [17], sampling approach [18], dynamic itemset counting [3], incremental mining [4], parallel and distributed mining [14], [2], [5], [19], and integrating mining with relational database systems [16]. The tight upper bound of the number of candidate patterns derived the association rules. That can be generated in the level-wise mining approach [6]. This result is effective at reducing the number of database scans.

3 Existing algorithms

3.1 Most interesting rule mining algorithm (MIR)

Roberto J. Bayardo Jr. and Rakesh Agrawal proposed the most interesting rule mining algorithm. They showed that a single and simple concept of rule goodness captures the best rules the concept involves a partial order on rules defined in terms of both rule support and confidence. Their paper also demonstrated that a set of rules optimal according to the partial order includes all rules that are the best based on any of the above metrics. In the context of mining conjunctive

association rules, this paper presented an algorithm that can efficiently mine an optimal set based on a partial order on a variety of real-world data-sets.

3.2 Non redundant rule mining algorithm (NRRM)

Mohammed J. Zaki presented a new framework for association rule mining based on the novel concept of closed frequent itemsets. The set of all closed frequent itemsets can be orders of magnitude smaller than the set of all frequent itemsets, especially for real datasets. At the same time, it doesn't lose any information; the closed itemsets uniquely determine the set of all frequent itemsets. It showed that the framework produces exponentially (in the length of the longest frequent itemset) fewer rules than traditional approaches and without loss of information. The framework allowed us to mine even dense datasets, where it was not possible to find all frequent itemsets. Experiments on several "hard" databases confirmed the utility of the framework in terms of reduction in the number of rules presented to the user and time taken for the same.

3.3 Discovery of unexpectedness association rules

Balaji Padmanabhan and Alexander Tuzhilin [26] proposed methods of discovery unexpected patterns that take into consideration prior background knowledge of managers. This prior knowledge constitutes a set of expectations or beliefs that managers have about the problem domain. It uses these beliefs to seed the search for patterns in data that contradict the beliefs. Patterns contradictory to prior knowledge are by definition unexpected. The proposed algorithm is called ZoomUAR algorithm which is used to discover unexpected association rules. That consists of two parts: ZoominUAR and ZoomoutUAR. ZoominUAR discovers all significant rules associated with belief. ZoomoutUAR considers each unexpected rule generated by ZoominUAR and tries to determine all the other more general rules that may be unexpected. This method is successfully reduced so many rules compare to Apriori. But the time complexity is higher than Apriori. In this method belief is major role to construct unexpected rules. It increased the dataset size.

4 n-cross Validation Technique

The n-cross Validation technique based approach is extended on the train/test/ validates approach. It normally collects disjoint (independent) samples from a base data set to build and tune predictive (supervised) models [10]. This paper applied a reliable "train/test/validate" approach that required three independent samples [7]. It built a predictive model with a train sample and then validated the model using an independent test sample. The process had the goals of reducing model over fit, providing a realistic estimate in model accuracy and improving generalization when the model is used on new data.

This paper applies the aforesaid idea by partitioning D into three disjoint subsets. The first subset called as the train set D^{tr} ,the second subset is called the test set D^{te} and the third subset called validate set D^{va}. This paper also introduces two fractional values: train fraction χ percentage to control the size of the train set and test fraction ψ percentage to control the size of the test set. The remaining itemsets are stored in the D^{va} .

Therefore

$$\left|D^{tr}\right| = \chi \times \left|D\right|$$

$$\left|D^{te}\right| = \psi \times (\left|D\right| - \left|D^{tr}\right|$$

$$\left|D^{va}\right| = \left|D\right| - (\left|D^{te}\right| + \left|D^{tr}\right|)$$

$$D = D^{tr} \cup D^{te} \cup D^{va}$$

and

The D^{tr} , D^{te} and D^{va} are none overlapping partition. That means $D^{tr} \cup D^{te} \cup D^{va} = \phi$

Example : Consider the following dataset. It contains 30 transactions and 6 attributes. The table 1 contains the sequence of transaction sets. The proposed nVApriori algorithm shuffles the transaction set as shown in the table 2. Forty percent was assigned to the train fraction χ and 30% to the test fraction ψ. D^{tr} is contained the following transaction sets : D^{tr} ={T_1, T_{16}, T_{11}, T_{29}, T_{15}, T_6, T_{18}, T_8, T_{30}, T_{25}, T_3, T_{28}}. The D^{te} is to contain {T_{13}, T_{23}, T_{22}, T_2, T_{17}}. Finally the remaining transaction set {T_7, T_{19}, T_{21}, T_{20}, T_5, T_{14}, T_{24}, T_{10}, T_{26}, T_{27}, T_{12}, T_4, T_9} are assigned in validate set D^{va}.

Table 1 : Sequential Dataset

T_1: a,b,c,f	T_{11}: a,b,c,d,e	T_{21}: a,b,c,d,e
T_2: a,b,d,	T_{12}: a,b,e	T_{22}: c,d,e
T_3: b,d,e	T_{13}: b,d,e,g	T_{23}: b,e,f,g
T_4: a,b,e,f,g	T_{14}: a,c,d,e,f	T_{24}: a,c,d,e,f
T_5: a,c,d	T_{15}: c,d,e,f	T_{25}: b,c,d,e,f
T_6: a,b,c,e,f	T_{16}: a,d,e,f	T_{26}: a,d,e
T_7: a,c,d,e	T_{17}: b,c,d,e,f	T_{27}: a,b,c,d,e
T_8: b,d,e	T_{18}: a,b,c,e	T_{28}: a,c,d,e,g
T_9: c,d,e,f,g	T_{19}: a,c,e	T_{29}: b,c,d,e,f
T_{10}: c,d,e	T_{20}: b,c,d,g	T_{30}: a,c,d,e

Table 2 : Shuffled Dataset

T_1: a,b,c,f	T_3: b,d,e	T_{20}: b,c,d,g
T_{16}: a,d,e,f	T_{28}: a,c,d,e,g	T_5: a,c,d
T_{11}: a,b,c,d,e	T_{13}: b,d,e,g	T_{14}: a,c,d,e,f
T_{29}: b,c,d,e,f	T_{23}: b,e,f,g	T_{24}: a,c,d,e,f
T_{15}: c,d,e,f	T_{22}: c,d,e	T_{10}: c,d,e
T_6: a,b,c,e,f	T_2: a,b,d	T_{26}: a,d,e
T_{18}: a,b,c,e	T_{17}: b,c,d,e,f	T_{27}: a,b,c,d,e
T_8: b,d,e	T_7: a,c,d,e	T_{12}: a,b,e
T_{30}: a,c,d,e	T_{19}: a,c,e	T_4: a,b,e,f,g
T_{25}: b,c,d,e,f	T_{21}: a,b,c,d,e	T_9: c,d,e,f,g

5 Proposed nVApriori algorithm

Input : Dataset (D),
Minimum support Threshold (α),
Minimum confidence Threshold (β),
Minimum lift Threshold (γ),
Number of times train/test/validate (n),
Train sample fraction (χ),
Test sample fraction (ψ)

Output : R rules

Step 1: For $I = 1$ to n do
Step 2: Create the partition D^{tr} , D^{te} and D^{va} based on χ and ψ
Step 3: Generate 1- itemset
Search frequent k – itemsets X on D^{tr} for $k \in \{1 \ldots k\}$
Compute train_support $Supp(X, D^{tr})$ using (1)
Step 4: Generate rules from the generated frequent item sets X.
For each rule $x \Rightarrow y \in D^{tr}$
Compute train_confidence $Conf(x \Rightarrow y)$ on D^{tr} using (2)
Compute train_lift $Lift(x \Rightarrow y)$ on D^{tr} using (3)
Step 5: Let the rules set be R^{tr}.
Eliminate rules from R^{tr}
Such that
$Supp(x \Rightarrow y) < \alpha$ or $Conf(x \Rightarrow y) < \beta$ or $Lift(x \Rightarrow y) < \gamma$
// validate the rules using test set //

Step 6: Validate rules R^{tr} on D^{te}.
Let set $R^{te} = R^{tr}$
For each frequent itemset X means $X = (x \cup y \in R^{te})$
Compute test_support $Supp(X, D^{te})$ using (4)
For each rule $x \Rightarrow y \in R^{te}$
Compute test_confidence $Conf(x \Rightarrow y)$ on D^{te} using (5)
Compute test_lift $Lift(x \Rightarrow y)$ on D^{te} using (6)
Eliminate rules from R^{te}
Such that
$Supp(x \Rightarrow y) < \alpha$ or $Conf(x \Rightarrow y) < \beta$ or $Lift(x \Rightarrow y) < \gamma$

// validate the rules using validate set //
Step 7: Validate rules R^{te} on D^{va}
Let set $R^{va}=R^{te}$
For each frequent itemset X means $X = (x \cup y \in R^{va})$
Compute validate_support $Supp(X, D^{va})$ using (7)
For each rule $x \Rightarrow y \in R^{va}$
Compute validate_confidence $Conf(x \Rightarrow y)$ on D^{va} using (8)
Compute validate_lift $Lift(x \Rightarrow y)$ on D^{va} using (9)
Eliminate rules from R^{va}
Such that
$Supp(x \Rightarrow y) < \alpha$ or $Conf(x \Rightarrow y) < \beta$ or $Lift(x \Rightarrow y) < \gamma$
Finally
Let the rules set be $R_I = R^{va}$
Next I
Step 8: Get intersection of n rule sets and compute the average rule metrics with (10), (11) and (12)

$$R = R_1 \cap R_2 \cap R_3 \cap \ldots \cap R_n$$

The proposed nVApriori algorithm needs the following inputs: the dataset (D) which containes the transaction sets. Minimum support (α), Minimum confidence (β) and Minimum lift (γ) control frequent itemsets and association rules. Train sample fraction (χ) and test sample fraction (ψ) are used to manage the size of partitions. Finally this algorithm requires the input n to repeat the cross validations. The output of algorithm produces strong and interesting rules reliable to the transaction sets. Step 2 is a small procedure to create three transaction set partitions. This procedure uses the Fisher-Yates shuffle algorithm [21] to shuffle the transaction sets. The Fisher–Yates shuffle is unbiased, to make every permutation equal. The modern version of the algorithm is also efficient, requiring only time proportional to the number of items being shuffled with no additional storage space.The basic process of Fisher–Yates shuffling is similar to randomly

picking transaction id from transaction id set, one after another until there are no more left. The algorithm provides a way of calculating this numerically in an efficient and rigorous manner with guaranteed and unbiased results. It has O(n) time complexity to shuffle the transaction set.

The shuffled transaction set is then divided into three partitions in the manner of logical non overlapping randomized or non randomized partitions. These three partitions are called as D^{tr}, D^{te} and D^{va}. The D^{tr} are controlled by the train sample fraction χ percentage from the shuffled transaction set. The remaining shuffled transaction set divided into test sample fraction ψ percentage. That partition is called D^{te}. The criteria of test sample fraction as follows: $\psi \leq \chi$. The size of D^{tr} and D^{te} are discussed in the section 3. This algorithm is not considered the size of partition D^{va}. In step 3, the set of frequent 1-itemsets is found. This set is denoted L_1. L_1 is used to find L_2, the set of frequent 2-itemsets, which is used to find L_3, and so on, until no more frequent k-itemsets can be found, and then algorithm ceases. In the cycle k, a set of candidate k-itemsets is generated at first. This set of candidates is denoted C_k. It computes the train_support for frequent itemset using (1) which is called train_support. If the train_support is not satisfied minimum support (α), that frequent itemset eliminate from list to further process. The remaining frequent itemset are used to construct further frequent itemset or association rule mining. Step 4 is to construct the association rule from the frequent itemsets which are found from the D^{tr}.

$$train\_Supp(x \Rightarrow y) = \frac{|D_{xy}^{tr}|}{|D^{tr}|} \text{.....(1)}.$$

After the association rules are mined, it calculates the train_confidence and train_lift using (2) and (3). Such that it eliminates the rules which are train_confidence($x \Rightarrow y$)<β or train_lift($x \Rightarrow y$)<γ .In step 5, the remaining rules are set into the R^{tr}.

$$train\_conf(x \Rightarrow y) = \frac{|D_{xy}^{tr}|}{|D_{x}^{tr}|} \text{.....(2)}$$

$$train\_lift(x \Rightarrow y) = \frac{|D^{tr}| \times |D_{xy}^{tr}|}{|D_{x}^{tr}| \times |D_{y}^{tr}|} \text{.....(3)}$$

Step 6 and step 7 are going to validate the derived rules R^{tr} with D^{te}. Step 6, Let the R^{tr} set to be R^{te}. this step find the frequent itemset form each rules and compute the test_support threshold for the frequent itemset on D^{te} using (4).

$$test\_Supp(x \Rightarrow y) = \frac{|D_{xy}^{te}|}{|D^{te}|} \text{.....(4)}$$

Also it computes test_confidence and test_lift for the rules on D^{te} using (5) and (6). The new metrics are assigned to the rules. These rules are eliminated which are not satisfied the following values α,β and γ.

$$test\_conf(x \Rightarrow y) = \frac{|D_{xy}^{te}|}{|D_{x}^{te}|} \text{.....(5)}$$

$$test\_lift(x \Rightarrow y) = \frac{|D^{te}| \times |D_{xy}^{te}|}{|D_{x}^{te}| \times |D_{y}^{te}|} \text{.....(6)}$$

Step 7, after the elimination of unsatisfied rules, those rules are stored in the R^{va}. It also same as step 6 but it used the D^{va} to compute the validate_support, validate_confidence and validate_lift using (7), (8) and (9). The rules are filtered with modified measures which are not satisfied α, β and γ.

$$validate\_Supp(x \Rightarrow y) = \frac{|D_{xy}^{va}|}{|D^{va}|} \text{.....(7)}$$

$$validate\_conf(x \Rightarrow y) = \frac{|D_{xy}^{va}|}{|D_{x}^{va}|} \text{.....(8)}$$

$$validate\_lift(x \Rightarrow y) = \frac{|D^{va}| \times |D_{xy}^{va}|}{|D_{x}^{va}| \times |D_{y}^{va}|} \text{.....(9)}$$

These rules are set in the R_I. These steps 1-7 are repeated n times to achieve cross validation of rules. Finally the algorithm produced n set of rules. Repeated rules are eliminated from the list and it computes the average of metric for rules using (10), (11) and (12). These rules are called most interesting strong and valid to transaction sets.

$$Supp(x \Rightarrow y) = \frac{1}{t}\sum_{i=1}^{t} Supp[(x \Rightarrow y), Di] \text{.....(10)}$$

$$Conf(x \Rightarrow y) = \frac{1}{t}\sum_{i=1}^{t} Conf[(x \Rightarrow y), Di] \text{.....(11)}$$

$$Lift(x \Rightarrow y) = \frac{1}{t}\sum_{i=1}^{t} Lift[(x \Rightarrow y), Di] \text{.....(12)}$$

6 nVApriori with sample dataset

Manually, the proposed algorithm was tested with the above dataset. The dataset contained 30 transactions and 6 items. The process set 20% as the minimum support, the train sample fraction was set to 40%, test fraction threshold 30% and

remaining set of 40% transaction stored in D^{va}. This section demonstrates the frequent itemset mining as follows: i=5, the frequent itemset mining produced the *c*,*d*,*e* itemsets from D^{tr}. This frequent itemset *c*,*d*,*e* validates with D^{te} and D^{va}. The *c*,*d*,*e* frequent itemset satisfied the minimum support threshold on D^{te} and D^{va}. So the *c*,*d*,*e* frequent itemset is called the transaction relevant frequent itemset.

7 Results and Discussion

All experiments described below were performed on a 1.6 GHz Intel Celeron Dual-core PC with 1GB of main memory and 160GB of HDD, running Microsoft windows vista. This paper used its own implementations MIR and NRRM algorithms and was coded in Java.

Table 3: Datasets

Dataset	**# of Rows**	**# of Columns**
T40I10D100K	100000	942
Mushroom	8124	119
Chess	3196	37
Heart Disease Prediction	655	25

Table 3 shows the characteristics of the real and synthetic datasets used in this evaluation. All datasets are taken from the UC Irvine Machine Learning Database Repository [24]. Typically, these real datasets are very dense, i.e., they produce many long frequent itemsets even for very high values of support. These datasets mimic the transactions in a retailing environment. Usually the synthetic datasets are sparse when compared to the real sets, but this work modified the generator to produce longer frequent itemsets.

The candidate generation and the support counting processes require an efficient data structure in which all candidate itemsets are stored since it is important to efficiently find the itemsets that are contained in a transaction or in another itemset. The proposed nVApriori algorithm implemented a hash-tree data structure. The time complexity and space complexity of hash tree is O(1). In order to efficiently find all k-subsets of a potential candidate itemset, all frequent itemsets in L_k are stored in a hash table. Candidate itemsets are stored in a hash-tree [4]. A node of the hash-tree either contains a list of itemsets (a leaf node) or a hash table (an interior node). In an interior node, each bucket of the hash table points to another node. The root of the hash-tree is defined to be at depth 1. An interior node at depth d points to nodes at depth $d+1$. Itemsets are stored in leaves. When it adds a k-itemset X during the candidate generation process, it starts from the root and go down the tree until the process reach a leaf. At an interior node at

depth *d*, it decide which branch to follow by applying a hash function to the X[*d*] item of the itemset, and following the pointer in the corresponding bucket. All nodes are initially created as leaf nodes. When the number of itemsets in a leaf node at depth *d* exceeds a specified threshold, the leaf node is converted into an interior node, only if $k > d$. In order to find the candidate-itemsets that are contained in a transaction T, it starts from the root node. If it is at a leaf, this paper finds which of the itemsets in the leaf are contained in T and increment their support. If that are at an interior node and the process have reached it by hashing the item *i*, It hash on each item that comes after *i* in T and recursively apply this procedure to the node in the corresponding bucket. For the root node, it hash on every item in T.

The proposed algorithm is tested with simple heart disease prediction dataset which contain 655 transactions and 25 items. The effect of filtering association rules are based D^{te} and D^{va} partitioned datasets. The association rules are tested for generality and validity by partitioning the input dataset into D^{tr}, D^{te} and D^{va}. Building the D^{tr}, D^{te} and D^{va} samples is repeated several times. This association rule algorithm produces different sets of rules with different samples, where each set of rules has different rules have slightly lower or higher metrics. The rules are valid on three samples. This paper extends the definition of association rules, given in Section 2.1, to have three sets of metrics per rule based on D^{tr}, D^{te} and D^{va}. That is, each rule has nine metrics in total. In general, metrics on the training set are used only for search purposes, and metrics on the test set are used to validate rules and are taken as the actual rule metrics.This proposed algorithm computes three sets of rules, R^{tr} on D^{tr} ,$R^{te} \subseteq R^{tr}$ such that R^{te} also has metrics above α, β, γ on D^{te} and $R^{va} \subseteq R^{te}$ such that R^{te} also has metrics above α, β, γ on D^{va}. The computation of R^{te} is as follows. Each association has three sets of metrics, one for D^{tr} one for D^{te} and one for D^{va}. This method search association rules based on D^{tr} to get R^{tr} based on thresholds α, β and γ. This approach sets R^{te}=R^{tr}. This process then compute support, confidence, and lift for each rule in R^{te} based on D^{te}. Rules whose test metrics on D^{te} are below α, β, γ are filtered out from R^{te}. This process is repeated a number of times (n) to achieve basic cross validation and to eliminate rules that cannot be generalized.

The association rules are tested for generality and validity by partitioning the input data set into a D^{tr}, D^{te} and D^{va}. A valid rule must have minimum metrics on D^{te} and D^{va}. The experiments show the importance of filtering rules on the test by varying *k*. Table 2 summarizes result. The reduction is small in the numbers of association, with a reduction of about 10% - 15%. The reduction becomes much more important for the number of rules. For k =2, the impact is small in most cases, which indicates most rules can be generalized. For k =3, the reduction is more than 50%, providing evidence that many rules one particular to the D^{tr}. At k = 4, the number of rules in D^{tr} is about 30% of the total with a reduction of about 70%, providing evidence that most rules may be particular to D^{tr}. The trend

indicates there will a combination explosion of rules that are valid only on D^{tr} to achieve lesser memory usage and time efficiency.

The difference in the relative number of patterns for associations and rules can be explained by the fast that association and filtered D^{tr},D^{va} and D^{te} based only one support, but rules require support, confidence and lift to be greater than or equal to the respective thresholds in D^{tr} and D^{te}. This kind of validation methodology made the database depended rules to compare with other methodology. The previous research works are attempted based on relationship or correlation analysis of transactions but the proposed work implies database dependent rules. The following tables 4 and 5 retrieved from implementation of proposed nVApriori algorithm with heart disease prediction dataset.

Table 4 : Number of associations and rules in D^{tr}, D^{te} and D^{va}

K	Minsup	# of associations in			# of rules			Number of time
		D^{tr}	D^{te}	D^{va}	D^{tr}	D^{te}	D^{va}	
2	0.01	493	467	415	8	5	3	17
3	0.01	3286	2948	2272	145	77	50	60
4	0.01	11610	9327	7044	1222	342	281	258

Table 5 : Number of rules in D^{tr} D^{te} and D^{va} sets varying minimum support

K	Minsup	D^{tr}	D^{te}	D^{va}	Time (in sec)
4	0.100	62	33	30	5
4	0.050	163	114	107	12
4	0.010	1222	342	300	43
4	0.005	2022	497	450	47

Table 5 contains summary of the results. At high support levels, the reduction in the number of rules is about 40% for low support levels, the number of rules goes down to less than 35%. This indicates that as because they do not meet the minimum metrics in the test sets. The last column in Table 5 contains total elapsed times in seconds. Time growth is not as fast compared to varying *k*; because test and validate set significantly reduces number of patterns. The proposed algorithm also tested with the T40I10D100K, Mushroom, Chess and heart disease prediction. The proposed nVApriori algorithm is compared with traditional Apriori, MIR and NRRM algorithm. Table 6 shows the comparative study of no. of association rule mined from T40I10D100K, Mushroom, Chess and

heart disease prediction dataset. The minimum support set as 50% and the minimum confidence set as 50%. The results are summarized in the Table 6. It reduced the 50% rules compared with non redundant algorithm, 42% rules are reduced from most interesting rule mining algorithm and 80% of rules reduced to compare with traditional Apriori. Finally, these rules have fairly high support, borderline confidence, and small lift. The nVApriori algorithm is mined some many rules with low support threshold. The proposed nVApriori algorithm is considered to mine strong rule with high support and confidence.

Table 6: No. of association rules in Apriori, MIR, NRRM and proposed nVApriori with minimum_supp=50%

Datasets	Apriori	MIR	NRRM	nVApriori
T40I10D100K	40321091	678542	22700	11209
Mushroom	18192345	409879	15632	11789
Chess	8171198	236735	151000	8910
Heart disease prediction	3210	2470	1056	710

The comparative study of proposed nVApriori algorithm is shown is figure 1-4. These figures are showed the comparative study of traditional Apriori, most interesting rule mining algorithm, non redundant rule mining and proposed nVApriori algorithm with T40I10D100K, mushroom, chess and heart disease prediction. These four datasets are having different transaction size, item size and other behaviors. So that the reason this selected to make the experimental studies. The figure's x axis has support level 10 to 100 percentages and y axis is carrying execution time in seconds. This experimental study considered the minimum support to count execution time. It does not consider the confidence and lift. But this work fix the minimum confidence =50% and lift=20% to count the execution time. From this experimental study, the proposed nVApriori algorithm is performed well as compared to traditional Apriori, MIR algorithm and NRRM algorithm. This work also concluded that the proposed nVApriori algorithm is better algorithm to mine transaction relevant rules from synthetic and real datasets.

Figure 1: Comparative study of execution time of T40I10D100K dataset

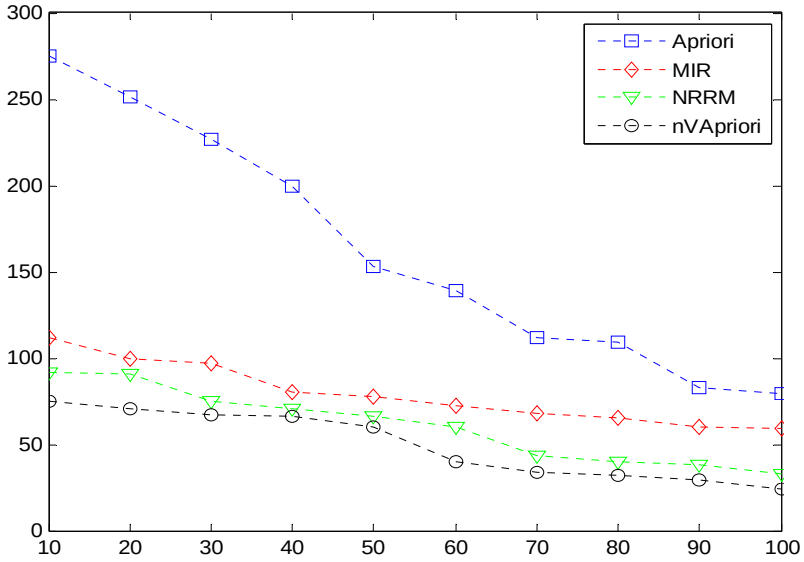


Figure 2: Comparative study of execution time of mushroom dataset

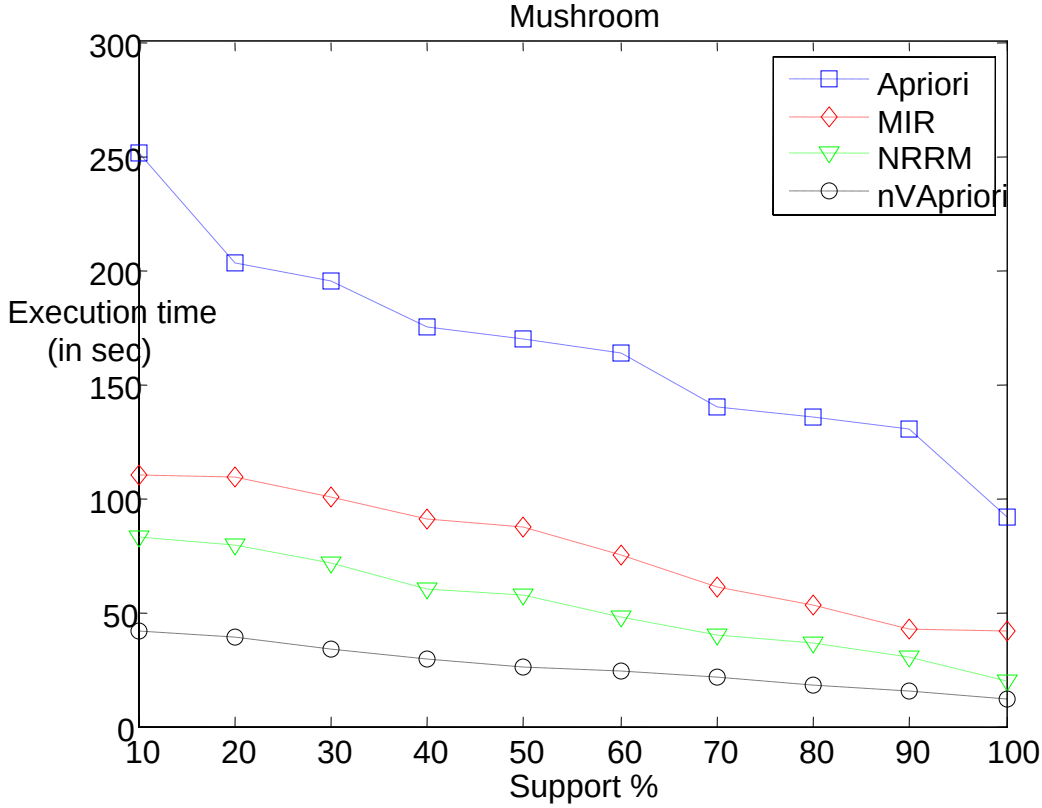


Figure 3: Comparative study of execution time of Chess dataset

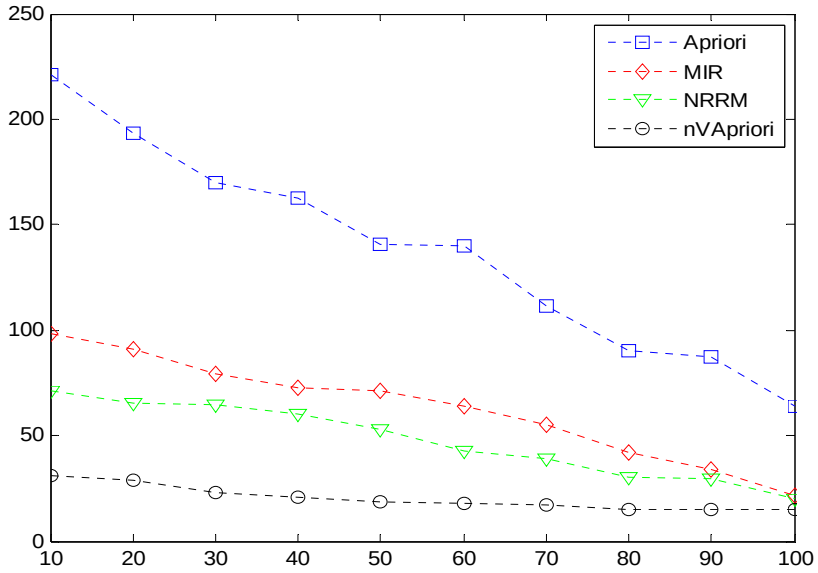
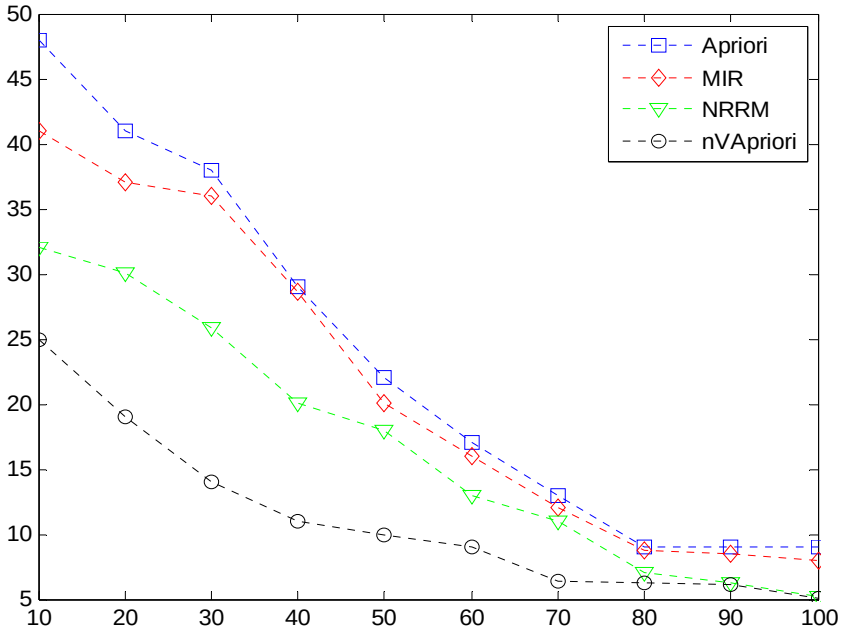


Figure 4: Comparative study of execution time of Heart disease prediction dataset



8 Conclusion and Future work

This paper focused on two main research issues. The first of which was the increasing number of rules obtained by standard association rule algorithms. The second was rules of validation on an independent set. These rules required to eliminate as unreliable, or rules that could not be generalized. In order to validate rules, this paper used the train-test-validate approach that uses three disjoint samples from a data set to search and validate rules. The proposed algorithm performs several train, test and validate cycle achieve the relevant rules. Experiments on T40I10D100K, Mushroom, Chess and heart disease prediction, these dataset are studied the impact of constraints and elimination of unreliable rules with validation on the test set. The reduction in output size provided by validation is significant. These methods may be more efficient compared to traditional Apriori, non redundant rule mining framework and most interesting rule mining and save the time for irrelevant rule findings from the dataset. In the future this algorithm can be applied to parallel and distributed environments.

References

[1] Agrawal, R., Imielinski, T., and Swami, A. N, (1993), Mining association rules between sets of items in large databases, In proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp:207-216.

[2] Agrawal R and Shafer JC, (1996), Parallel mining of association rules: design, implementation, and experience, IEEE Transaction Knowledge and Data Engineering, pp:962–969.

[3] Brin S, Motwani R, Ullman JD and Tsur S, (1997), Dynamic itemset counting and implication rules for market basket analysis, In proceeding of the 1997 ACM-SIGMOD International conference on management of data (SIGMOD'97), Tucson, AZ, pp:255-264.

[4] Cheng H, Yan X and Han J, (2004), IncSpan: incremental mining of sequential patterns in large, In proceedings of the 2004 ACM SIGKDD International conference on knowledge discovery in databases (KDD'04), Seattle, WA, pp:527–532.

[5] Cheung DW, Han J, Ng V, Fu A and Fu Y, (1996), A fast distributed algorithm for mining association rules, In proceedings of the 1996 International conference on parallel and distributed information systems, Miami Beach, FL, pp:31–44

[6] Geerts F, Goethals B and Bussche J, (2001), A tight upper bound on the number of candidate patterns, In proceedings of the 2001 International conference on data mining (ICDM'01), San Jose, CA, pp:155–162

[7] Hastie.T, Tibshirani.R, and Friedman J.H, (2001), The Elements of Statistical Learning, 1st ed., New York: Springer-Verlag.

[8] Jiawei,Han and Micheline Kambar, (2003), Data Mining: concepts and techniques, Morgan Kaufman Publishers, San Francisco.

[9] Mannila.H, Toivonen . H, and Verkamo.I, (1994), Efficient algorithms for discovering association rules, Knowledge Discovery in Databases (KDD'94), AAAI Press, pp.181-192.

[10] Mitchell.T.M, (1997) Machine Learning. New York: McGraw-Hill.

[11] Ordonez. C and Omiecinski. E,(1999), Discovering association rules based on image content, In proceedings IEEE Advances in Digital Libraries Conference (ADL'99), pp:38–49.

[12] Ordonez. C, Santana C.A, and Braal.L, (2000), Discovering interesting association rules in medical data, in proceedings ACM Data Mining and Knowledge Discovery Workshop, pp:78–85.

[13] Park JS, Chen MS and Yu PS, (1995), An effective hash-based algorithm for mining association rules, In: Proceeding of the 1995 ACM-SIGMOD international conference on management of data (SIGMOD'95), San Jose, CA, pp:175–186.

[14] Park JS, Chen MS and Yu PS, (1995), Efficient parallel mining for association rules. In: Proceeding of the 4th international conference on information and knowledge management, Baltimore, MD, pp:31–36.

[15] Roberto J. Bayardo Jr and Rakesh Agrawal, (1999), "Mining the most interesting rules" Appears in Proc. of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp:145-154.

[16] Sarawagi S, Thomas S and Agrawal R, (1998), Integrating association rule mining with relational database systems: alternatives and implications. In: Proceeding of the 1998 ACM-SIGMOD international conference on management of data (SIGMOD'98), Seattle, WA, pp:343–354.

[17] Savasere A, Omiecinski E and Navathe S, (1995), An efficient algorithm for mining association rules in large databases. In: Proceeding of the 1995 international conference on very large data bases (VLDB'95), Zurich, Switzerland, pp:432–443.

[18] Toivonen H, (1996), Sampling large databases for association rules, In proceeding of the 1996 international conference on very large data bases (VLDB'96), Bombay, India, pp:134–145.

[19] Mohammed J. Zaki, Parthasarathy S, Ogihara M and Li W, (1997), Parallel algorithm for discovery of association rules, Data mining knowledge discovery, pp:343–374.

[20] Mohammed J. Zaki, (2004), Mining Non-Redundant Association Rules, International journal of Data Mining and Knowledge Discovery, Vol. 9, Issue 3, pp:223-248.

[21] Fisher, R.A and Yates, F. (1948) , Statistical tables for biological, agricultural and medical research (3rd ed.). London: Oliver & Boyd. pp. pp:26–27.

[22] S-C. Ngan, T. Lam, R.C-W. Wong and A.W-C. Fu, (2005), Mining N-most interesting itemsets without support threshold by the COFI-tree, International Journal of Business Intelligence and Data Mining, Vol. 1, No. 1, pp:88-108

[23] Pang-ning Tan, Vipin kumar and Jaideep Srivastava, (2002), Selecting the right interestingness measures for association pattern, In proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, Edmonton, pp: 32 – 41.
FIMI dataset . http://fimi.cs.helsinki.fi/

[24] Ken Mcgarry. (2005), A survey of interestingness measures for knowledge discovery, The knowledge engineering review, Vol. 00:0, pp: 1-24.

[25] Balaji Padmanabhan and Alexander Tuzhilin (1999), Unexpectedness as a measure of interestingness in knowledge discovery, Special issue on WIT'97, Vol.27, Issue.3, pp.303-318.