AN IMPROVISED METHOD AND TECHNIQUE FOR CLASSIFICATION USING ASSOCIATION RULE MINING

Aswini kumar mohanty
Capital engineering college, khorda

Abstract:- Association rule mining and classification are two important data mining techniques in the knowledge discovery process. The integration of these two techniques is an important research focus and has many applications in data mining. The integration of these two techniques created new approaches called Class Association Rule Mining or Associative Classification Technique. The two combined approaches provide better classification accuracy when classifying data. Content-based information retrieval research areas require high efficiency and performance. In these applications, association rule mining detects association patterns from data, and we classify target classes based on the association patterns. Our paper mainly focuses on the combination of classification and association rule mining for accurate data classification. In this paper, we proposed to implement two new algorithms CPAR (Classification Based on Predictive Association Rule) and CMAR (Classification Based on Multiple-class Association Rules), which combine the advantages of both associative classification and traditional rule-based classification. Instead of producing a large number of frequent item rules as in associative classification, CPAR adopts a greedy search algorithm to produce rules directly from the training data. In addition, CPAR generates and tests more rules than traditional rule-based classifiers to avoid missing important rules. To avoid over fitting, CPAR uses the expected accuracy to evaluate each rule and uses the best k rules in prediction. CMAR applies a CR-tree structure to efficiently store and retrieve the mined association rules and efficiently prunes the rules based on confidence, correlation, and database coverage. Classification is performed based on a weighted χ2 analysis using several strong association rules. The extensive experiments show that CMAR is consistent, highly effective in classifying different kinds of databases, and has better average classification accuracy compared to FOIL (First Order Inductive Learner) and PRM (Predictive Rule Mining). The proposed algorithms are better in terms of memory requirements, time consumption and eliminate intermediate data structures during implementation.

Keywords: Association Rule Mining, Classification, Data Mining, Knowledge Discovery, FOIL (First Order Inductive Learner), PRM (Predictive Rule Mining), CMAR (Multi-Class Association Rule Based Classification), CPAR (Predictive Association Rule Based Classification), CBA (Classification Based Association).

I. INTRODUCTION

Classification rule mining and association rule mining are two important data mining techniques. Classification rule mining is used to discover a small set of rules in a database to create an accurate classifier. Association rule mining is used to uncover all interesting relationships in a potentially large database. Association rule mimicry finds all rules in the database that meets a certain minimum support and minimum confidence threshold. For association rule mining, the discovery goal is not predetermined, while for classification rule mining, there is only one predetermined goal. These two techniques can be integrated to form a framework called the associative classification method. The integration is performed to obtain a special subset of

association rules whose right-hand side is limited to the classification class attribute. These subsets of rules are referred to as class association rules. The use of association rules for classification is limited to problems where instances can only belong to a discrete number of classes. This is because association rule mining is only possible for categorical attributes. The head Y of any association rule X → Y is a disjunction of items. However, association rules in their general form cannot be used directly. We need to narrow down their definition. Any item not present in the rule body can appear in the rule header. When we want to use rules for classification, we are interested in rules that are able to assign class membership. Therefore, we limit the Y header of the X → Y class association rule to one entry. The attribute of this attribute-value pair must be a class attribute. The class association rule is, of course, a predictive task. Using the discriminative power of class association rules, we can also create a classifier.
Creating accurate and efficient classifiers for large databases is one of the fundamental tasks of data mining and machine learning research. Given a set of cases with class labels as a training set, the classifier is supposed to build a model (called a classifier) to predict future data objects for which the class labels are unknown. Previous studies have developed heuristic/greedy search techniques for building classifiers such as decision trees [10], rule-based learning [2,4,13,18], naive Bayes classification [4,9,17] and statistical approaches [8]. These techniques induce a representative subset of rules (e.g., a decision tree or set of rules) from training datasets for quality prediction. Recent studies propose the extraction of a set of high-quality association rules from a set of training data that meet certain user-specified frequency and confidence thresholds. Effective and efficient classifiers have been created by carefully selecting rules, e.g., CBA [9], CAEP [3], and ADT [11]. Such a method takes the most effective rule(s) from all the rules obtained for classification. Because association rules examine highly reliable associations between multiple variables, they can overcome some of the limitations introduced by the decision tree induction method, which examines one variable at a time. Extensive performance studies [6, 9, 3, 11] shows that association-based classification can generally have better accuracy. In recent years, a new approach called associative classification [7, 6] has been proposed, which integrates association rule mining [1] and classification. It uses an association rule mining algorithm such as Apriori [1] or FPgrowth [5] to generate a complete set of association rules.

It then selects a small set of high-quality rules and uses that set of rules for prediction. Experiments in [7, 6, 18, 20] show that this approach achieves higher accuracy than traditional classification approaches such as C4.5 [8, 14]. In this paper, we propose two new algorithms called CPAR (classification based on predictive association rules) and CMAR (classification based on multiple association rules). CPAR adopts the basic idea of FOIL [9] in rule generation and integrates features of associative classification into predictive rule analysis. Compared with associative classification, CPAR has the following advantages: (1) CPAR generates a much smaller set of high-quality predictive rules directly from the dataset; (2) to avoid generating redundant rules, CPAR generates each rule with respect to a set of \already generated" rules; and (3) when predicting the class label of an example, it uses the best CPAR of the rules that the example satisfies. In addition, CPAR uses the following features to further improve its accuracy and efficiency: (1) CPAR uses dynamic programming to avoid repeated computations when generating rules; and (2) when generating rules, instead of selecting only the best literal, all nearest best literals are selected to avoid omission of important rules. CPAR generates a smaller set of rules with higher quality and lower redundancy compared to associative classification. As a result, CPAR is much more time-efficient in rule generation and prediction, but achieves the same high accuracy as associative classification. CMAR selects a small set of highly reliable, highly related rules and analyzes the correlation between these rules. To avoid bias, we developed a new technique called weighted χ2 , which derives a good measure of how strong the rule is in both conditional support and class partitioning. An extensive performance study shows that CMAR generally has higher prediction accuracy than CBA [9] and C4.5 [10]. Second, to improve accuracy and efficiency, CMAR uses a new data structure, CR-tree, to compactly store and efficiently retrieve a large number of classification rules. CR-tree is a prefix tree structure for exploring sharing between rules, thus achieving considerable compactness. The CR tree itself is also an index structure for rules and effectively serves to retrieve rules. Third, to accelerate the mining of the complete rule set, CMAR adopts a variant of the recently developed FP-growth method. The growth of FP is much faster than Apriori-like methods used in previous association-based classification such as [9, 3, 11], especially when there are huge number of rules, large training data sets, and long pattern rules.

II. RELATED WORK

Data analysis algorithms (or data mining algorithms as they are more popular today) can be divided into three main categories based on the nature of their information acquisition [1]: clustering (also called segmentation or unsupervised learning), predictive modeling (also called classification or learning under by supervision) and frequent pattern extraction. Clustering is a major class of data mining algorithms. The goal of the search process used by these algorithms is to identify all sets of similar examples in the data in some optimal way. One of the oldest clustering algorithms is k-means [2]. Two disadvantages of this algorithm are the initialization problem and the fact that the cluster must be linearly separable. To solve the initialization problem, global k-means [3] was proposed, which is an incremental-deterministic algorithm that uses k-means as a local search procedure. The k-means kernel algorithm [4] avoids the constraint of linearly separable clusters and mapped the data points from the input space to a multidimensional feature using a nonlinear Ø transformation, and k-means is applied in the feature space. The k-means global kernel [5] is an algorithm that mapped data points from the input space to a multidimensional feature space using a kernel function and optimizes the clustering error in the feature space by finding a near-optimal solution.

Due to its deterministic nature, it is independent of the initialization problem and the ability to identify a nonlinearly separable cluster in the input space. Thus, the global kernel k-means algorithm combines the advantages of global k-means and kernel k-means. Another approach for data clustering is hierarchical clustering, which is based on the Hungarian method [6] and the computational complexity of the proposed algorithm is O (n2). Important classification algorithms are decision tree, Naive-Bayes classifier and statistics [2]. They use heuristic search and greedy search techniques to find subsets of rules to find classifiers. C4.5 and Classification And Regression Tree(CART) are the most well-known decision tree algorithms. The last class of data mining algorithms is frequent pattern extraction. For large databases [7] describes an Apriori algorithm that generates all significant association rules between items in the database. The algorithm makes several passes through the database. The boundary set for a pass consists of those sets of items that are extended during the pass. In each pass, support is measured for candidate itemsets that are derived from the tuples in the databases and itemsets contained in the bounds set. Initially, the boundary set consists of only one element, which is the empty set.

At the end of the pass, the support for the candidate item set is compared to minsupport. At the same time, it is determined whether the itemset should be added to the boundary set for the next pass. The algorithm terminates when the boundary set is empty. After finding all itemsets that meet the minsupport threshold, association rules are generated from those itemsets. Bing Liu et al.[8] proposed an association-based classification algorithm (CBA) that discovers class association rules (CARs). It consists of two parts, the rule generator, which is called CBA-RG, is based on the Apriori algorithm for finding association rules, and the classifier, which is called CBA-CB. In the Apriori Algorithm, a set of items (set of items) was used, while in CBA-RG, a set of rules was used, which consists of a condset (set of items) and a class. The class association rules used to create the classifier in [8][9] are more accurate than the C4.5 algorithm [2][3][16]. However, the association-based classification (CBA) algorithm needs an evaluation rule before it can create a classifier. The rating depends on the support and reliability of each rule.

This makes the accuracy of CBA less accurate than classification based on predictive association rules. A neural network is a parallel processing network that generates by simulating the visual intuitive thinking of a person, based on the research of biological neural network according to the properties of biological neurons and neural network, and by simplifying, summarizing and refining [9]. It uses the idea of non-linear mapping, the method of parallel processing and the structure of the neural network itself to express the related knowledge of input and output. At first, the application of neural network in data mining was not optimistic because neural networks can have complex structure, long training time and difficult to understand representation of results. But its advantages such as high affordability of noise data and low error rate, continuous improvement and optimization of various network training algorithms, especially continuous improvement and improvement of various network pruning algorithms and rule extraction algorithms, make the application of neural network in data mining increasingly popular with the vast majority of users. Xianjun Ni [10] describes a neural network-based data mining process. This process consists of three main steps such as data preparation, rule extraction and rule evaluation. Classification is currently considered one of the most common tasks in data mining [14, 20]. Classifying real-world examples is a common thing that everyone practices throughout their lives. One can classify human beings based on their race, or one can categorize products in a supermarket based on consumer purchasing decisions. Classification generally involves examining the properties of new objects and attempting to assign them to one of a predefined set of classes [38]. Given a collection of records in a dataset, each record consists of a group of attributes; one of the attributes is class.

The goal of classification is to build a model from classified objects in order to classify previously unseen objects as accurately as possible. There are many classification approaches for knowledge extraction from data, such as divide and conquer [13], separate and conquer [15], covering and statistical approaches [20, 6]. The divide-and-conquer approach starts by selecting an attribute as the root node and then creates a branch for each possible level of that attribute. This splits the training instance into subsets, one for each possible attribute value. The same process will be repeated until all instances that fall into one branch have the same classification or until the remaining instances cannot be split further.

A segregate and conquer approach, on the other hand, starts by creating rules in a greedy fashion (one at a time). Once a rule is found, all instances that the rule applies to will be removed. The same process is continued till the best rule found has a large error rate. Statistical approaches such as Naïve Bayes [19] use probabilistic measures, i.e. probability, to classify test objects. Finally, the coverage approach [6] sequentially selects each of the available classes and finds a way to cover most of the training objects into that class in order to produce rules with maximum accuracy. Numerous algorithms have been derived from these approaches, such as decision trees [12, 10], PART, RIPPER, and Prism [6]. In studies [14, 7, 6, 19], a little work has been done on multiple level classification. Most of the research to date on multi-label classification is related to text categorization [20]. In this article, it will only consider traditional classification algorithms that generate rules with a single class.

III. DESIGN AND IMPLEMENTATION OF THE SYSTEM

The overall design of the association rule mining system by classification is described in Figure 1.

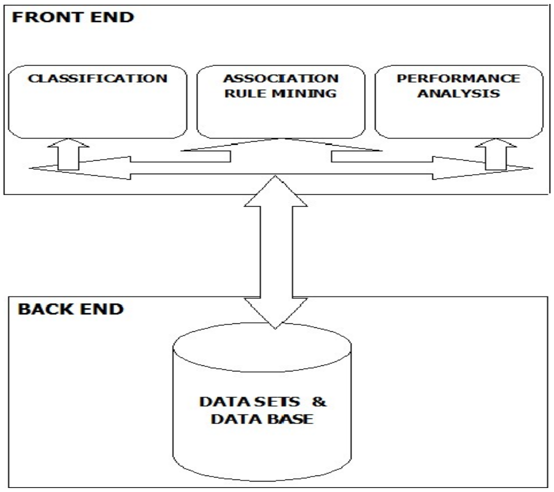

Figure 1. System Architecture

The System is divided into 4 Modules.

A) Data Source/ Data Base Module
B) Classification Module
C) Association Rule Generation Module
D) Performance Analysis Module

A) Data source / database module:

This module stores data in the form of datasets.

Here we have a dataset of several attribute values in the form of transaction records and we have a dataset that contains the schema of the dataset. This schema is useful for classifying data.

B) Classification module:

This module reads data from a dataset and performs classification operations and generated classes.

C) Association rule generation module:

This module uses classes and performs association rule mining and generates frequent itemsets, generates association rules.

D) Performance Analysis Module:

This module calculates the time complexity, space complexity, accuracy, and number of association rules for each implementation based on the number of classes for various algorithms such as CPAR, CMAR, FOIL, and PRM. It then compares their values and analyzes efficient algorithms.

IV. ASSOCIATIVE CLASSIFICATION

Associative classification is a special case of associative rule discovery in which only the class attribute on the right-hand side of the rule (subsequent) is considered; for example, in a rule like X □Y, Y must be a class attribute. One of the main advantages of using classification based on of association rules compared to classic classification approaches is that the output of the associative classification algorithm is represented by simple if–then rules, which makes it easier for the end user to understand and interpret. Moreover, unlike decision tree algorithms, a rule in associative classification can be updated or tuned without affecting the complete set of rules, while the same task requires reshaping the entire tree in a decision tree approach. Let us define an associative classification problem where the training data set T has m different attributes A1, A2, . . ., Am and C contains a list of classes. The number of tuples in T is denoted by |T|. Attributes can be categorical (that is, they take a value from a finite set of possible values) or continuous (where they are real or integer). For categorical attributes, all possible and expected values are mapped to a set of positive integers. A discretization method is used for continuous attributes.

Definition 1 A row or training object in T can be described as a combination of attribute names Ai and values aij, plus a class denoted cj.

Definition 2 An item can be described as an attribute name Ai and a value ai, denoted by <(Ai, ai)>.

Definition 3 A set of items can be described as a set of disjoint attribute values contained in the training object, denoted <(Ai1, ai1), . . ., (Aik, aik)>.

Definition 4 A rule item r has the form <itemset, c>, where c 2 C is a class.

Definition 5 The actual occurrence (actoccr) of a rule item r in T is the number of rows in T that match the item set r. Definition 6 The suppcount of a rule item r is the number of rows in T that match the item sets r and belong to class c in r.

Definition 7 The occurrence of an item set i (occitm) in T is the number of rows in T that match i.

Definition 8 Itemset i exceeds the threshold minsupp if (occitm(i)/|T|)>=minsupp.

Definition 9 A rule entry r exceeds the minsupp threshold if (suppcount(r)/|T|)>=minsupp.

Definition 10 A rule entry r exceeds the minconf threshold if (suppcount(r)/actoccr(r)) >=minconf.

Definition 11 Any itemset i that exceeds the threshold value minsupp is said to be a frequent itemset.

Definition 12 Any rule entry r that exceeds the threshold minsupp is said to be a frequent rule entry.

Definition 13 CAR is represented in the form: (Ai1, ai1)
^. . .^ (Aik, aik) □ c, where the left side (antecedent) of the rule is the set of items and the successor is the class.

A classifier is the representation form H : A□Y, where A is the set of itemsets and Y is the set of classes. The main task of associative classification is to construct a set of rules (a model) that is able to predict as accurately as possible the classes of previously unseen data, known as the test dataset. In other words, the goal is to find the classifier h € H that maximizes the probability that h(a) = y for each test object. The task of associative classification is different from discovering association rules. The most obvious difference between associative rule discovery and associative classification is that associative classification only considers the class attribute in the subsequent rules. However, the former allows multiple attribute values in subsequent rules. Table 1 shows the main important differences between associative classification and association rule discovery, where prevention of overfitting is crucial in associative classification but not in association rule discovery because associative classification involves using a subset of the discovered rule set to predict classes of new data. objects. Overfitting often occurs when discovered rules perform well on the training data set and poorly on the test data set. This can be due to several reasons, such as a small number of training data objects or noise.

The problem of constructing a classifier using an associative classifier can be divided into four main steps as follows.

- Step 1: Revealing all frequent rule items.
- Step 2: Creating all CARs that have confidence above the minconf threshold from frequent rule items extracted in step 1.
- Step 3: Selecting one subset of CARs to create a classifier from those generated in Step 2.
- Step 4: Measuring the quality of the derived classifier on test data item occurs when the discovered rules perform well on the training data set and poorly on the test data set as well as items. This can be due to several reasons, such as a small number of training data objects or noise.

Table 1 The main differences between AC and association rule discovery.

Association rule discovery	**Associative classification**
No class attribute involved (Unsupervised learning).	A class must be given (supervised learning)
The aim is to discover associations between items in a transactional database.	The aim is to construct a classifier that can forecast the classes of test data objects
There could be more than one attribute in the Consequent of a rule.	There is only attribute (class attribute) in the consequent of a rule.
Overfitting is usually not an issue	Overfitting is an important issue

V. GENERATION OF CLASSIFICATION RULES FOR CMAR

In this section, we develop a new associative classification method called CMAR, which performs classification based on multiple association rules. CMAR consists of two phases: rule generation and classification. In the first stage, rule generation, CMAR computes a complete set of rules of the form R: P □ c, where P is a pattern in the training data set and c is a class label, so that sup(R) and conf (R) to overcome the given limits of support and reliability, respectively. In addition, CMAR prunes some rules and selects only a subset of high-quality rules for classification. In the second phase, classification, for a given data object Obj, CMAR extracts a subset of rules corresponding to the object and predicts the class label of the object by analyzing this subset of rules. In this section, we develop methods for generating classification rules. To find rules for classification, CMAR first mines the training data set to find the complete set of rules passing certain support and confidence thresholds. This is a typical task for mining frequent patterns or association rules [1]. To make mining highly scalable and efficient, CMAR adopts a variant of the FP-growth method [5]. FP-growth is a common pattern mining algorithm that is faster than conventional Apriori-like methods, especially in situations where there are large datasets, low support threshold, and/or long patterns. The general idea of mining rules in CMAR is given in the following example.

Example 1. (Mining Class Association Rules) Given that the training dataset TH is shown in Table 1. Let the support threshold be 2 and the confidence threshold be 50%. CMAR mines the class association rules as follows.

Table 1 A Training Data Set

RowId	**A**	**B**	**C**	**D**	**Class Label**
1	a1	b1	c1	d1	A
2	a1	b2	c1	d2	B
3	a2	b3	c2	d3	A
4	a1	b2	c3	d3	C
5	a1	b2	c1	d3	C

.

First, CMAR scans the training dataset TH once and finds a set of attribute values that occur at least twice in T . The set is F={a1,b2,c3,d1} and is called the set of time items. All other attribute values that do not meet the support threshold cannot play any role in the class association rules and may therefore be pruned. Then CMAR sorts the attribute values in F in descending order, i.e. F-list =a1-b2-c3-d. Then, CMAR rescans the training dataset to build an FP-tree as shown in Figure 2 .

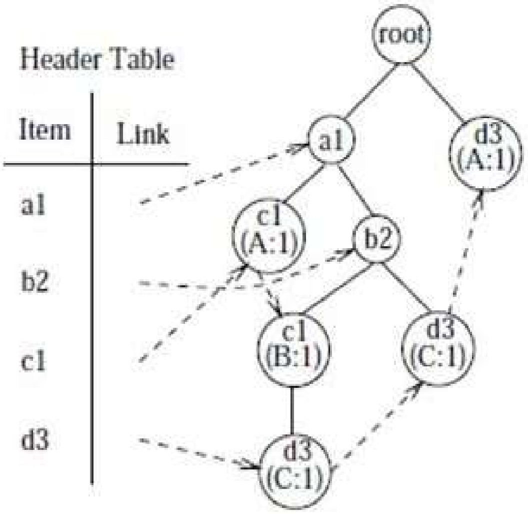

Figure 2. FPTree from training data set

FP-tree is a prefix tree w.r.t. F-list. For each tuple in the training data set, attribute values appearing in the F-list are extracted and sorted by the F-list. For example, for the first tuple, (a1,c1) is extracted and inserted into the tree as the leftmost branch in the tree. The class label is assigned to the last node in the path. The tuples in the training data set share prefixes. For example, the second tuple carries the values of the attributes (a1,b2,c1) in list F and shares a common prefix a1,b2 with the first tuple. So it also shares the subpath a1,b2 with the leftmost branch. All nodes with the same attribute value are connected as a queue started from the header table. Third, based on the F-list, the set of class association rules can be divided into 4 non-overlapping subsets: (1) those with d3 ; (2) those that have c1 but no d3; (3) those that have b2 but no d3 or c1; and (4) those having only a1. CMAR gets these subsets one by one.

Fourth, to find a subset of rules with d3, CMAR traverses nodes with attribute value d3 and looks "up" to collect a projected database d3 that contains three tuples: (a1,b2,c1,d3) : (a1, b2, d3): and d3. contains all tuples having d3. The problem of finding all frequent patterns with d3 in the entire training set can be reduced to mining frequent patterns in the projected d3 database. Recursively, in the projected database d3, a1 and b2 are frequent attribute values, i.e. they exceed the support threshold. We can mine the projected database recursively by constructing FP-trees and projected databases. It just so happens that in the d3projected database, a1 and b2 always occur together, so a1b2 is a frequent pattern. a1 and b2 are two subpatterns of a1b2 and have the same number of supports as a1b2. To avoid triviality, we only use the frequent pattern a1b2d3 . Based on the information about the class label distribution, we generate a rule a1b2d3 □ C with support 2 and confidence 100%. After finding rules with d3, all nodes of d3 are merged into their parent nodes. This is class label information registered in node d3 is registered in its parent node. The FP-tree is reduced as shown in the figure 3. Please note that this tree reduction operation is performed at the same scan of the projected d3 database collection.

The remaining rule subsets can be mined similarly. There are two main differences in rule mining in CMAR and the standard FP-growth algorithm. On the one hand, CMAR finds frequent patterns and generates rules in one step. Typically, association rules must be mined in two steps. This is also the case with traditional associative classification methods. First, all frequent patterns (ie patterns passing through the support threshold) are found. Then, based on the extracted frequent patterns, all association rules meeting the confidence threshold are generated. The difference of CMAR from other associative classification methods is that, for each pattern, CMAR maintains a distribution of different class labels among the data objects matching the pattern. This is done without any overhead in the (conditional) database count procedure. Thus, once a frequent pattern is found (i.e., the pattern traversal support threshold), rules about the pattern can be generated immediately. Therefore, CMAR has no separate rule generation step. On the other hand, CMAR uses the class label distribution for trimming. For any frequent pattern P, let c be the most dominant class in the set of data objects corresponding to /. If the number of objects with a class label and corresponding P is less than the support threshold, there is no need to search for any super pattern (superset) P' of P, because no rule of the form P□C can meet the support threshold either.

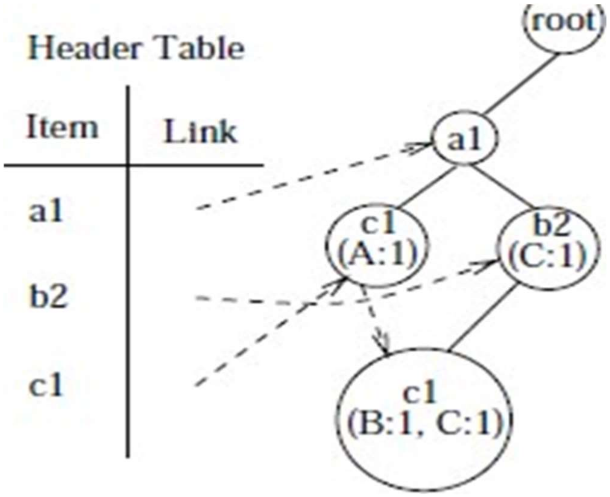

Figure 3. FP Tree merging after nodes of d3.

A) Storing rules in the CR tree.

Once a rule is generated, it is stored in a CR-tree, which is a tree structure of prefixes. We demonstrate the general idea of a CR-tree with the following example

. **Example 2 (*CR-tree*)** after mining training data set, four rules are found as shown in Table 2.

Table 2 Rules found in training data set.

Rule Id	***Rule***	***Support***	***Confidence***
1	***abc□ A***	***80***	***80%***
2	***abcd □A***	***63***	***90%***
3	***abe□ B***	***36***	***60%***
4	***bcd□ D***	***210***	***70%***

A *CR-tree* is built for the set of rules, as shown in Figure 4, while the construction process is explained as follows

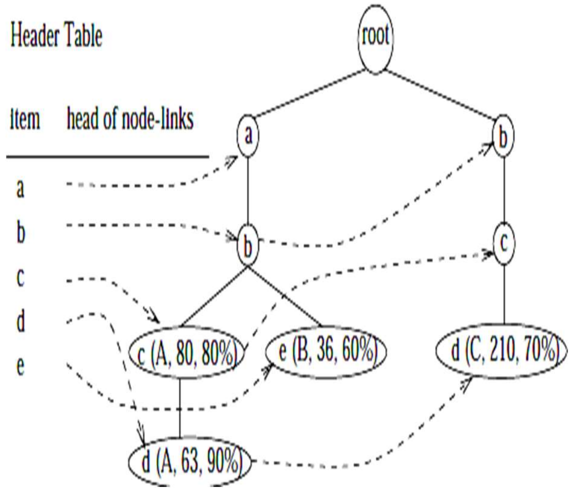

Figure 4 CR Tree for Rules in Example2

A CR-tree has a root node. All attribute values that appear on the left side of the rules are ordered by their frequency, i.e. the most frequently occurring attribute value comes first. The first rule, abc□ A, is inserted into the tree as a path from the root node. The class label as well as the support and confidence of the rule, denoted as (A,80,80%), are registered at the last node in the path, i.e., the node for this rule.

The second rule, abcd□A , shares the prefix abc with the first rule. It is therefore inserted into the tree by expanding the new node d to the path formed by the first rule. Again, the class label, support and trust of the rule are registered in the last node, i.e. the third and fourth rules can be pushed or inserted similarly. All nodes with the same attribute value are linked by node-link into a queue. The header of each queue is stored in the header table. 13 cells are needed for the left side of the rules to store the original set of rules. Only 9 nodes are needed when using a CR tree. As can be seen from the above example, the CR-tree structure has some following advantages. A CR-tree is a compact design. It examines potential sharing between rules, so it can save a lot of storage space for rules. Our experimental results show that in many cases about 50-60% of space can be saved by using CR-tree. The CR-tree itself is an index for the rules. For example, if we want to get all the rules with attribute value b and d in the ruleset in example 2, we only need to traverse the nodes d that start in the header table and keep looking for b up. . Once the CR-tree is established, rule retrieval becomes efficient. This makes it much easier to trim rules and use rules for classification.

B) Pruning rules.

The number of rules generated by class association rule mining can be huge. To make the classification effective as well as efficient, we need to trim the rules to remove redundant and noisy information. According to the ability of the rules for classification, a global order of rules is compiled. Given two rules R1 and R2, R1 is said to have higher rank than R2, denoted R1 > R2 , if and only if (1) conf(R1) > Conf(R2) (2) conf(R1) = conf(R2) but Sup(R1) > Sup(R2) or (3) conf(R1)=conf(R2), Sup(R1)=Sup(R2), but R1 has fewer attribute values on the left than R2. Moreover, the rule R1: P□C is called a general rule w.r.t. rule R2: P' □ C' , if and only if

P/ is a subset of P'. CMAR uses the following methods for pruning rules. First, use a general rule and a high confidence rule to cut out the more specific and the less confident. Given two rules R1 and R2, where I is the general rule w.r.t. R2. CMAR prunes R2 if R1 also has a higher rating than R2. . This is because we only need to consider general rules with high confidence R1, and therefore more specific rules with low confidence should be truncated. This pruning is done when the rule is inserted into the CR-tree. When a rule is inserted into the tree, it starts traversing the tree to check if the rule can be pruned or if it can prune other rules that are already inserted. Our experimental results show that this pruning is effective. Second, selecting only positively correlated rules. For each rule R:P□C, we test whether P is positively correlated with c using χ2 testing. Only rules that are positively correlated, i.e. those for which the χ2 value exceeds the significance level threshold, are used for later classification. All other rules are trimmed. The reason for this trimming is that we use rules reflecting strong implications for classification. By removing those rules that are not positively correlated, we reduce the noise.

After selecting a set of classification rules, CMAR is ready to classify new objects. Given a new data object, CMAR collects a subset of rules corresponding to the new object from the rule set for classification. In this section, we discuss how to determine a class label based on a subset of rules. Trivially, if all rules matching a new object have the same class label, CMAR simply assigns that label to the new object. If the rules are not consistent in the class labels, CMAR groups the rules according to the class labels. All rules in a group share the same class label and each group has its own designation. CMAR compares group effects and returns with the strongest group. In order to compare the strength of groups, we need to measure the "combined effect" of each group. Intuitively, if the rules in a group are highly and potentially correlated and have good support, the group should have a strong effect. There are many possible ways to measure the combined effect of a group of rules. For example, the strongest rule can be used as a proxy. This means that the rule with the highest χ2 value is selected. However, simply choosing the rule with the highest χ2 value can be advantageous for minority classes, as the following example shows.

VI. GENERATION OF RULES FOR CLASSIFICATION BY USE OF CPAR

CPAR (Classification based on Predictive Association Rules), which combines the leverages of both associative classification and traditional rule-based classification. Rather than generating a large number of candidate rules as in case of associative classification, CPAR snatches a greedy algorithm to generate rules directly from the training data. In addition, CPAR generates and tests more rules than traditional rule-based classifiers to avoid missing important rules. To avoid overfitting, CPAR uses the expected accuracy to evaluate each rule and uses the best k rules in prediction. CPAR stands in the middle between exhaustive and greedy algorithms, combining the advantages of both. CPAR creates rules by adding literals one at a time, similar to PRM. However, instead of ignoring all but the best literals, CPAR keeps all literals close to the best during the rule generation process. In this way, CPAR can select more than one literal at a time and create several rules at the same time. The following is a detailed description of the CPAR rule generation algorithm. Suppose that at some step in the rule generation process, after finding the best literal p, another literal q is found that has a similar gain to p (eg, differs by at most 1%). In addition to continuing to create a rule by attaching p to r, q also attaches to the current rule r to create a new rule r0 that is en-queued. Each time a new rule is to be built, the queue is first checked. If it is not empty, a rule is extracted from it, which is taken as the current rule. This constitutes a depth-first search when generating rules.

Example. Figure 5 depicts an example of how CPAR generates rules. After selecting the first literal (A1 = 2), two literals (A2 = 1) and (A3 = 1) are found to have similar gain, which is higher than the other literals. First, the literal (A2 = 1) is selected and a rule is generated in that direction. Then the rule (A1 = 2; A3 = 1) is considered the current rule. Again, two literals with similar gain (A4 = 2) and (A2 = 1) are selected and a rule is generated along each of the two directions. This generates three rules:

(A1 = 2; A2 = 1; A4 = 1).

(A1 = 2; A3 = 1; A4 = 2; A2 = 3).

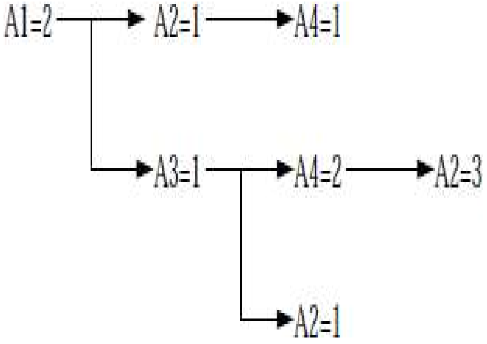

Figure 5 Some Rules Generated by CPAR. CPAR's rule generation takes O(n_k |R})
time.

VII. EXPERIMENTAL RESULTS

We conducted an extensive performance study to evaluate the accuracy and efficiency of CPAR, CMAR and compare it with FOIL, PRM.

We have validated our approach using a large set of experiments addressing the following problems:

1. Implementation of classification and association rules in terms of execution time, memory usage.

2. Compliance with the Classification and Association Rules in terms of classes and accuracy.

3. Implementation of classification and association rules in terms of classes and number of generated rules.

4. Scalability of the approach.

All experiments are performed on a standard architectural computer with 8GB of main memory and a Microsoft Windows10 operating system. The following diagram shows the time complexity comparison between different FOIL, PRM algorithms, CPAR, CMAR, CRAM using a line graph.

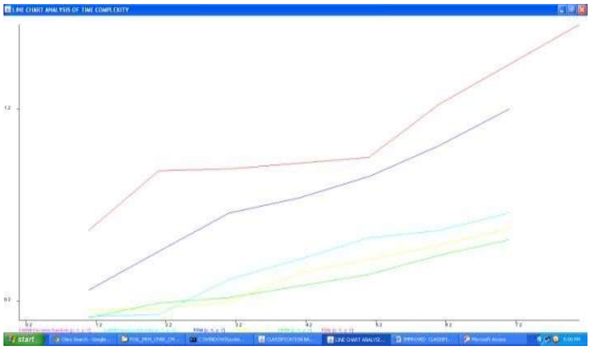

Fig 7.1 Comparison of Time Complexity of algorithms.

The Following Diagram shows the comparison of space complexity between different algorithms FOIL, PRM
,CPAR, CMAR, CRAM using Line Chart.

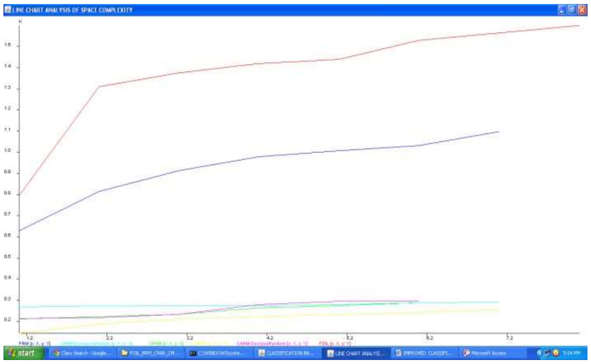

Fig 7.3 Comparison of Accuracy of algorithms.

The Following Diagram shows the comparison of no ofRules generated between different algorithms FOIL, PRM ,CPAR, CMAR, CRAM using Line Chart.

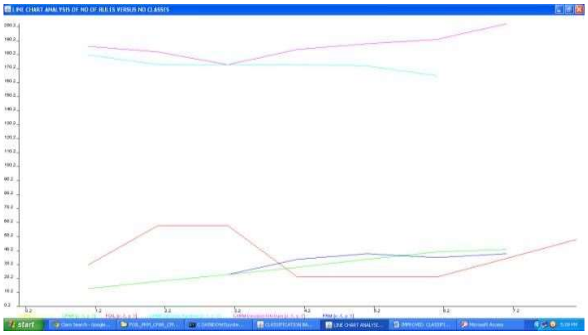

Fig 7.4 Comparison of no. of Rules of algorithms.

The Following Diagram shows the comparison of timecomplexity between different algorithms FOIL, PRM

,CPAR, CMAR, CRAM using Bar Chart

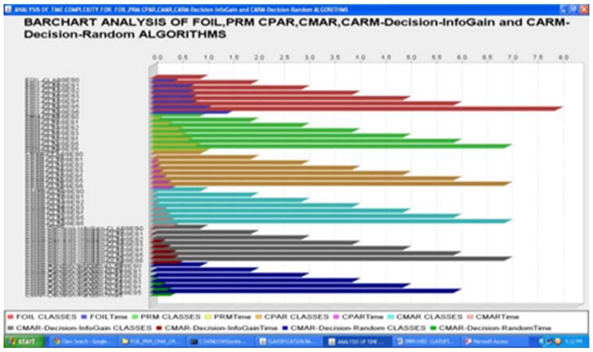

Fig 7.5 Comparison of Time Complexity of algorithms.

The Following Diagram shows the comparison of space complexity between different algorithms FOIL, PRM
,CPAR, CMAR, CRAM using Bar Chart.

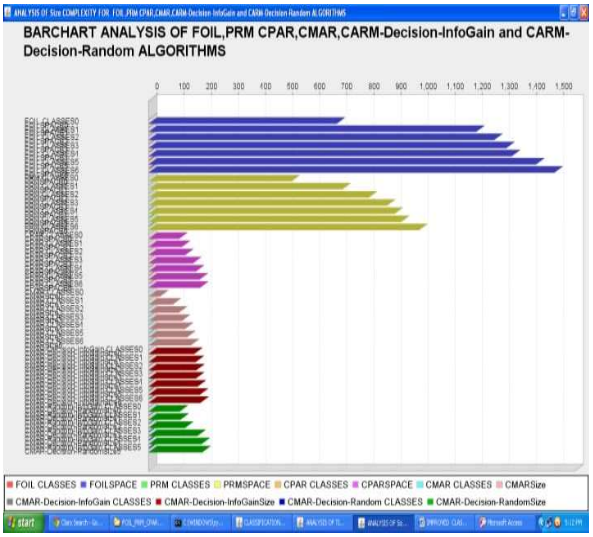

Fig 7.6 Space Complexity comparison of algorithms

The Following Diagram shows the comparison of Accuracy, Complexity of different algorithms FOIL, PRM, CPAR, CMAR, CRAM using Bar Chart.

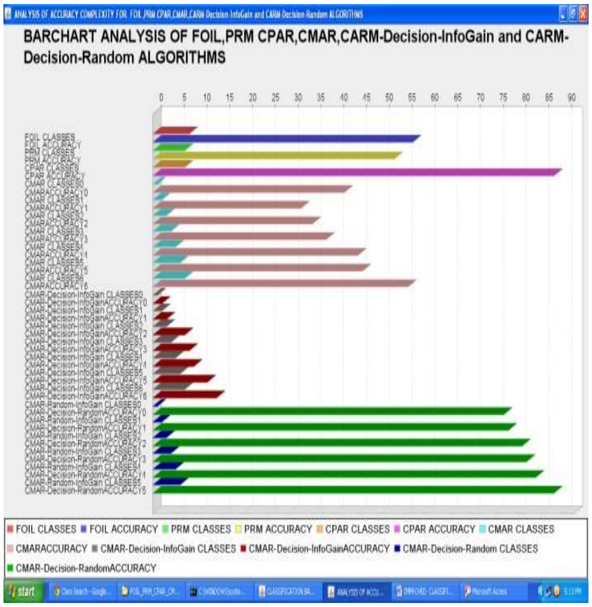

Fig 7.7 Comparison of Accuracy of algorithms.

The Following Diagram shows the comparison of No ofRules among different algorithms FOIL, PRM ,CPAR, CMAR, CRAM using Bar Chart.

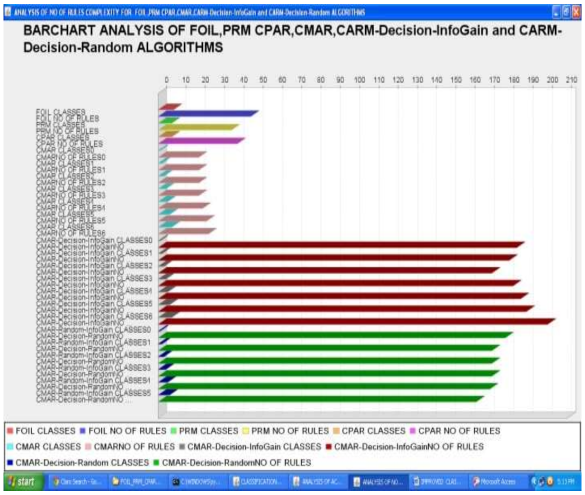

Fig 7.8 Comparison of No of Rules of algorithms.

VIII. CONCLUSION

In this paper, it was investigated two main issues of associative classification: (1) efficiency in handling a large number of mined association rules and (2) efficiency in predicting new class labels with high classification accuracy. We proposed two new associative classification methods, CMAR, i.e. classification based on multiple association rules, and CPAR (classification based on predictive association rule).

The CMAR method has several distinctive features: (1) its classification is performed based on a weighted χ2 analysis forced on multiple association rules, resulting in better overall classification accuracy; (2) it effectively prunes rules based on confidence, correlation, and database coverage.

and (3) its effectiveness is achieved by extending the efficient method of frequent pattern mining, FP-growth, construction of FP-tree associated with class distribution, and application of CR-tree structure for efficient storage and retrieval of mined association rules.

CPAR is developed to integrate classification and association rule mining. Based on our performance study, CPAR achieves high accuracy and efficiency, which can be attributed to the following salient features: (1) it uses a greedy rule generation approach that is much more efficient than generating all candidate rules, (2) it uses a dynamic programming approach to avoid repeated computations when generating rules, (3) selects multiple literals and generates multiple rules simultaneously, and (4) uses expected accuracy to evaluate rules and uses the best to rules in prediction. CPAR represents a new approach to efficient and high-quality classification. The experiments show that both CMAR and CPAR show better performance than FOIL and PRM.

IX. REFERENCE

[1]. Devasri Rai, A.S.Thoke and Keshri Verma, "Enhancement of Associative Rule based FOIL and PRM Algorithms", Proc.. 2012.

[2]. Quinlan, J.R.(1993). C4.5: Programs for Machine Learning . San Mateo, Morgan Kaufmann , San Francisco

[3]. F.Clark and T.Niblett. The cn2 induction algorithm. Machine Learning, 2:261-283,1989

[4]. Memory based reasoning(Data Mining Techniques), by M.J.A. Berry and G.S Linoff,2004

[5]. R.Andrews, J.Diederich, and A. Ticke,"A survey and critique of Techniques for extracting rules from trained artificial neural networks, "Knowledge Based Systems , volume ,pp. 373-389,1995.

[6]. P.Langely , W.Iba , and K.Thomson . "An analysis of Bayesian Classifiers," in National Conference on Artificial Intelligence (1992).pp.223-228

[7]. Liu B., Hsu W. And Ma Y. Integrating Classification and Association rule Mining . In Proceedings of the 4th international Conference on Knowledge Discovery and Data Mining (KDD-2001, pages 80-86, new York ,USA , August 1998. The AAAI press.

[8]. J.R. Quinlan and R.M. Cameron-Jones. FOIL: A midterm report. In Proc. 1993 European Conf. Machine Learning . pp. 3{20,Vienna Austria, 1993.

[9]. X.Yin and J.Han, "CPAR: Classification Based on Predictive Association Rules", Proceedings of Siam International conference on Data Mining ,2003, pp.331-335

[10]. Prafulla Gupta & Durga Toshniwal, "Performance Comparision of Rule Based Classification Algorithms", International Journal of Computer Science & Informatics, Volume-1,Issue-II,2011,pp37-42.

[11]. W. Li ,J.Han and J.Pie CMAR: Accurate and efficient Classification based on multiple class-association rules. In ICDM01, 2011,pp.369{376.sanjose, CA ,Nov.2001.

[12]. Thabtah , F., Cowling, P. And Peng , Y.H.(2004).MMAC: A New Multi-Class , Multi-Label Associative Classification Approach. Fourth IEEE International Conference on Data Mining (ICDM04).

[13]. "Classification Based on Predictive Association rule, " Available online:http://www.csc.liv.ac.uk/~frans/KDD/Software/FOIL_PRM_CPAR/ cpar.html.

[14].Uci:Blake , c.i.,&Merz,C,J(1998) UCI repository of machine learning data bases from www.ics.uci.edu/~ mlearn/MLrepository.html

[15].Coenen, F.(2003), The LUCS-KDD Discretized /normalised ARM and CARM data library

http://www.csc.liv.ac.uk/~frans/KDD/Software/LUCS_KDD_DN/, Department of Computer Science , The University of Liverpool, UK.
[16]. Zuoliang Chen, Guoqing Chen, "Building An Associative Classifier Based On Fuzzy Association Rules", International Journal of Computational Intelligence Systems, Vol.1, No. 3 (August, 2008), 262 – 273. [17]. Xin Lu, Barbara Di Eugenio and Stellan Ohlsson, " Learning Tutorial Rules Using Classification Based On Associations", Xin Lu, Computer Science (M/C 152), University of Illinois at Chicago, 851 S Morgan St.,
Chicago IL, 60607, USA. Email: xlu4@uic.edu
[18]. "Alaa Al Deen" Mustafa Nofal and Sulieman Bani-Ahmad, " Classification based on association-rule mining techniques: a general surveyand empirical comparative evaluation", Ubiquitous Computing and Communication Journal, Volume 5 Number 3, www.ubicc.org.
[19]. A. Zemirline, L. Lecornu, B. Solaiman, and A. Ech-cherif, "An Efficient Association Rule Mining Algorithm for Classification", L. Rutkowski et al. (Eds.): ICAISC 2008, LNAI 5097, pp. 717–728, 2008. C Springer-Verlag Berlin Heidelberg 2008.
[20]. Bing Liu Wynne Hsu Yiming Ma, "Integrating Classification and Association Rule Mining ",Appeared in *KDD-98*, New York, Aug 27-31, 1998.