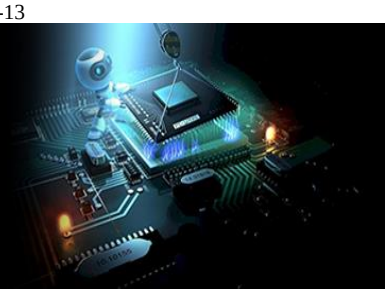
International Journal of Engineering in Computer Science



E-ISSN: 2663-3590
P-ISSN: 2663-3582
IJECS 2024; 6(1): 01-13
Received: 01-11-2023
Accepted: 06-12-2023

Aswini Kumar Mohanty
Capital Engineering College, Khorda, Odisha, India

Corresponding Author:
Aswini Kumar Mohanty
Capital Engineering College, Khorda, Odisha, India

An effective and empirical technique for classification using association rule

Aswini Kumar Mohanty

DOI: https://doi.org/10.33545/26633582.2024.v6.i1a.101

Abstract
Organizational policy so called association rule mining and classification are two important data mining terminology and technology in the knowledge discovery process. The combination of these two methods is an important research topic and has many applications in data mining. The combination of these two methods creates a new method called group mining policy or group classification system. The combination of these two methods provides better classification accuracy when classifying data. The research field of content-based data collection requires high efficiency and productivity. Join the mining rule to find the engagement pattern from the data in these applications; we will classify the targets based on the engagement pattern. Our paper focuses on the combination of classification and association rule mining to achieve classified data. In this paper, we propose the use of two new algorithms, CPAR (classification based on predictive association rules) and CMAR (classification based on multi-category association rules), which provide integration and distribution benefits when necessary. CPAR uses anger detection techniques to generate rules directly from training data rather than creating common constructs such as classification entities. Additionally, CPAR develops and tests regulations rather than policy-based procedures to avoid significant regulation. To avoid over-fitting, CPAR evaluates each rule using expected accuracy and uses the top-k rules for prediction. CMAR uses the CR tree model to efficiently store and retain rules in the mined organization and truncate rules based on trust, necessity, and evidence. Distributions were based on a weighted χ analysis using multiple association rules. Extensive experiments show that CMAR is consistent, efficient, and has better mean variance than FOIL (first-person inductive learner) and PRM (predictive rule mining) for classifying different products. The proposed algorithm is better in terms of required memory, time consumption and eliminating intermediate data structure when used.

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)

1. Introduction
Classification rule mining and association rule mining are two important data science technologies. Distributed rule mining is used to discover small patterns in data to create a distinct truth. Organizational rule mining called as association rule mining is used to discover all relationships in big data. Association rule mocks find all rules in the file that meets certain minimum support and minimum confidence conditions. While the exploration target for shared mining rights is not determined in advance, only a single target is determined for the distribution of mining rights. These two methods can be combined to form a foundation called the statistical method. Association is done to obtain a special set of association rules whose right-hand side is limited to the properties of the categorical class. Subsets of these rules are called organizational rules. Classification using association rules is limited to situations where the condition can only occur in a separate group. This is because the organization's rule mining only works on categorical attributes. The leading Y of every association rule $X \rightarrow Y$ is the separation of objects. However, general union rules cannot be directly applied. We must narrow down their definitions. Any element not found in the rule body may appear in the rule header. When we want to use rules for classification, we are interested in rules that can ensure category participation. Therefore, we limit the Y header of the $X \rightarrow Y$ category association rule to a single element. The attribute of this attribute-value pair must be an attribute category. Of course, classroom rules are a predictable task. We can create a classification using the ability to distinguish between group rules.

Designing accurate and effective classifiers for large data sets is one of the most important tasks of data mining and machine learning. Given the events in the class list as a training method, the operator needs to build a model (called a classifier) whose records Unknown. Previous studies have developed heuristic/greedy search strategies for class formation, such as decision trees [10], rule-based learning [2, 4, 13, 18], naive Bayesian distribution [4, 9, 17], and statistical methods [8]. These methods teach representative rules (e.g., decision trees or rule sets) from training data to make good predictions. Recent research suggests extracting effective collaborative processes from training data that meet specific user frequencies and trust levels. Create an efficient and effective classification by carefully choosing rules such as CBA [9], CAEP [3], and ADT [11]. This method selects the best rules from all received rules. Because association rules examine the reliability of associations across many different dimensions, they can overcome some of the limitations of decision tree induction, which examines a single variable at a time. Performance studies [6, 9, 3, 11] show that clustering-based classification can often have better accuracy. In recent years, a new method called association classification [7, 6] has been proposed, which combines association policy mining [1] and classification. It uses association rule mining algorithms such as Apriori [1] or FP-growth [5] to generate all association rules.

It then chooses a small set of good rules and uses that rule to make predictions. Experiments in [7, 6, 18, 20] show that this method has a higher accuracy than traditional classifications such as C4.5 [8, 14]. In this article, we present two new processes called CPAR (Functional Collaboration Rules Based Classification) and CMAR (Multiple Collaboration Rules Based Classification). CPAR uses the simple concept of FOIL [9] in policy making and integrates aspects of integration into policy forecasting. Compared with statistical analysis, CPAR has the following advantages: (1) CPAR produces a small set of best methods from the dataset; (2) To avoid the creation of duplicate rules, CPAR targets "currently created" rules. All rules are created; (3), when predicting the class label of a sample, it uses the best CPAR of the rules that the sample satisfies. Moreover, CPAR also uses the following features to improve its accuracy and efficiency: (1) CPAR uses dynamic programming to avoid double calculation when creating rules; (2) When creating rules, instead of choosing the best one, choose the closest one. Immutable values to avoid missing important rules. Compared to clustering, CPAR creates smaller policies with better quality and lower cost. Therefore, CPAR takes more time in accurate generation and forecasting but is on par with participation distribution.

CMAR selects a small number of reliable and highly correlated systems and verifies the compatibility of these codes. To avoid bias, we developed a new method called weight $\chi 2$, which provides a good measure of the strength of promotion and distribution rules. Performance studies show that CMAR is generally more predictive than CBA [9] and C4.5 [10]. Second, to improve accuracy and efficiency, CMAR uses new CR tree data to store the classification code and make the most of it. The CR-tree is a tree structure that was first used to explore code sharing and thus realize the great promise. The CR tree itself is also a standard code model and is useful for storing code. Third, to increase the purpose of the goal to complete the process FP grows faster than Apriori-like methods previously used in association classification (e.g.,) [9, 3, 11], especially when there are many rules, data are large training papers, and long examples.

2. Related Work

Data analysis algorithms (or today's most popular data mining algorithms) can be divided into three main groups according to the nature of the data they retrieve [1]: clustering (also called segmentation or unsupervised learning), predictive modeling (also known as classification) or supervised learning) and frequent removal of models. Clustering is a broad class of data mining algorithms. These algorithms use a search process whose goal is to identify some of the best paths to all similar examples in the data. One of the oldest and stout clustering algorithms is k-means [2]. Two disadvantages of this algorithm are the initialization problem and the fact that the clusters must be linearly separable. To solve the initial problem, an additional decision algorithm, global k-means [3], was proposed that uses k-means as the local search algorithm. The k-means kernel algorithm [4] eliminates the linear separation limitation and uses the non-linear Ø transform to map the points of the access point to multiple locations and uses k-means for the feature space. K-means global kernel [5] is an algorithm that uses the kernel function to map data points from the access point to the multipoint location, optimizing error in certain areas by searching for near-perfect.

Due to the decision it makes, it is independent of the initial problem and can define unseparated groups in the input space. Therefore, the global kernel k-means algorithm combines the advantages of global k-means and kernel k-means. Another data integration is hierarchical clustering based on the Hungarian method [6], and the computational complexity of the proposed algorithm is O(n2). The main classification techniques are decision trees, unbiased Bayes and statistics [2]. They use heuristic search and greedy search techniques to come up with a set of rules to find distributors. C4.5 and Classification and Regression Trees (CART) are the most famous decision tree algorithms. The final category of data mining algorithms is active pattern extraction. For big data [7], describes an Apriori algorithm that generates all key rules of objects in the data. The algorithm passes over the database many times. The previous set includes items that expand over time. In each case, the support of candidate features provided by tuples in the data and features in the set region is evaluated. Initially, the bounding set has only one element, the empty space.

At the end of the test, the support of the candidate's product is compared with minsupport. Also decide whether the product will be added to the limit of the next contest. The algorithm ends when the boundary set is empty. When all objects that meet Mintsis support are found, the relevant rules will be created from these objects. Liu Bing *et al*. [8] proposed a correlation-based classification algorithm (CBA) to discover cluster association rules (CAR). It consists of two parts: a rule generation (called CBA-RG) and a classification rule (called CBA-CB) based on the Apriori algorithm to find association rules. While the Apriori algorithm uses a set of objects, CBA-RG uses a set having a condset (object) and a group. The class rule used to create the class in [8, 9] is more accurate than the C4.5 algorithm [2, 3, 16]. However, correlation-based classification (CBA) algorithms require statistical parameters to generate the classification. Evaluation is based on the support and reliability of each rule.

This makes CBA less of a classification based on guessing association rules. Neural network is a network that simulates human visual perception, based on the research of neural networks, created by simplifying, combining and optimizing the properties of biological systems and neural networks [9]. It uses the concept of nonlinear mapping, parallel processing methods and the structure of the neural network to express the input and output of information. At first, it is not good to apply neural networks in data search because the neural network model is complex, the training time is long, and the results are not easy to understand. However, it has the advantages of high data noise and low error, and the continuous improvement and optimization of many network training algorithms, especially the continuous improvement and optimization of various network pruning algorithms and rule extraction algorithms. In data search, neural networks are common and popular among users. Xianjun Ni [10] describes the data mining process based on neural networks. The process includes three main steps: data preparation, policy extraction, and policy evaluation. Classification is currently considered one of the most important tasks in data mining [14, 20].

Sharing real-world examples is something everyone has experienced at some point in their lives. Just as people can be classified according to their race, products on the market can also be classified according to the consumer's decision. Classification often involves examining the characteristics of new products and attempting to assign them to a set of predefined categories [38]. Considering the information recorded in the file, each file has a set of characteristics; one of the attributes is group.

The purpose of classification is to create patterns from individual objects to identify previously unseen objects as accurately as possible. There are many methods to extract information from data, such as divide and conquer [13], isolate and manage [15], coverage and statistics [20, 6]. The divide-and-conquer method first selects an attribute as its root and then creates a branch for each possible level of that attribute. This divides the training samples into subsets, one for each fit value. The same process will be repeated until all events falling within a branch have the same distribution or until no more events can be split.

On the other hand, it's a way of dividing and conquering before greedily establishing the rules (one by one). Once a policy is found, all events to which that policy applies will be deleted.

The best policy is to do the same process until you find a major error. Statistical methods such as Naive Bayes [19] use statistical parameters (i.e. probability) to classify test items. Finally, the coverage method [6] selects each class in turn and finds a way to cover most of the training material in that class to generate the code with the highest accuracy. Many algorithms such as decision trees [12, 10], PART, RIPPER and Prism [6] are derived from this model. In the research, some studies were carried out in several layers [14, 7, 6, 19]. So far, most of the research on multiple documentations has focused on text classification [20]. In this study, only the classification of rule-forming algorithms into a group is discussed.

3. Design and implementation of the system

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

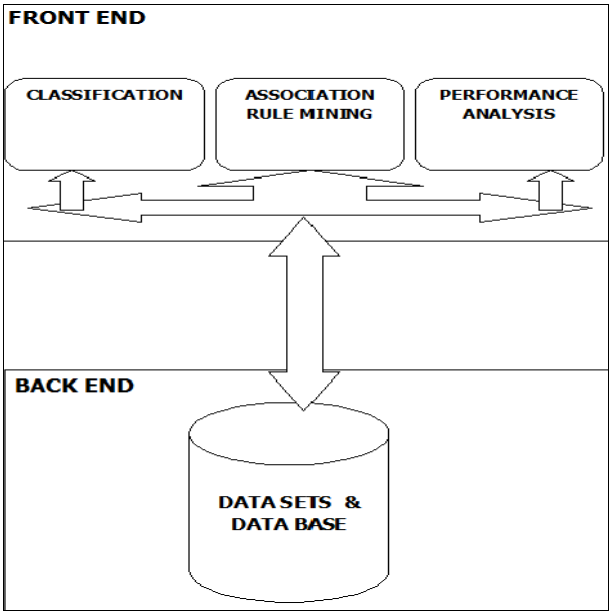


Fig 1: System Architecture

A) Data source / database module

This module stores information in the form of datasets.

Here we have a dataset with many attribute values in the form of data exchange and a dataset containing the dataset schema. Useful for categorizing such data.

B) Classifications module

This module reads the data from the data set and performs the classification and sorting process.

C) Association rules creation module

This module uses the group for my association rules and creates active objects to create association rules.

D) Performance Analysis part

This model calculates the number of groups based on various algorithms such as CPAR, CMAR, FOIL and PRM. Then compare your results and determine the best algorithm.

4. Associative classification

Relational classification is a special type of association rule discovery in which only category features on the right-hand (next-door) side of the rule are taken into account; For example, in a rule like X → Y, Y must be the attribute category. One of the main advantages of using association-based classification over classical classification methods is that the results of the association classification algorithm are represented by simple if-then rules, making it easier for end users to understand and interpret. Also, unlike decision tree algorithms, the rules in integration can be changed or changed without affecting the general rules, while the same task must replace the entire tree in the decision tree. Let us analyze the classification problem in an organization where the data set T has m variables A1, A2, Includes group names., Am and C. The number of bundles in T is denoted by T. Attributes can be categorical (that is, they take values from the limit of possible values) or continuous (they are real numbers or numbers). For categorical behavior, each possible and desired value corresponds to a set of positive numbers. Use discretization techniques for continuous features.

Definition 1: An array or training object in T can be defined as a combination of the attribute name Ai and the value aij plus the category expression cj.

Definition 2: An object can be defined as the attribute name Ai and the value ai represented as <(Ai, ai)>.
Definition 3: An itemset can be defined as a set of discrete attribute values present in the training object, denoted as <(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 of the correct element (actoccr) r in T is the number of rows in T that match the set of elements r.

Definition 6: The support number of a rule r is the line that matches element r in T and is in class c in r.

Definition 7: The number of times and itemset i (occitm) appears in T is the number of rows matching i in T.

Definition 8: Item set minsupp above threshold i if (occitm(i)/T)>=minsupp.

Definition 9: Rule entry r above minsupp threshold if (suppcount(r)/T)>=minsupp.

Definition 10: Rule entry r above the minconf threshold if (suppcount(r)/actoccr(r)) >=minconf.

Definition 11: A minsupp itemset i that exceed the threshold is called an active itemset.
Definition 12: An entry rule r that exceeds the minsupp threshold is called an active entry rule.

Definition 13: CAR is expressed as: (Ai1, ai1)
^. . .^ (Aik, aik)→c, where the left side of the rule (main) is the process and its successor is the class.

A classifier is a representation of H: A → Y; where A is the itemset and Y is the category. The main task of relational analysis is to create a set of rules (models) that can predict the category of previously unseen data (so-called test data) in order to correct them as much as possible. In other words, the goal is to find the classifier h € H that maximizes the probability h(a) = y for all test items. The role of classification of organizations is different from the pursuit of organizational rights. The main difference between organizational law discovery and organizational classification is that integration only determines category properties in subsequent laws. But the former permits many necessary features to be utilized in subsequent codes. Table 1 shows the premier important discerns between classification based on association and association rule discovery, where prohibition of over-fitting is more important as well as necessary in associative classification that not required in case of association rule discovery because associative classification acts upon using a subset of the discovered rule set to predict classes of new data objects. Over-fitting more often occurs when discovered rules perform well on the training data set and badly on the test data set. This can be due to many causes, such as a small number of training data objects or noise.
The problem of generating a classifier by the use of correlation called associative classifiers which can be segregated into four main steps below.

- **Step 1:** Display all the frequently used items.
- **Step 2:** confirm all CARs those have a higher confidence level than the default minconf in the active set extricated in step 1.
- **Step 3:** Select a subset of CARs to construct the classifier built in step 2.
- **Step 4:** Evaluating the quality of the attained classifier on test data item occurs when the discovered rules perform well on the training data set and badly 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). The aim is to discover associations between items in a transactional database. There could be more than one attribute in Consequent of a rule. Overfitting is usually not an issue	A class must be given (supervised learning) The aim is to construct a classifier that can forecast the classes of test data objects There is only attribute (class attribute) in the consequent of a rule. Overfitting is an important issue

5. Generation of classification rules for CMAR

In this section, we develop a new classification system called CMAR, which classifies according to multiple classification rules. CMAR has two phases: custom design and deployment. In the first step, rule generation, CMAR is calculated for all sets R in the form: P->c; where P is the model in the training data and c is the category labels; hence sup(R) and conf(R) overcome certain support and dependency constraints respectively. In addition, CMAR deletes some rules by selecting only a set of good rules for distribution. In the second stage, classification stage, for object data Obj, CMAR extracts the rule subset corresponding to the object and predicts the name of the object by analyzing the rule subset. In this section, we develop ways to create separate rules. To find classification rules, CMAR first searches the training data to find the successful set that passes certain levels of support and confidence. This is an important task of the frequent pattern or relation search for active standards or organizational association rules [1]. To make mining more efficient and profitable, CMAR takes a different approach to FP development [5]. FP-growth is a pattern mining algorithm that is faster than Apriori-like methods, especially when there are large datasets, low support, and/or long models. The example below gives a full idea of the mining rules in CMAR.
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

Row Id	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.

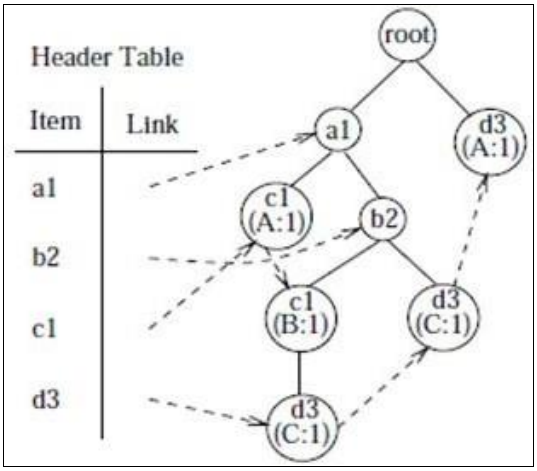


Fig 2: FPTree from training data set

FP-tree is a prefix tree with respect to 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 that contains all tuples having d3. The riddle of chancing all frequent patterns of attributes with d3 in the entire training data set can be dropped to mining frequent patterns of attributes in the estimated d3 database. Repetitively, in the decided database d3, a1 and b2 are frequent trait values, i.e. they surpass the support threshold. We can booby-trap the prognosticated database recursively by constructing FP- trees and prognosticated databases. It just so happens that in the d3 prognosticated database, a1 and b2 always do together, so a1b2 is a frequent pattern. a1 and b2 are two sub patterns of a1b2 and have the same number of supports as a1b2. To avoid negligibility, we only use the frequent pattern a1b2d3. Grounded on the information about the class marker distribution, we induce a rule a1b2d3->C with support 2 and confidence 100. After chancing rules with d3, all bumps of d3 are intermingled into their parent bumps. This is class marker information registered in knot d3 is registered in its parent knot. The FP- tree is reduced as shown in the figure 3. Please note that this tree reduction operation is performed at the same checkup of the projected d3 database collection.
The remaining rule subsets can be booby-trapped also. 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. Generally, association rules must be booby-trapped in two ways. This is also the case with traditional associative bracket styles. First, all frequent patterns (i.e. patterns passing through the support threshold) are set up. Also, grounded on the uprooted frequent patterns, all association rules meeting the confidence threshold are generated. The difference of CMAR from other associative bracket styles is that, for each pattern, CMAR maintains a distribution of different class markers among the data objects matching the pattern. This is done without any outflow in the (tentative) database count procedure. So, once a frequent pattern is set up (i.e., the pattern traversal support threshold), rules about the pattern can be generated incontinently. Therefore, CMAR has no segregate rule generation step. On the other hand, CMAR uses the class marker distribution for trouncing. For any frequent pattern P, let c be the most dominant class in the set of data objects corresponding to/. still, there's 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, If the number of objects with a class marker and corresponding P is lower than the support threshold.

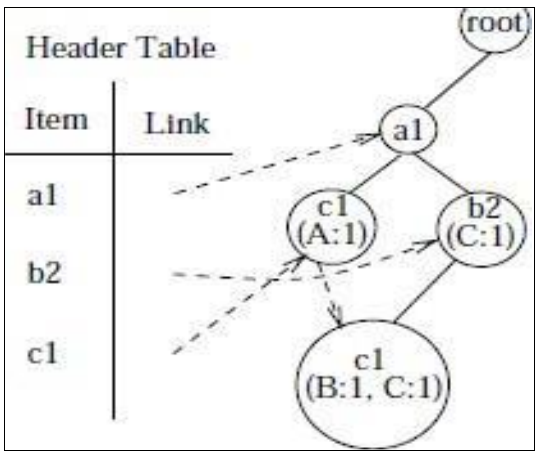


Fig 3: FP Tree merging after nodes of d3

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): four rules are found after mining training data set 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.

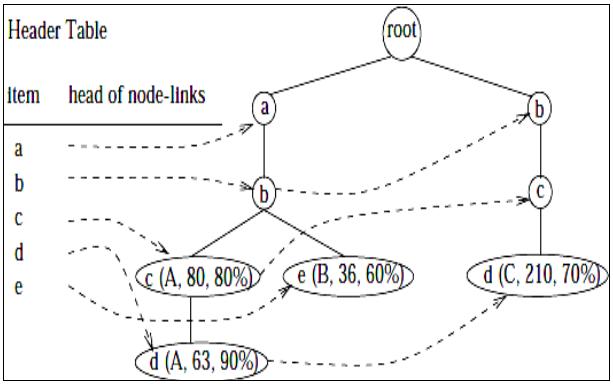


Fig 4: CR Tree for Rules in Example2

A CR tree has a root node. All attribute values that appear on the left side of a rule are sorted by frequency. That is, the most frequently occurring attribute values come first. The first rule abc◊ A is inserted into the tree as the path to the root node. The class label, denoted by (A, 80, 80%), and the support and confidence of the rule are recorded at the last node of the path, i.e., the node of that rule. The second rule, abcd◊A, shares the abc prefix with the first rule. Therefore, a new node d is inserted into the tree by extending it with the path generated by the first rule. Again, the rule's class label, support, and trust are recorded in the last node. This means that the third and fourth rules can be pushed or inserted similarly. All nodes with the same attribute values are connected to the queue using node links. The headers for each queue are stored in a header table. To the left of the rules, 13 cells are required to store the initial rule set. If you use a CR tree, only 9 nodes are needed. As you can see from the example above, the CR tree structure has the following advantages: The CR tree has a compact structure. It saves a lot of rule storage space by checking for potential rule sharing. Experimental results show that using CR trees can often save around 50-60% of space. The CR tree itself is an index of rules. For example, to get all rules with attribute values b and d in the rule set an Example 2, you can traverse the node d starting in the header table and continue searching for b. . After generating the CR tree, rule search becomes efficient. This makes it much easier to refine your rules and use them 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 with respect to 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 $\chi 2$ testing. Only rules that are positively correlated, i.e. those for which the $\chi 2$ 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 $\chi 2$ value is selected. However, simply choosing the rule with the highest $\chi 2$ value can be advantageous for minority classes, as the following example shows.

6. 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 produces 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 (e.g., 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 illustration 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).

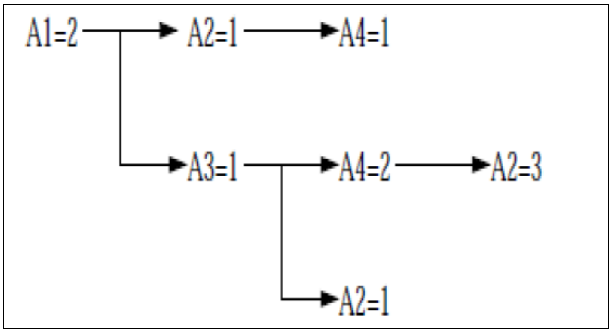


Fig 5: Some Rules Generated by CPAR. CPAR's rule generation takes $O(n_k |R\})$ time.

7. Experimental Results

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

The approaches have been validated using a large set of experiments addressing to 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.
5. 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.

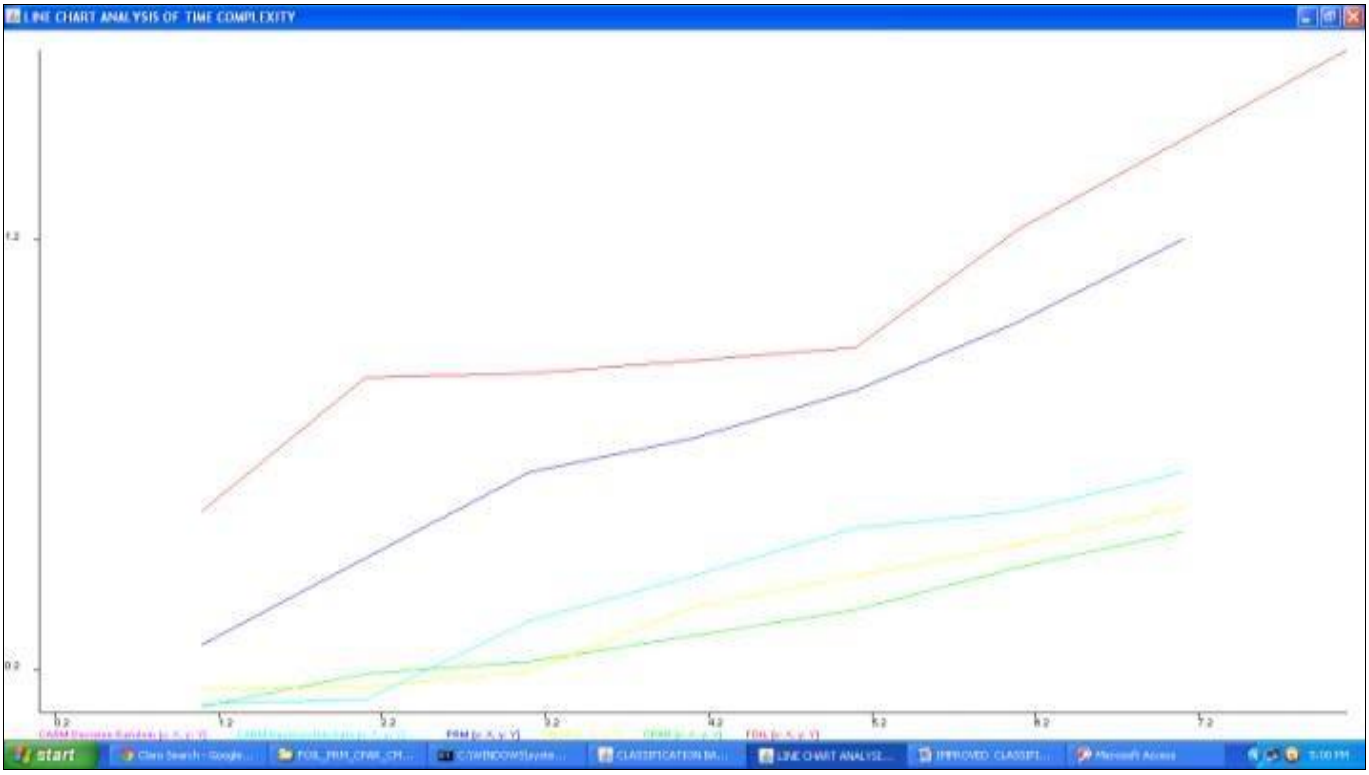


Fig 6: Comparison of Time Complexity of algorithms

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

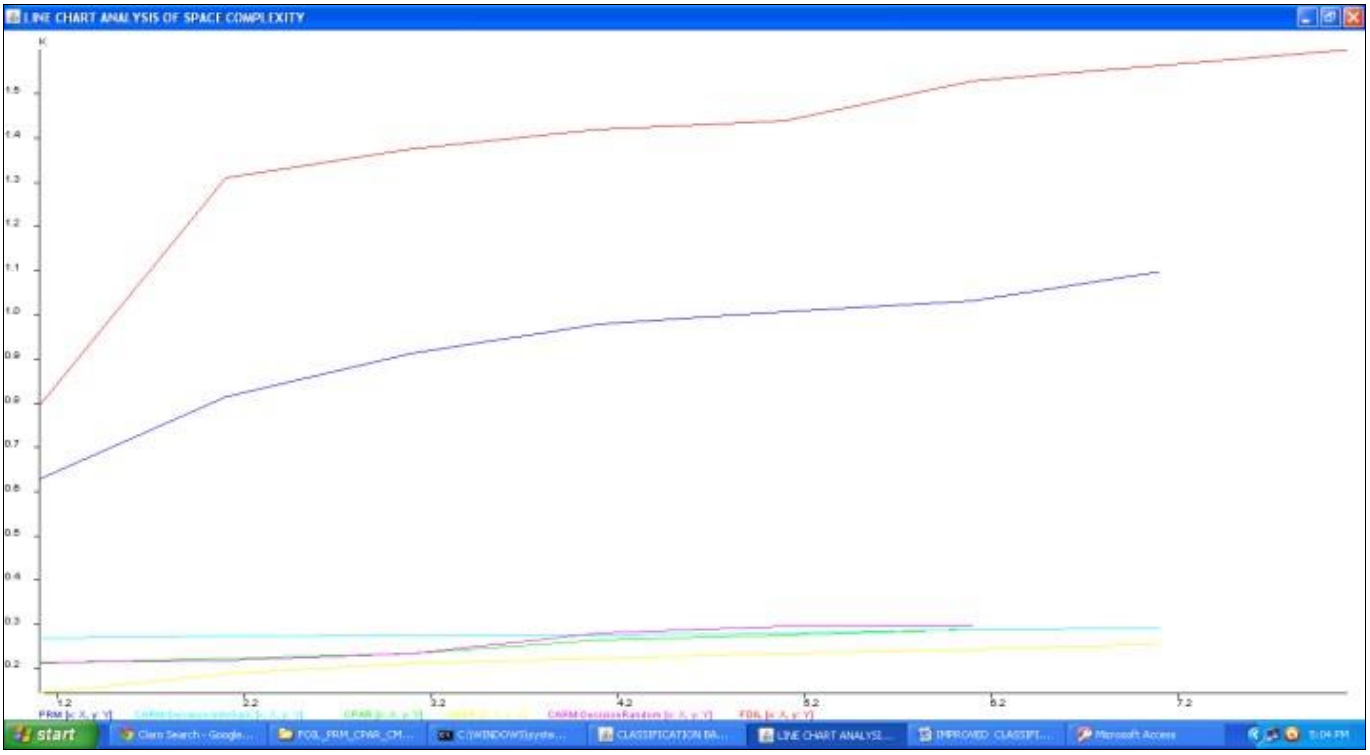


Fig 7: Space Complexity comparison of algorithms

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

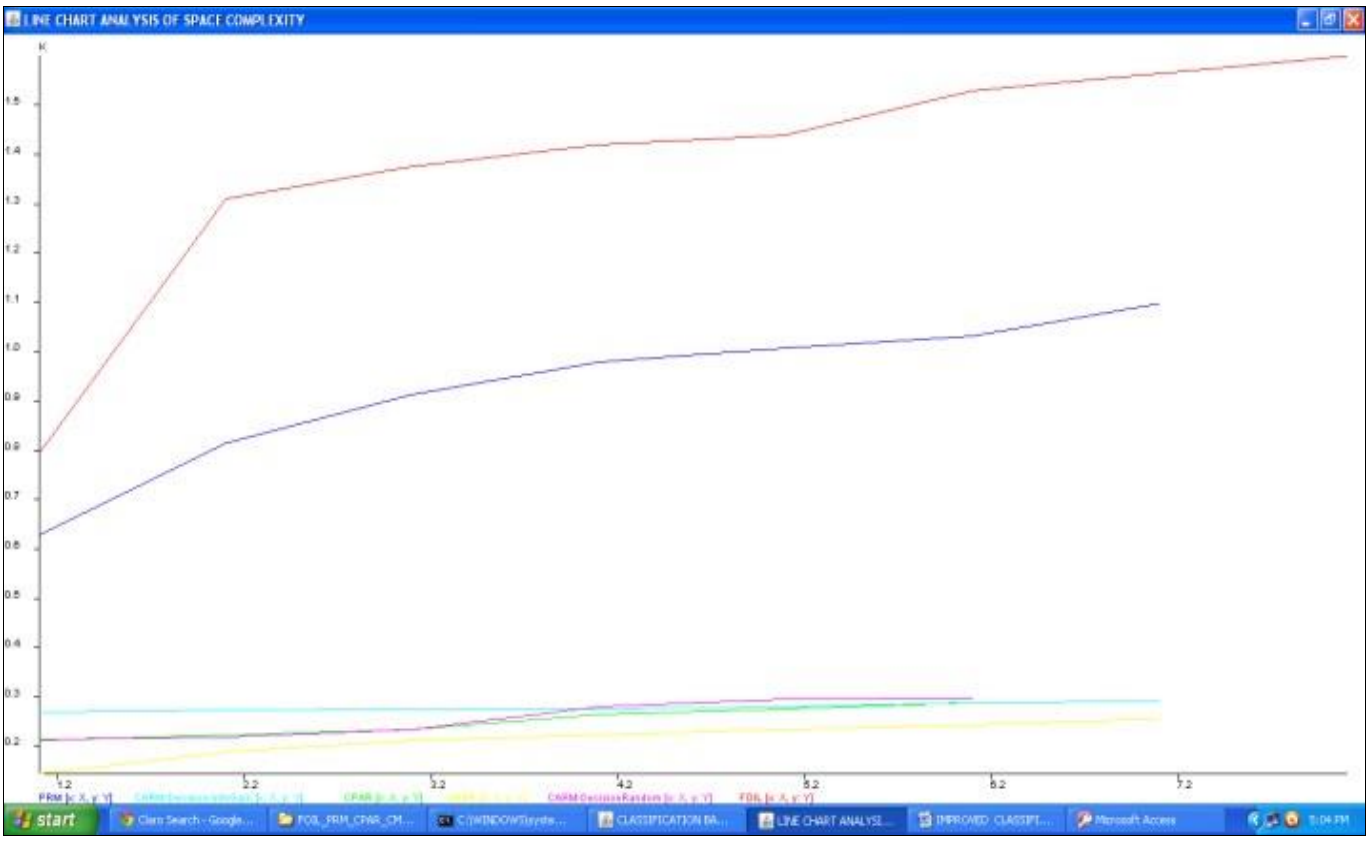


Fig 8: Comparison of Accuracy of algorithms.

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

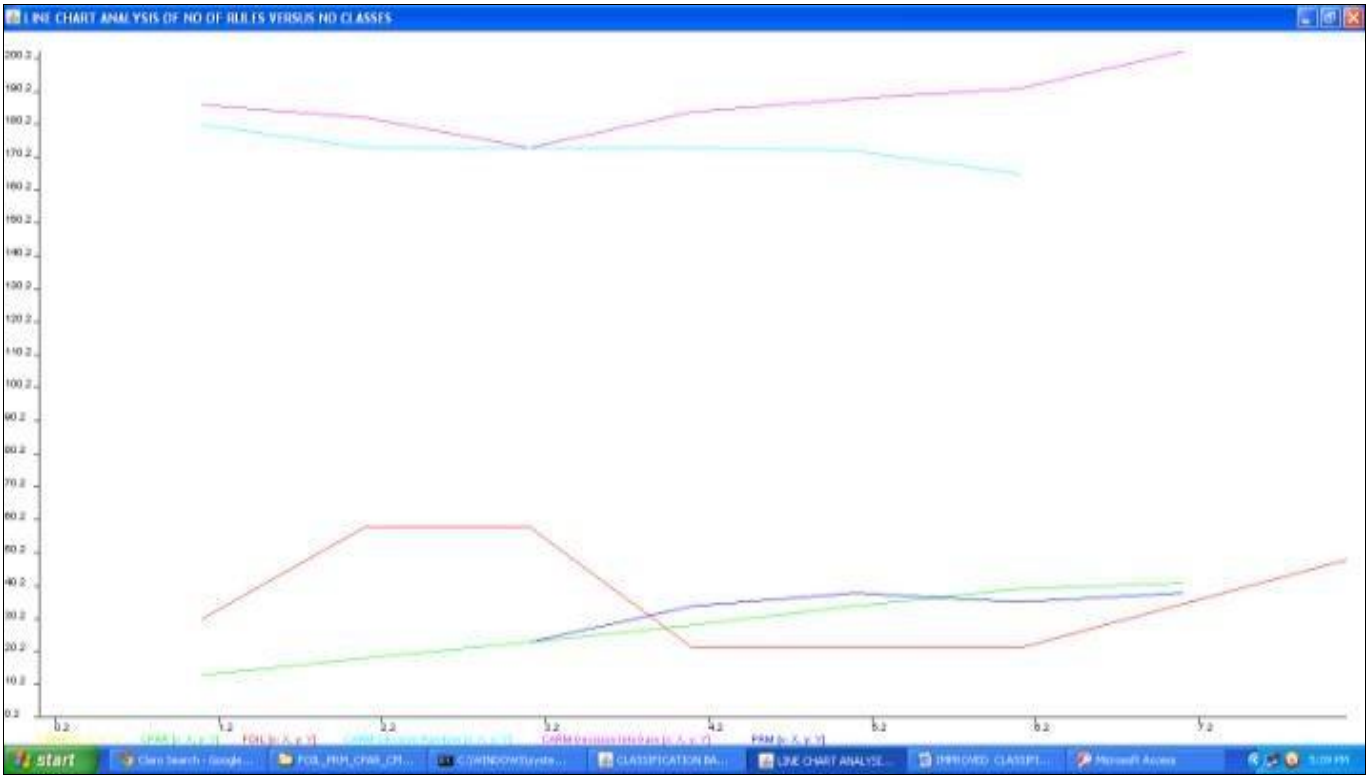


Fig 9: Comparison of no. of Rules of algorithms

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

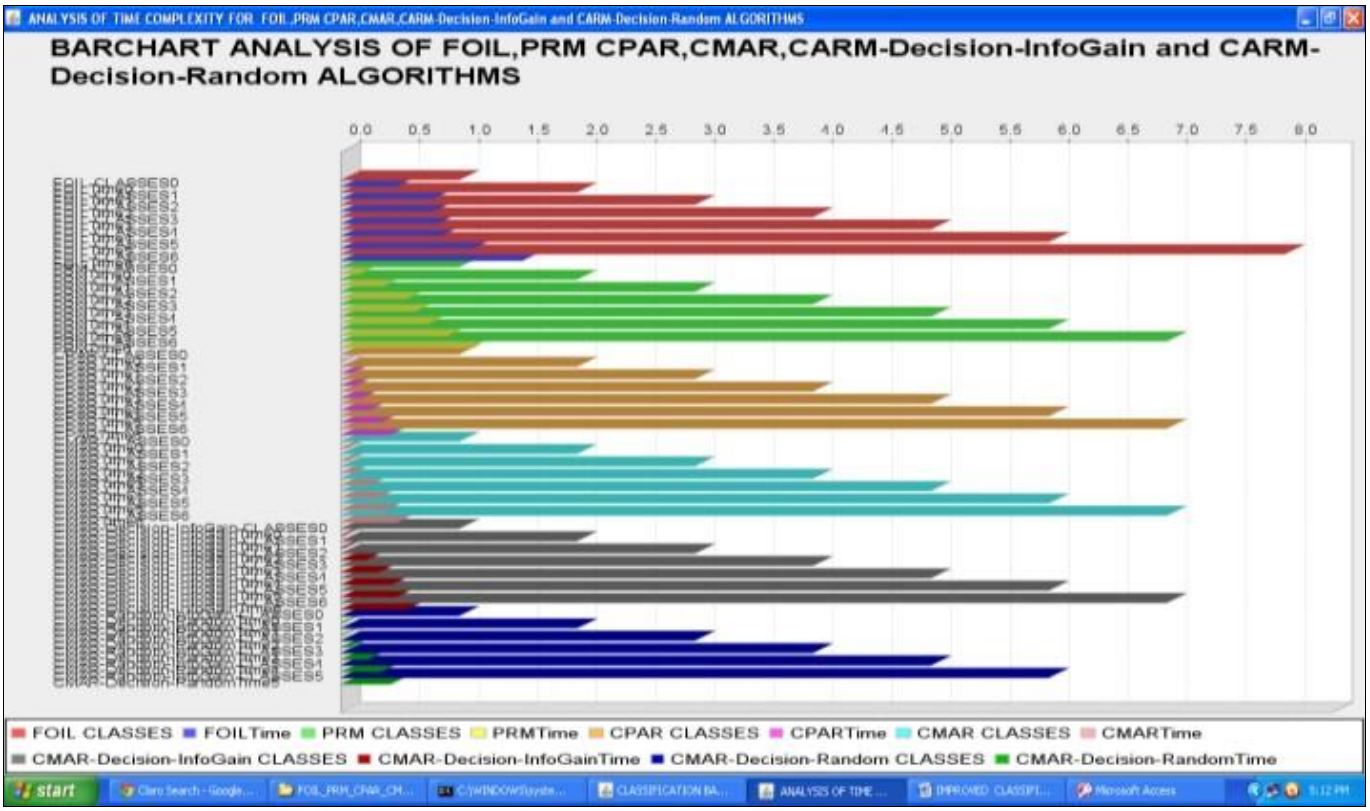


Fig 10: Comparison of Time Complexity of algorithms.

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

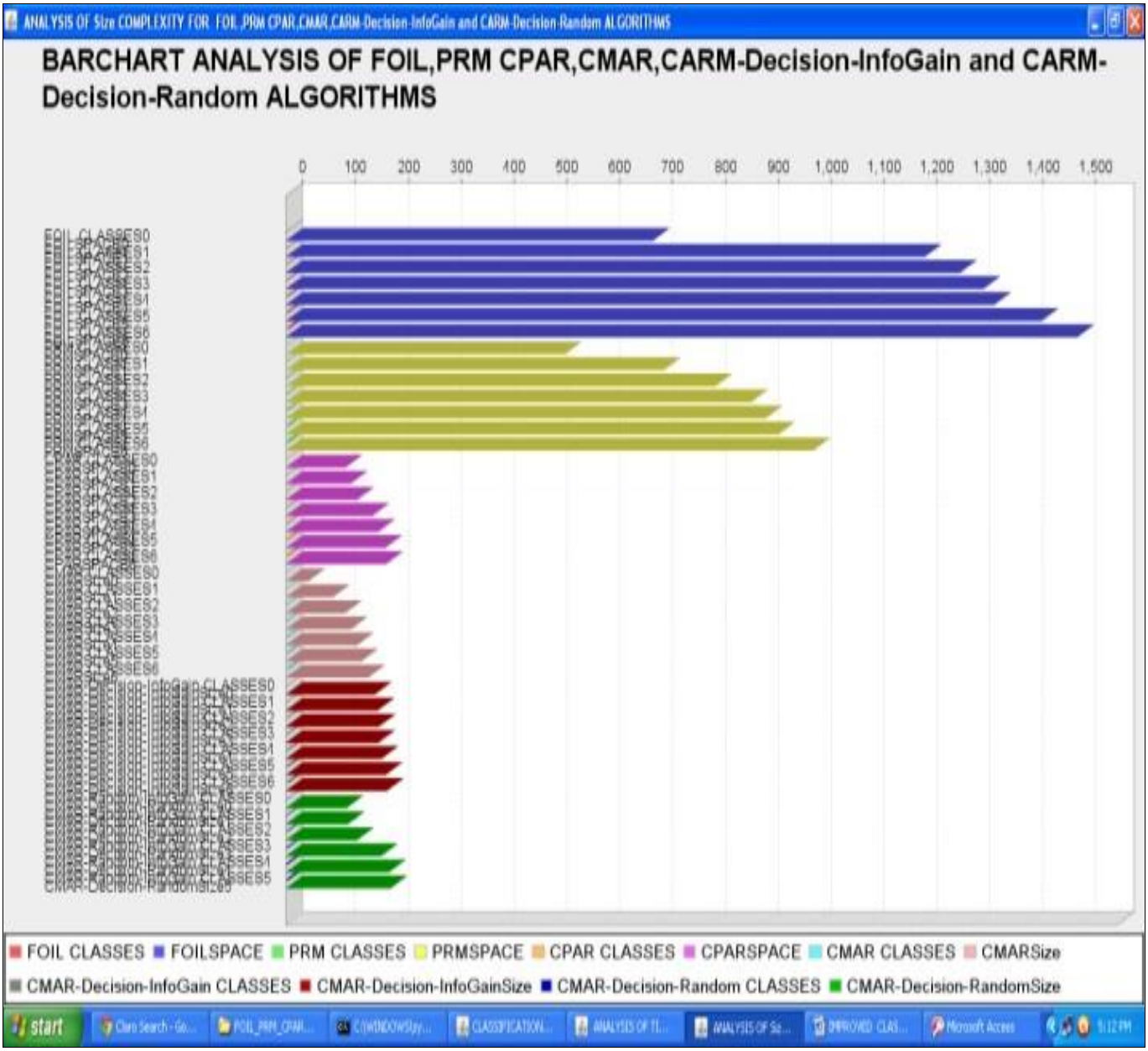


Fig 11: Space Complexity comparison of algorithms

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

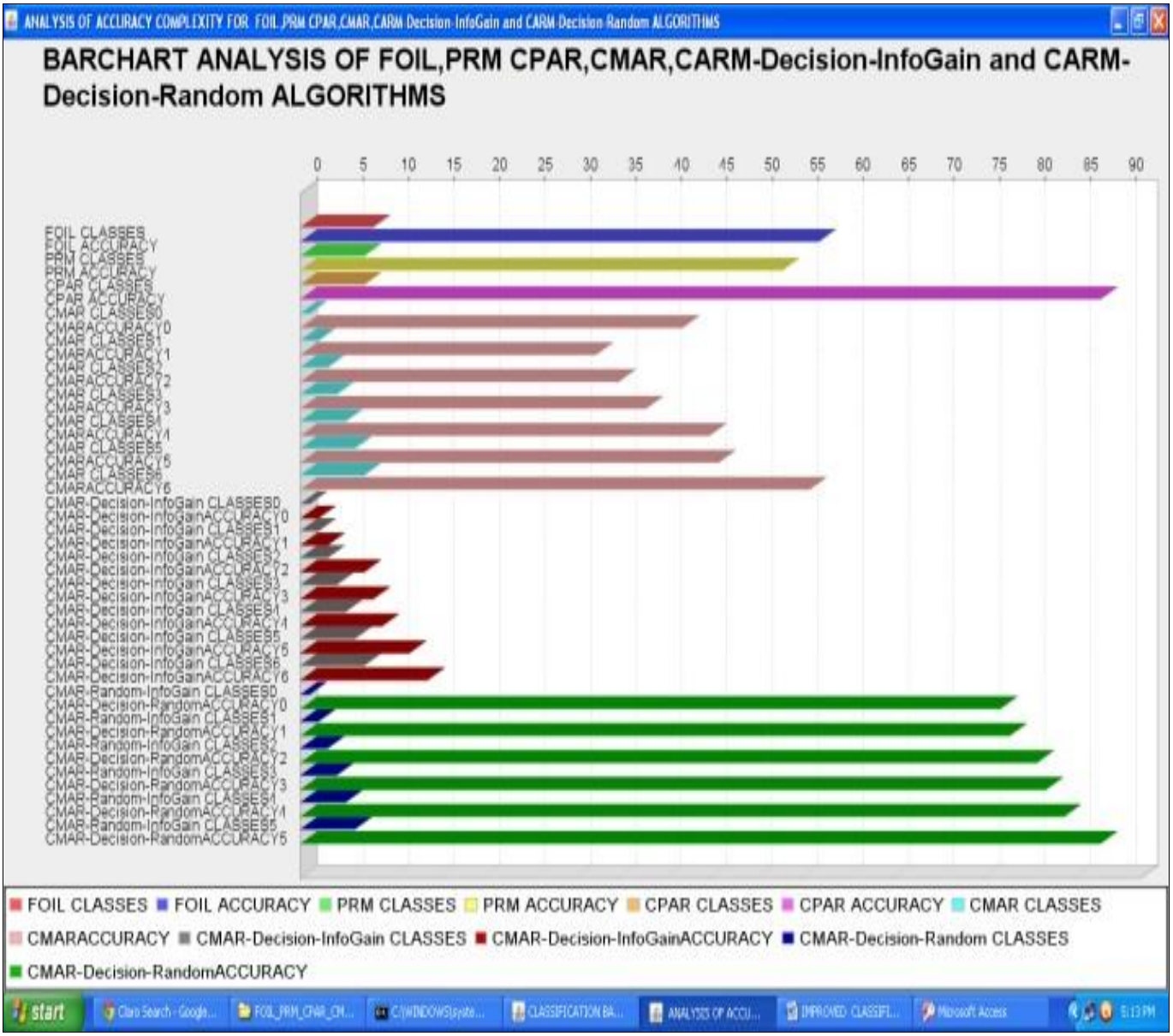


Fig 12: Comparison of Accuracy of algorithms.

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

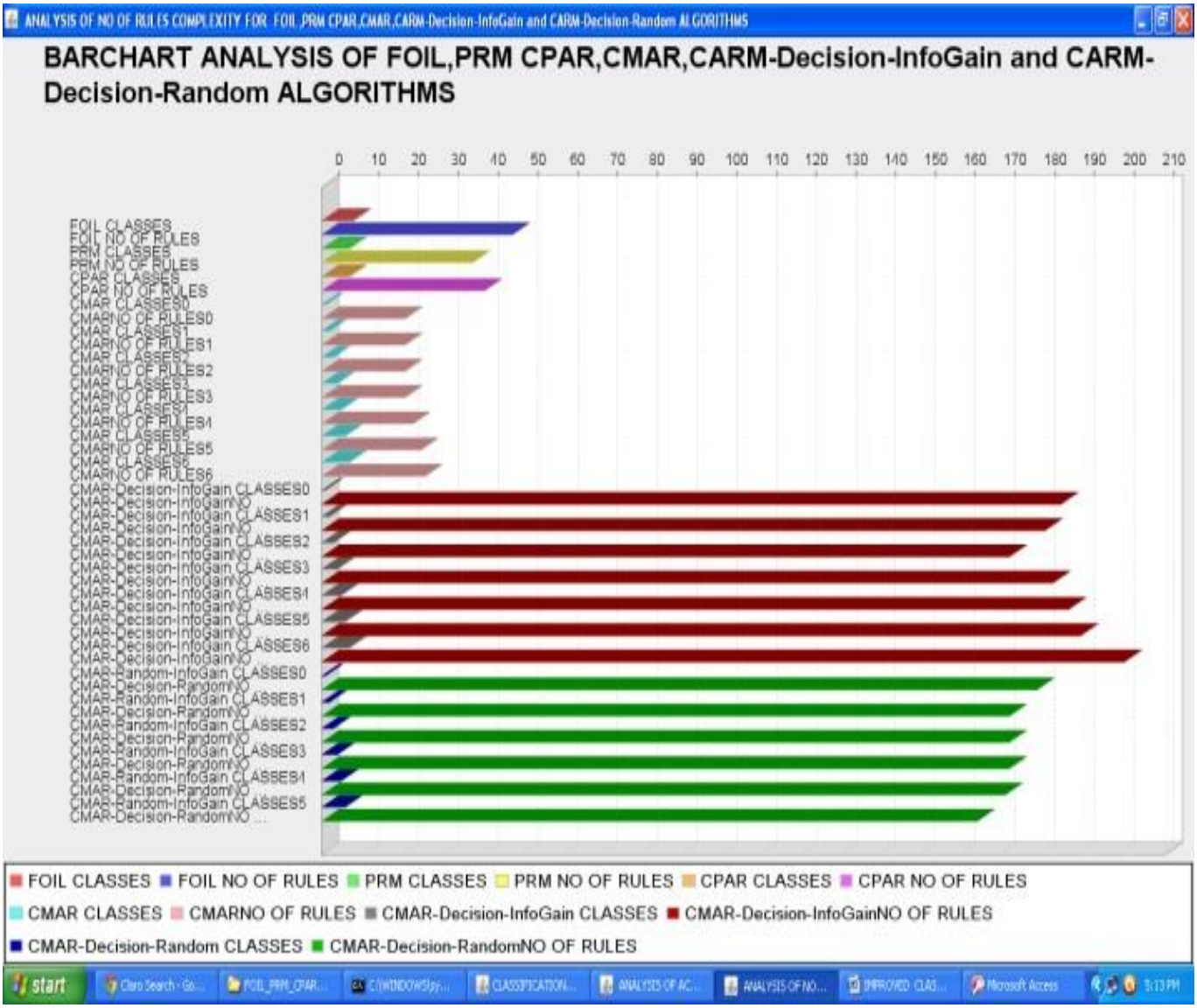


Fig 13: Comparison of No of Rules of algorithms.

8. Conclusion

In this paper, we investigate two main problems in association rule mining classification. (1) Productive efficiency in processing a large number of discovered association rules and (2) Efficacy as well as efficiency in predicting new class labels with high classification accuracy. We proposed two new association classification methods: CMAR, a classification based on multiple association rules, and CPAR, a classification based on predictive association rules. The CMAR method has several unique features. (1) Classification is performed based on a weighted $\chi 2$ analysis imposed by multiple association rules, which improves the overall classification accuracy. (2) Effectively reduce rules based on trust, correlation, and database coverage. (3) We extend efficient methods such as frequent pattern analysis, FP growth, building FP trees associated with class distributions, and applying CR tree structures to achieve efficiency by efficiently storing and retrieving the discovered association rules. CPAR is designed to integrate classification and association rule analysis. Performance studies show that CPAR achieves high accuracy and efficiency, which can be explained by the following important characteristics: (1) we use a greedy rule generation approach, which is much more efficient than generating all candidate rules. (2) A dynamic programming approach to avoid repeated computations when generating rules, (3) selecting multiple literals and generating multiple rules simultaneously, (4) evaluating the rules using expected accuracy and which one is best for prediction to be used. CPAR represents a new approach to efficient, high-quality classification. Experiments show that both CMAR and CPAR perform better than FOIL and PRM.

10. References

1. Devasri Rai, Thoke AS, Keshri Verma. Enhancement of Associative Rule based FOIL and PRM Algorithms, Proc; c2012.
2. Quinlan JR. C4.5: Programs for Machine Learning. San Mateo, Morgan Kaufmann, San Francisco; c1993.
3. Clark P, Niblett T. The CN2 induction algorithm. Machine learning. 1989 Mar;3:261-83.
4. Berry MJA, Linoff GS. [Memory based reasoning(Data Mining Techniques); c2004
5. Andrews R, Diederich J, Ticke A. A survey and critique of Techniques for extracting rules from trained artificial neural networks, “Knowledge Based Systems; c1995. p. 373-389.
6. Langely P, IBA W, Thomson K. An analysis of

Bayesian Classifiers, in National Conference on Artificial Intelligence; c1992. p. 223-228
7. Liu B, Hsu W, 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, The AAAI press; c1998 Aug.
8. Quinlan JR, Cameron-Jones RM. FOIL: A midterm report. In Proc. 1993 European Conf. Machine Learning. pp. 3{20, Vienna Austria; c1993.
9. Yin X, Han J. CPAR: Classification Based on Predictive Association Rules, Proceedings of Siam International conference on Data Mining; c2003. p. 331-335.
10. Prafulla Gupta, Durga Toshniwal. Performance Comparision of Rule Based Classification Algorithms, International Journal of Computer Science & Informatics. 2011;1(2):37-42.
11. Li W, Han J, Pie J. CMAR: Accurate and efficient Classification based on multiple class-association rules. In ICDM01, 2011, 369. {376. San Jose, CA; c2001 Nov.
12. Thabtah F, Cowling P, Peng YH. MMAC: A New Multi-Class, Multi-Label Associative Classification Approach. Fourth IEEE International Conference on Data Mining (ICDM04); c2004.
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 CI, Merz CJ. UCI repository of machine learning data bases from www.ics.uci.edu/~ mlearn/MLrepository.html; c1998.
15. Coenen F. The LUCS-KDD Discretized / Normalized ARM and CARM data library; c2003. 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. 2008 Aug;1(3):262-273.
17. Xin Lu, Barbara Di Eugenio, 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, Sulieman Bani-Ahmad. Classification based on association-rule mining techniques: A general survey and empirical comparative evaluation, Ubiquitous Computing and Communication Journal. 2010;5(3):9-17. www.ubicc.org.
19. Zemirline A, Lecornu L, Solaiman B, Ech-cherif A. An Efficient Association Rule Mining Algorithm for Classification, L. Rutkowski, *et al*. (Eds.): ICAISC, LNAI 5097, C Springer-Verlag Berlin Heidelberg; c2008. p. 9717-728.
20. Bing Liu Wynne Hsu Yiming Ma, Integrating Classification and Association Rule Mining, Appeared in KDD-98, New York; c1998 Aug, 27-31.