ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 5, May 2013

Improved Classification Based Association Rule Mining

Bangaru Veera Balaji[1], Vedula Venkateswara Rao[2]

M.Tech Final, Department of CSE, Sri Vasavi Engineering College, Tadepalligudem, India[1]
Associate Professor, Department of CSE, Sri Vasavi Engineering College, Tadepalligudem, India [2]

Abstract: Association rule mining and classification are two important techniques of data mining in knowledge discovery process. Integration of these two techniques is an important research focus and has many applications in data mining. Integration of these two techniques has produced new approaches called Class Association Rule Mining or Associative Classification Technique. These two combined approaches provide better classification accuracy in classifying the data. Content based information retrieval research areas require high efficiency and performance. In these applications association rule mining discovers association patterns from data and based on association patterns we classify target classes. Our paper mainly focuses on combining classification and association rule mining for classifying the data accurately. 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 combines the advantages of both associative classification and traditional rule-based classification. Instead of generating a large number of candidate rules as in associative classification, CPAR adopts a greedy algorithm to generate rules directly from training data. Moreover, CPAR generates and tests more rules than traditional rule-based classifiers to avoid missing important rules. To avoid over fitting, CPAR uses expected accuracy to evaluate each rule and uses the best k rules in prediction. CMAR applies a CR-tree structure to store and retrieve mined association rules efficiently, and prunes rules effectively based on confidence, correlation and database coverage. The classification is performed based on a weighted $\chi2$ analysis using multiple strong association rules. Our extensive experiments show that CMAR is consistent, highly effective at classification of various kinds of databases and has better average classification accuracy in comparison with FOIL (First Order Inductive Learner) and PRM (Predictive Rule Mining). The proposed algorithms are superior in terms of memory requirements, time complexity and eliminate intermediate data structures in implementation.

Keywords: Association Rule Mining, Classification, Data Mining , Knowledge Discovery, FOIL(First Order inductive Learner), PRM(Predictive Rule Mining), CMAR(Classification Based on Multiple-class Association Rules), CPAR(Classification Based on Predictive Association Rule) , CBA(Classification Based Association).

I. INTRODUCTION

Classification rule mining and Association rules mining are the two important data mining techniques. Classification rule mining is used to discover a small set of rules in the database to form an accurate classifier. Association rules mining are used to reveal all the interesting relationship in a potentially large database. Association rule miming finds all rules in the database that satisfy some minimum support and minimum confidence threshold. For association rule mining, the target of the discovery is not predetermined, while for classification rule mining there is one and only one predetermined target. These two techniques can be integrated to form a framework called Associative Classification method. The integration is done in order to get a special subset of association rules whose right-hand re restricted to classification class attribute. These subsets of rules are referred as Class Association Rules. The use of association rules for classification is restricted to problems where the instances can only belong to a discrete number of classes. The reason is that association rule mining is only possible for categorical attributes. The head Y of an arbitrary association rule X → Y is a disjunction of items. However, association rules in their general form cannot be used directly. We have to restrict their definition. Every item which is not present in the rule body may occur in the head of the rule. When we want to use rules for classification, we are interested in rules that are capable of assigning a class membership. Therefore we restrict the head Y of a class association rule X → Y to one item. The attribute of this attribute-value-pair has to be the class attribute. A class association rule is obviously a predictive task. By using the discriminative power of the Class Association Rules we can also build a classifier.

Building accurate and efficient classifiers for large databases is one of the essential tasks of data mining and machine learning research. Given a set of cases with class labels as a training set, classification is to build a model (called classifier) to predict future data objects for which the class label is unknown. Previous studies have developed heuristic/greedy search techniques for building classifiers, such as decision trees [10], rule learning [2,4,13,18], naïve Bayes classification [4,9,17], and statistical approaches [8]. These techniques induces a representative subset of rules (e.g., a decision tree or a set of rules) from training data sets for quality prediction. Recent studies propose the extraction of a set of high quality association rules from the training data set which satisfy certain user-specified frequency and confidence thresholds. Effective and efficient classifiers have been built by careful selection of rules, e.g., CBA [9], CAEP [3], and ADT [11]. Such a method takes the most effective rule(s) from among all the rules mined for classification. Since association rules explore highly confident associations among multiple variables, it may overcome some constraints introduced by a decision-tree induction method which examines one variable at a time. Extensive performance studies [6, 9, 3, 11] show that association based classification may have better accuracy in general. In recent years, a new approach called associative classification [7, 6] is proposed to integrate association rule mining [1] and classification. It uses association rule mining algorithm, such as Apriori [1] or FPgrowth [5], to generate the complete set of association rules.

Then it selects a small set of high quality rules and uses this rule set for prediction. The 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, (**C**lassification based on **M**ultiple **A**ssociation **R**ules). CPAR inherits the basic idea of FOIL [9] in rule generation and integrates the features of associative classification in predictive rule analysis. In comparison 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 by considering the set of \already generated" rules; and (3) when predicting the class label of an example, CPAR uses the best k rules that this example satisfies. Moreover, CPAR employs the following features to further improve its accuracy and efficiency: (1) CPAR uses dynamic programming to avoid repeated calculation in rule generation; and (2) when generating rules, instead of selecting only the best literal, all the close-to-the-best literals are selected so that important rules will not be missed. CPAR generates a smaller set of rules, with higher quality and lower redundancy in comparison with associative classification. As a result, CPAR is much more time-efficient in both rule generation and prediction but achieves as high accuracy as associative classification. CMAR selects a small set of high confidence, highly related rules and analyzes the correlation among those rules. To avoid bias, we develop a new technique, called weighted $\chi 2$, which derives a good measure on how strong the rule is under both conditional support and class distribution. An extensive performance study shows that CMAR in general has higher prediction accuracy than CBA [9] and C4.5 [10]. Second, to improve both accuracy and efficiency, CMAR employs a novel data structure, CR-tree, to compactly store and efficiently retrieve a large number of rules for classification. CR-tree is a prefix tree structure to explore the sharing among rules, which achieves substantial compactness. CR-tree itself is also an index structure for rules and serves rule retrieval efficiently. Third, to speed up the mining of complete set of rules, CMAR adopts a variant of recently developed FP-growth method. FP-growth is much faster than Apriori-like methods used in previous association-based classification, such as [9, 3, 11], especially when there exist a huge number of rules, large training data sets, and long pattern rules.

II. RELATED WORK

The data analysis algorithms (or data mining algorithms, as they are more popularly known nowadays) can be divided into three major categories based on the nature of their information extraction [1]: Clustering (also called segmentation or unsupervised learning), Predictive modelling (also called classification or supervised learning), and Frequent pattern extraction. Clustering is the 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 fashion. One of the oldest algorithms for clustering is k-means [2]. The two disadvantages of this algorithm are initialization problem and that the cluster must be linearly separable. To deal with the initialization problem, the global k-means has been proposed [3], which is an incremental-deterministic algorithm that employs k-means as a local search procedure. Kernel k-means algorithm [4] avoids the limitation of linearly separable clusters and it mapped the data points from input space to a higher dimensional feature through a nonlinear transformation Ø and the k-means is applied in the feature space. Global kernel k-means [5] is an algorithm which mapped data points from input space to a higher dimensional feature space through the use of a kernel function and optimizes the clustering error in the feature space by locating near-optimal solution.

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

At the end of a pass, the support for a candidate itemset is compared with the minsupport. At the same time it is determined if an itemset should be added to the frontier set for the next pass. The algorithm terminates when the frontier set is empty. After finding all the itemsets that satisfy minsupport threshold, association rules are generated from that itemsets. Bing Liu and et al.[8] had proposed an Classification Based on Associations (CBA) algorithm that discovers Class Association Rules (CARs). It consists of two parts, a rule generator, which is called CBA-RG, is based on Apriori algorithm for finding the association rules and a classifier builder, which is called CBA-CB. In Apriori Algorithm, itemset (a set of items) were used while in CBA-RG, ruleitem, which consists of a condset (a set of items) and a class. Class Association Rules that are used to create a classifier in [8][9] is more accurate than C4.5 [2][3][16] algorithm. But the Classification Based on Associations (CBA) algorithm needs the ranking rule before it can create a classifier. Ranking depends on the support and confidence of each rule.

It makes the accuracy of CBA less precise than Classification based on Predictive Association Rules. Neural network is a parallel processing network which generated with simulating the image intuitive thinking of human, on the basis of the research of biological neural network according to the features 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 associated knowledge of input and output. Initially, the application of the neural network in data mining was not optimistic, because neural networks may have complex structure, long training time, and uneasily understandable representation of results. But its advantages such as high affordability to the noise data and low error rate, the continuously advancing and optimization of various network training algorithms, especially the continuously advancing and improvement of various network pruning algorithms and rules extracting algorithm, make the application of the neural network in the data mining increasingly favored by the overwhelming majority of users. Xianjun Ni [10] describes Data mining process based on neural network. This process is composed of three main steps as data preparation, rule extraction and rules assessment. Classification presently is considered one of the most common data mining tasks [14, 20]. Classifying real world instances is a common thing anyone practices through his life. One can classify human beings based on their race or can categorize products in a supermarket based on the consumers shopping choices. In general, Classification involves examining the features of new objects and trying to assign it to one of the predefined set of classes [38]. Given a collection of records in a data set, each record consists of a group of attributes; one of the attributes is the 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 extracting knowledge 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 a root node, and then it makes a branch for each possible level of that attribute. This will split the training instances into subsets, one for each possible value of the attribute. The same process will be repeated until all instances that fall in one branch have the same classification or the remaining instances cannot be split any further.

The separate-and-conquer approach, on the other hand, starts by building up the rules in greedy fashion (one by one). After a rule is found, all instances covered by the rule will be deleted. The same process is repeated until the best rule found has a large error rate. Statistical approaches such as Naïve Bayes [19] use probabilistic measures, i.e. likelihood, to classify test objects. Finally, covering approach [6] selects each of the available classes in turn, and looks for a way of covering most of training objects to that class in order to come up with maximum accuracy rules. Numerous algorithms have been derived from these approaches, such as decision trees [12, 10], PART, RIPPER]and Prism[6].While single label classification, which assigns each rule in the classifier to the most obvious label, has been widely studied [14, 7, 6, 19], little work has been done on multi- label classification. Most of the previous research work to date on multi-label classification is related to text categorization [20]. In this paper, only traditional classification algorithms that generate rules with a single class will be considered..

III. SYSTEM DESIGN AND IMPLEMENTATION

The overall system design of Classification Based Association Rule Mining is described in Figure 1.

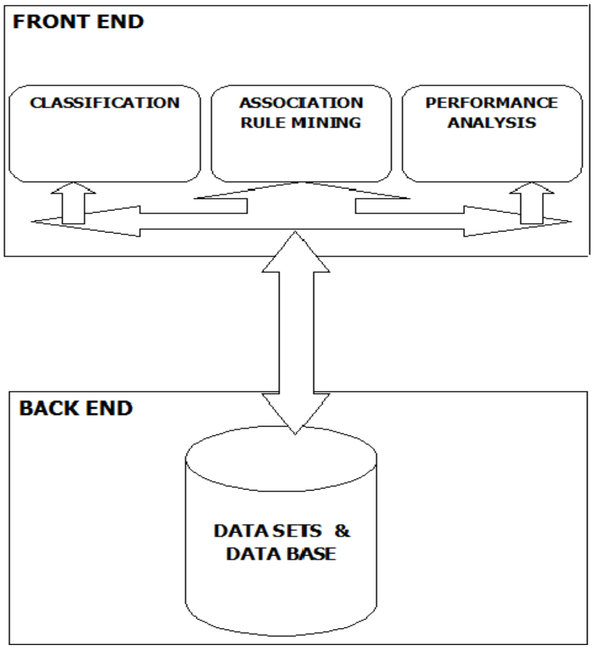

Figure 1. System Architecture

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/ Data Base Module**:

This Module maintains data in the form of data sets.

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

B) Classification Module:

This Module reads the data from data set and performs classification operation and generated classes.

C) Association Rule Generation Module:

This Module uses classes and performs association rule mining and generates frequent item sets, generates association rules.

D) Performance Analysis Module:

This Module computes time complexity, space complexity, accuracy and no of association rules for each execution based on no of classes for different algorithms such as CPAR, CMAR, FOIL and PRM. Then it compares their values and analyzes the efficient algorithms.

IV. Associative Classification

Associative Classification is a special case of association rule discovery in which only the class attribute is considered in the rule's right-hand side (consequent); for example, in a rule such as X → Y, Y must be a class attribute. One of the main advantages of using a classification based on association rules over classic classification approaches is that the output of an Associative Classification algorithm is represented in simple if–then rules, which makes it easy for the end-user to understand and interpret it. Moreover, unlike decision tree algorithms, one can update or tune a rule in Associative Classification without affecting the complete rules set, whereas the same task requires reshaping the whole tree in the decision tree approach. Let us define the Associative Classification problem, where a training data set T has m distinct attributes A1, A2, . . ., Am and C is a list of classes. The number of rows in T is denoted |T|. Attributes can be categorical (meaning they take a value from a finite set of possible values) or continuous (where they are real or integer). In the case of categorical attributes, all possible values are mapped to a set of positive integers. For continuous attributes, a discretization method is used.

Definition 1 A row or a training object in T can be described as a combination of attribute names Ai and values a_{ij}, plus a class denoted by c_j.

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

Definition 3 An item set can be described as a set of disjoint attribute values contained in a training object, denoted <(A_{i1}, a_{i1}), . . ., (A_{ik}, a_{ik})>.

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

Definition 5 The actual occurrence (actoccr) of a ruleitem r in T is the number of rows in T that match the itemset of r.

Definition 6 The support count (suppcount) of ruleitem r is the number of rows in T that match the itemsets of r, and belong to the class c of r.

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

Definition 8 An itemset i passes the minsupp threshold if (occitm(i)/|T|)>=minsupp.

Definition 9 A ruleitem r passes the minsupp threshold if (suppcount(r)/|T|)>=minsupp.

Definition 10 A ruleitem r passes the minconf threshold if (suppcount(r)/actoccr(r)) >=minconf.

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

Definition 12 Any ruleitem r that passes the minsupp threshold is said to be a frequent ruleitem.

Definition 13 A CAR is represented in the form: (Ai1, ai1) ^. . .^ (Aik, aik) → c, where the left-hand side (antecedent) of the rule is an itemset and the consequent is a class.

A classifier is a mapping form H : A → Y, where A is a set of item sets and Y is the set of classes. The main task of Associative Classification is to construct a set of rules (model) that is able to predict the classes of previously unseen data, known as the test data set, as accurately as possible. In other words, the goal is to find a classifier h € H that maximizes the probability that h(a) = y for each test object.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 5, May 2013

An Associative Classification task is different from association rule discovery. The most obvious difference between association rule discovery and Associative Classification is that the latter considers only the class attribute in the rules consequent. However, the former allows multiple attribute values in the rules consequent. Table 1 shows the main important differences between Associative Classification and association rule discovery, where overfitting prevention is essential in Associative Classification, but not in association rule discovery as Associative Classification involves using a subset of the discovered set of rules for predicting the classes of new data objects. Overfitting often occurs when the discovered rules perform well on the training data set and badly on the test data set. This can be due to several reasons such as a small amount of training data objects or noise.

The problem of constructing a classifier using Associative Classifier can be divided into four main steps, as follows.

- Step 1: The discovery of all frequent ruleitems.
- Step 2: The production of all CARs that have confidences above the minconf threshold from frequent ruleitems extracted in Step 1.
- Step 3: The selection of one subset of CARs to form the classifier from those generated at Step 2.
- Step 4: Measuring the quality of the derived classifier on test data objects. occurs when the discovered rules perform well on the training data set and badly on the test data set. This can be due to several reasons such as a small amount 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 the 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

V. GENERATING CLASSIFICATION RULES FOR CMAR

In this section, we develop a new associative classification method, called *CMAR*, which performs **C**lassification based on **M**ultiple **A**ssociation **R**ules. *CMAR* consists of two phases: *rule generation* and *classification*. In the first phase, *rule generation*, *CMAR* computes the *complete* set of rules in the form of R: P → c, where P is a pattern in the training data set, and c is a class label such that sup(R) and conf(R) pass the given support and confidence thresholds, respectively. Furthermore, *CMAR* prunes some rules and only selects 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 matching the object and predicts the class label of the object by analyzing this subset of rules. In this section, we develop methods to generate rules for classification. 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 frequent pattern or association rule mining task [1]. To make mining highly scalable and efficient, *CMAR* adopts a variant of *FP-growth* method [5]. *FP-growth* is a frequent pattern mining algorithm which is faster than conventional *Apriori*-like methods, especially in the situations where there exist large data sets, low support threshold, and/or long patterns. The general idea of mining rules in *CMAR* is shown in the following example.

Example 1 (Mining class-association rules) Given a training data set TH as shown in Table 1. Let the support threshold is 2 and confidence threshold is 50%. *CMAR* mines class-association rules as follows.

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

Table 1 A Training Data Set.

First, *CMAR* scans the training data set T H once, find the set of attribute values happening at least twice in T . The set is F={a1,b2,c3,d1} and is called *frequent item set*. All other attribute values, which fail the support threshold, cannot play any role in the class-association rules, and thus can be pruned. Then, *CMAR* sorts attribute values in F in support descending order, i.e., *F-list =a1-b2-c3-d.* Then, *CMAR* scans the training data set again to construct an *FP-tree*, as shown in Figure 2.

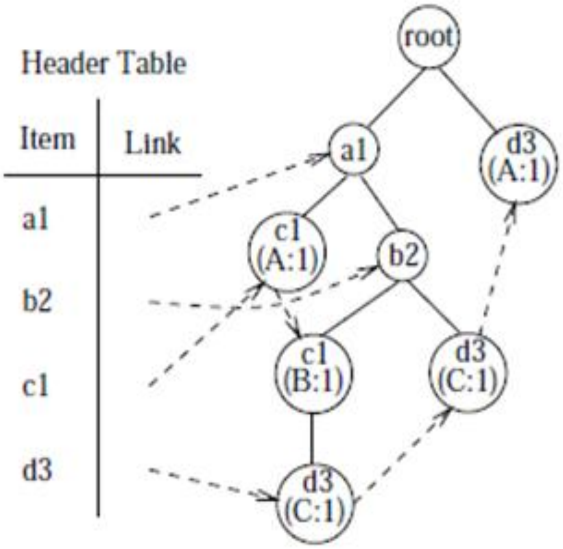

Figure 2. FPTree from training data set

FP-tree is a prefix tree w.r.t. *F-list*. For each tuple in the training data set, attributes values appearing in *F-list* are extracted and sorted according to *F-list*. For example, for the first tuple, (a1,c1) are extracted and inserted in the tree as the left-most branch in the tree. the class label is attached to the last node in the path. Tuples in the training data set share prefixes. For example, the second tuple carries attribute values (a1,b2,c1) in *F-list* and shares a common prefix a1,b2 with the first tuple. So, it also shares the a1,b2 sub-path with the left-most branch. All nodes with same attribute value are linked together as a queue started from the *header table*. Third, based on *F-list*, the set of class-association rules can be divided into 4 subsets without overlap: (1) the ones having d3 ; (2) the ones having c1 but no d3 ; (3) the ones having b2 but no d3 nor c1 ; and (4) the ones having only a1. *CMAR* finds these subsets one by one.

Fourth, to find the subset of rules having d3, *CMAR* traverses nodes having attribute value d3 and look “upward” to collect a d3 projected database, which contains three tuples: (a1,b2,c1,d3) : (a1,b2,d3) : and d3. It contains all the tuples having d3. The problem of finding all frequent patterns having d3 in the whole training set can be reduced to mine frequent patterns in d3 projected database. Recursively, in d3 projected database, a1 and b2 are the frequent attribute values, i.e., they pass support threshold. We can mine the projected database recursively by constructing *FP-tree*s and projected databases. It happens that, in d3projected database, a1 and b2 always happen together and thus a1b2 is a frequent pattern. a1 and b2 are two sub patterns of a1b2 and have same support count as a1b2. To avoid triviality, we only adopt frequent pattern a1b2d3 . Based on the class label distribution information, we generate rule a1b2d3 → C with support 2 and confidence 100%. After search for rules having d3, all nodes of d3 are merged into their parent nodes, respectively. That is, the class label information registered in a d3 node is registered in its parent node. The *FP-tree* is shrunk as shown in Figure 3. Please note that this tree-shrinking operation is done at the same scan of collecting the d3 projected database.

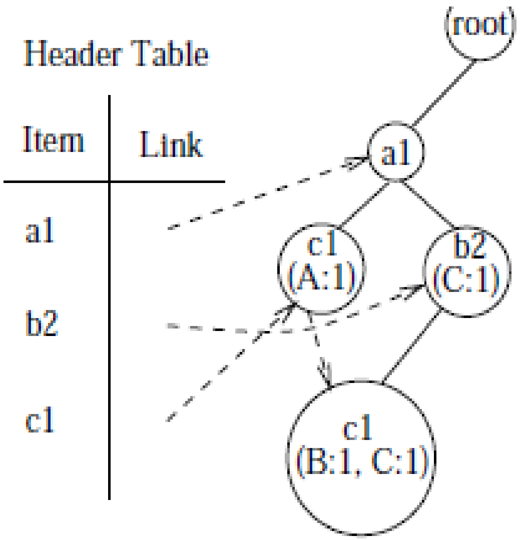

Figure 3. FP Tree merging after nodes of d3.

The remaining subsets of rules can be mined similarly. There are two major differences in the rule mining in *CMAR* and the standard *FP-growth* algorithm. On one hand, CMAR *finds frequent patterns and generates rules in one step.* Conventionally, association rules must be mined in two steps. This is also the case for traditional associative classification methods. First, all the frequent patterns (i.e., patterns passing support threshold) are found. Then, all the association rules satisfying the confidence threshold are generated based on the mined frequent patterns. The difference of *CMAR* from other associative classification methods is that for every pattern, *CMAR* maintains the distribution of various class labels among data objects matching the pattern. This is done without any overhead in the procedure of counting (conditional) databases. Thus, once a frequent pattern (i.e., pattern passing support threshold) is found, rules about the pattern can be generated immediately. Therefore, *CMAR* has no separated rule generation step. On the other hand, CMAR uses class label distribution *to prune*. For any frequent pattern P, let c be the most dominant class in the set of data objects matching / . If the number of objects having class label and matching P is less than the support threshold, there is no need to search any super pattern (superset) P’ of P since any rule in the form of P’ → C cannot satisfy the support threshold either.

A) Storing Rules in CR Tree.

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

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

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%***

Table 2Rules found in training data set.

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

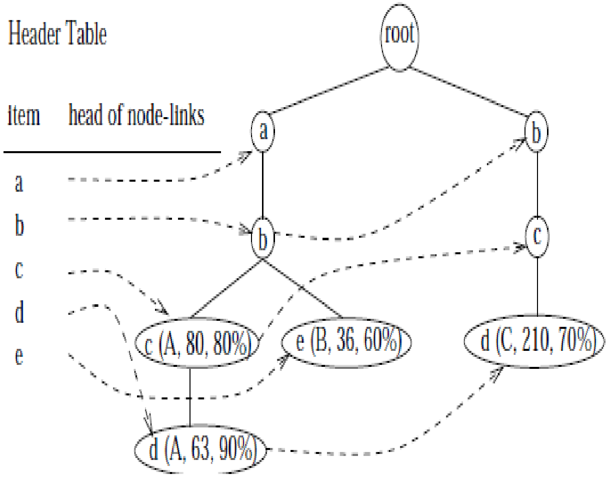

Figure 4 CR Tree for Rules in Example2.

A *CR-tree* has a *root node*. All the attribute values appearing at the left hand side of rules are sorted according to their frequency, i.e., the most frequently appearing attribute value goes first. The first rule, abc→ A is inserted into the tree as a path from 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., node for this rule.

The second rule, abcd→A , shares a *prefix abc* with the first rule. Thus, it is inserted into the tree by extending a new node d to the path formed by the first rule. Again, the class label, support and confidence of the rule are registered at the last node, i.e., d. The third and fourth rules can be inserted similarly. All the nodes with the same attribute value are linked together by *node-link* to a queue. The head of each queue is stored in a *Header table*. To store the original rule set, 13 cells are needed for the left hand sides of the rules. Using *CR-tree*, only 9 nodes are needed.

As can be seen from the above example, the *CR-tree* structure has some advantages as follows. *CR-tree* is a compact structure. It explores potential sharing among rules and thus can save a lot of space on storing rules. Our experimental results show that, in many cases, about 50-60% of space can be saved using *CR-tree*. *CR-tree* itself is an index for rules. For example, if we want to retrieve all the rules having attribute value b and d in the set of rules in Example 2, we only need to traverse node-links of d , which starts at the header table, and keep looking upward for b. Once a *CR-tree* is built, rule retrieval becomes efficient. That facilitates the pruning of rules and using rules for classification dramatically.

B) Pruning Rules.

The number of rules generated by class-association rule mining can be huge. To make the classification effective and also efficient, we need to prune rules to delete redundant and noisy information. According to the facility of rules on classification, a global order of rules is composed. Given two rules R1 and R2 , R1 is said **having higher rank** than R2, denoted as 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 in its left hand side than R2 does. In addition, a rule R1: P → C is said a **general rule** w.r.t. rule R2: P’ → c’ , if and only if P/ is a subset of P’. *CMAR* employs the following methods for rule pruning. First, *using general and high-confidence rule to prune more specific and lower confidence ones*. Given two rules R1 and R2, where I is a general rule w.r.t. R2. *CMAR* prunes R2 if R1 also has higher rank than R2. . The rationale is that we only need to consider general rules withR1high confidence, and thus more specific rules with low confidence should be pruned. This pruning is pursued when the rule is inserted into the *CR-tree*. When a rule is inserted into the tree, retrieval over the tree is triggered to check if the rule can be pruned or 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 by χ2 testing. Only the rules that are positively correlated, i.e., those with χ2 value passing a significance level threshold, are used for later classification. All the other rules are pruned. The rationale of this pruning is that we use the rules reflecting strong implications to do classification. By removing those rules not positively correlated, we prune noise.

After a set of rules is selected for classification, *CMAR* is ready to classify new objects. Given a new data object, *CMAR* collects the subset of rules matching the new object from the set of rules for classification. In this section, we discuss how to determine the class label based on the subset of rules. Trivially, if all the rules matching the new object have the same class label, *CMAR* just simply assigns that label to the new object. If the rules are not consistent in class labels, *CMAR* divides the rules into groups according to class labels. All rules in a group share the same class label and each group has a distinct label. *CMAR* compares the effects of the groups and yields to the strongest group. 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 positively correlated and have good support, the group should have strong effect. There are many possible ways to measure the combined effect of a group of rules. For example, one can use the strongest rule as a representative. That is, the rule with highest $\chi 2$ value is selected. However, simply choosing the rule with highest $\chi 2$value may be favorable to minority classes, as illustrated in the following example.

VI. Generating Rules For Classification using CPAR

CPAR (Classification based on Predictive Association Rules), which combines the advantages of both associative classification and traditional rule-based classification. Instead of generating a large number of candidate rules as in associative classification, CPAR adopts a greedy algorithm to generate rules directly from training data. Moreover, CPAR generates and tests more rules than traditional rule-based classifiers to avoid missing important rules. To avoid over fitting, CPAR uses expected accuracy to evaluate each rule and uses the best k rules in prediction. CPAR stands in the middle between exhaustive and greedy algorithms and combines the advantages of both. CPAR builds rules by adding literals one by one, which is similar to PRM. However, instead of ignoring all literals except the best one, CPAR keeps all close-to-the-best literals during the rule building process. By doing so, CPAR can select more than one literal at the same time and build several rules simultaneously. The following is a detailed description of the rule generation algorithm of CPAR. Suppose at a certain step in the process of building a rule, after finding the best literal p, another literal q that has similar gain as p (e.g., differ by at most 1%) is found. Besides continuing building the rule by appending p to r, q is also appended to the current rule r to create a new rule r0, which is pushed into the queue. Each time when a new rule is to be built, the queue is first checked. If it is not empty, a rule is extracted from it and is taken as the current rule. This forms the depth first-search in rule generation.

Example. Figure 5 shows an example of how CPAR generates rules. After the first literal (A1 = 2) is selected, two literals (A2 = 1) and (A3 = 1) are found to have similar gain, which is higher than other literals. Literal (A2 = 1) is first selected and a rule is generated along this direction. After that, the rule (A1 = 2; A3 = 1) is taken as 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. In this way, three rules are generated:

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

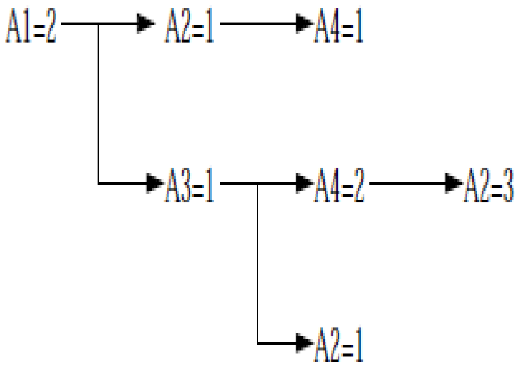

Figure 5 Some Rules Generated by CPAR.

CPAR's rule generation takes O(n_k |R}) time.

VII. Experimental Results

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

We validated our approach by means of a large set of experiments addressing the following issues:

1. Performance of the Classification and association rules, in terms of execution time, memory usage.
2. Performance of the Classification and association rules, in terms of classes and accuracy.
3. Performance of the Classification and association rules, in terms of classes And No of rules generated.
4. Scalability of the approach.

All the experiments are performed on a Core i3 Pentium PC with 2GB main memory, running Microsoft Windows/XP.

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

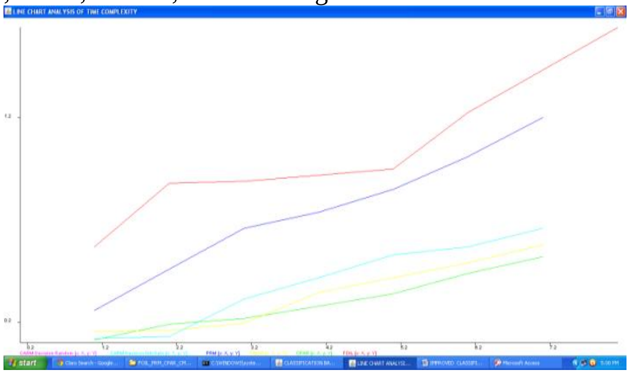

Fig 7.1 Time Complexity comparison of algorithms.

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

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

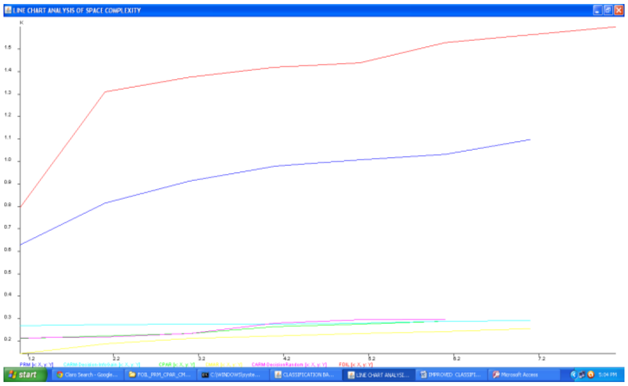

Fig 7.2 Space Complexity comparison of algorithms.

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

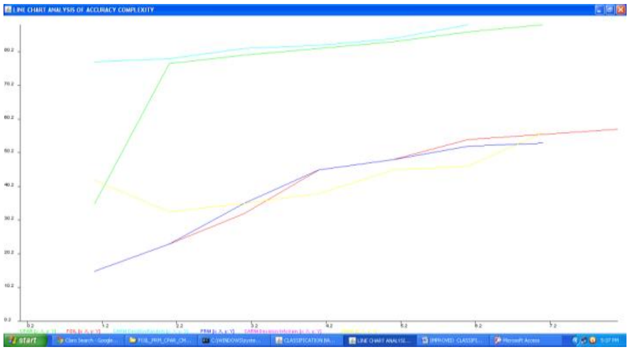

Fig 7.3 Accuracy comparison of algorithms.

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

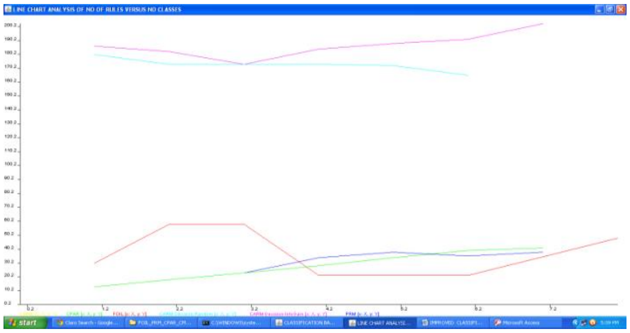

Fig 7.4 No of Rules comparison of algorithms.

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

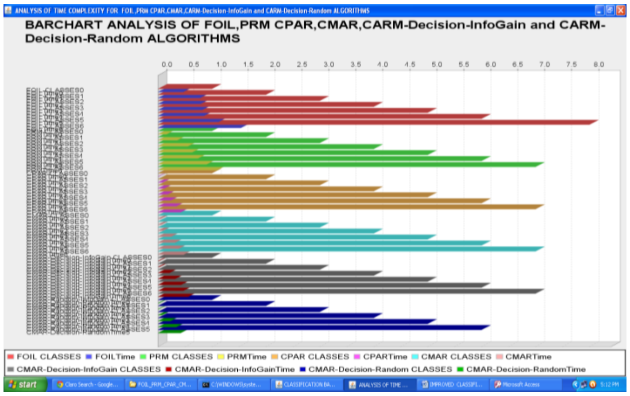

Fig 7.5 Time Complexity comparison of algorithms.

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

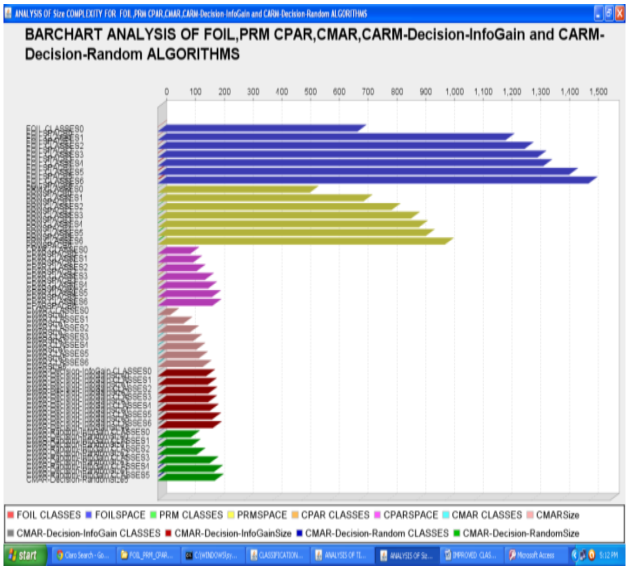

Fig 7.6 Space Complexity comparison of algorithms.

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

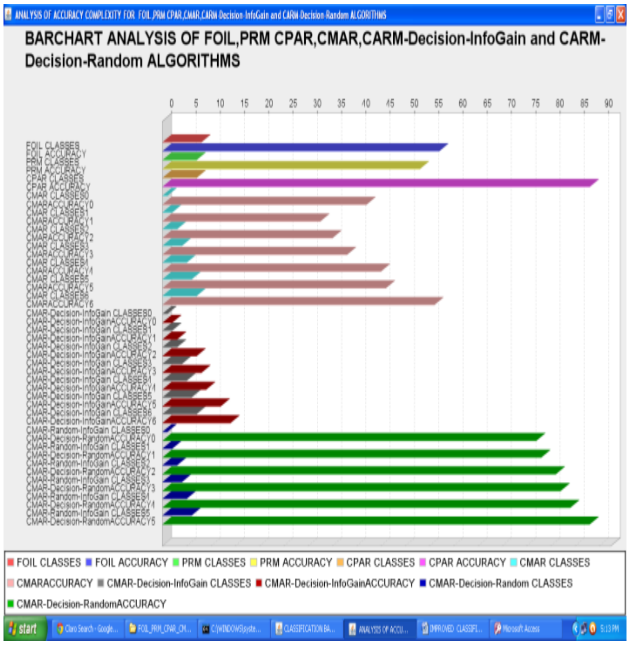

Fig 7.7 Accuracy comparison of algorithms.

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

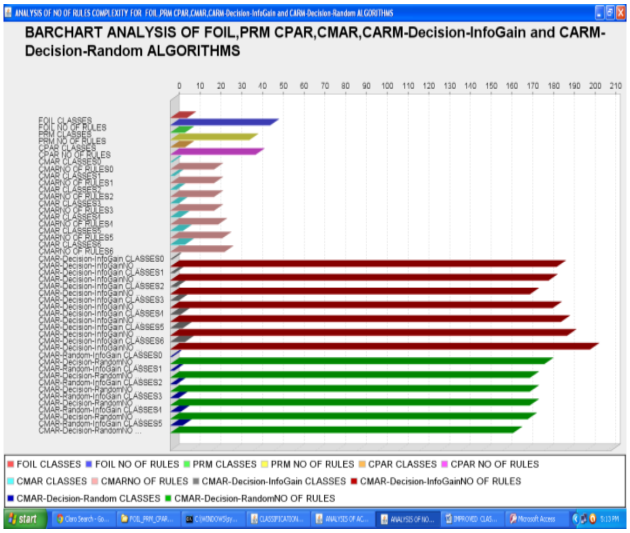

Fig 7.8 No of Rules comparison of algorithms.

VIII. CONCLUSION

In this paper, we examined two major challenges in associative classification: (1) efficiency at handling huge number of mined association rules, and (2) effectiveness at predicting new class labels with high classification accuracy. We proposed two novel associative classification methods, *CMAR*, i.e., **C**lassification based on **M**ultiple **A**ssociation **R**ules and CPAR (Classification Based on Predictive Association Rule).

The method CMAR has several distinguished features: (1) its classification is performed based on a weighted $\chi 2$ analysis enforced on multiple association rules, which leads to better overall classification accuracy, (2) it prunes rules effectively based on confidence, correlation and database coverage, and (3) its efficiency is achieved by extension of an efficient frequent pattern mining method, *FP-growth*, construction of a class distribution-associated *FP-tree*, and applying a *CR-tree* structure to store and retrieve mined association rules efficiently.

CPAR, is developed to integrate classification and association rule mining. Based on our performance study, CPAR achieves high accuracy and efficiency, which can be credited to the following distinguished features: (1) it uses greedy approach in rule generation, which is much more efficient than generating all candidate rules, (2) it uses a dynamic programming approach to avoid repeated calculation in rule generation, (3) it selects multiple literals and builds multiple rules simultaneously, and (4) it uses expected accuracy to evaluate rules, and uses the best

k rules in prediction. CPAR represents a new approach towards efficient and high quality classification.

Our Experiments shows both CMAR and CPAR shows better efficiency than FOIL and PRM.

REFERENCES

[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.san jose, 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.