CLASSIFICATION BASED ON ASSOCIATION-RULE MINING TECHNIQUES: A GENERAL SURVEY AND EMPIRICAL COMPARATIVE EVALUATION

"Alaa Al Deen" Mustafa Nofal and Sulieman Bani-Ahmad
Department of Information Technology
Al-Balqa Applied University Jordan, Al-Salt, 19117
alano911@yahoo.co.uk, sulieman@case.edu

ABSTRACT

In this paper classification and association rule mining algorithms are discussed and demonstrated. Particularly, the problem of association rule mining, and the investigation and comparison of popular association rules algorithms. The classic problem of classification in data mining will be also discussed. The paper also considers the use of association rule mining in classification approach in which a recently proposed algorithm is demonstrated for this purpose. Finally, a comprehensive experimental study against 13 UCI data sets is presented to evaluate and compare traditional and association rule based classification techniques with regards to *classification accuracy*, *number of derived rules*, *rules features* and *processing time*.

Keywords: Data mining, Classification, Association, Associative Classification, MMAC, CBA, C4.5, PART, RIPPER

1. INTRODUCTION

Constructing fast and accurate classifiers for large data sets is an important task in data mining and knowledge discovery. There is growing evidence that merging classification and association rule mining together can produce more efficient and accurate classification systems than traditional classification techniques [26]. In this paper, a recent proposed classification algorithm [37] will be discussed in details.

Classification is one of the most important tasks in data mining. There are many classification approaches for extracting knowledge from data such as statistical [21], divide-and-conquer [15] and covering [6] approaches. Numerous algorithms have been derived from these approaches, such as Naiave Bayes [21], See5 [34], C4.5 [30], PART [14], Prism [6] and IREP [16]. However, traditional classification techniques often produce a *small subset of rules*, and therefore usually miss detailed rules that might play an important role in some cases [29].

Another vital task in data mining is the discovery of association rules in a data set that pass certain user constraints [1, 2]. Classification and association rule discovery are similar except that classification involves prediction of one attribute, i.e., the class, while association rule discovery can predict any attribute in the data set. In the last few years, a new approach that integrates association rule mining with classification has emerged [26, 37, 22]. Few accurate and effective classifiers based on associative classification approach have been presented recently, such as CPAR [39], CMAR [22], MMAC [37] and CBA [26]. Many experimental studies [26, 39, 37] showed that classification based on association rules mining is a high potential approach that constructs more predictive and accurate classification systems than traditional classification methods like decision trees [30, 34]. Moreover, many of the rules found by associative classification methods can not be found by traditional classification techniques.

In this paper, the details of a recent proposed classification based on association rules techniques is surveyed and discussed, which extends the basic idea of association rule [1] and integrates it with classification to generate a subset of effective rules. This proposal uses association rule mining approach in the classification framework. It has been named multi-class classification based on association rules [37]. It utilizes a efficient techniques for discovering frequent itemsets and employs a rule ranking method to ensure that general and detailed rules with high confidence are part of the classification system.

The main contribution of this paper is that several popular association rule-mining techniques are theoretically compared in terms of a number of criteria. Further, a comparison of several classification algorithms is conducted. Moreover, the integration of association rule-mining with classification is also investigated, for that the recently proposed algorithm (the MMAC algorithm) is designed and implemented. Finally, an experimental study to compare MMAC with a set five popular classification algorithms and the MMAC algorithm is conducted using a group of real and artificial benchmark UCI datasets. More specifically, our testbed involves 13 artificial datasets and 10 real world application datasets.

The major findings of the paper are:

- Performance of some simple classification algorithms like OneR is quite well on real world application data, even if they perform poor on artificial data sets.
- There is consistency on the classification accuracy and number of rules produced by decision trees C.45 and PART algorithms.
- Naive Bayes and OneR algorithms are the fastest ones to construct the classification system due the simplicity of such methods in constructing the rules.
- RIPPER on the other hand, is the slowest algorithm in building the classification system due to the optimization phase it employs to deduce the size of the rules set.
- In terms of accuracy, the MMAC algorithm is the best, probably due to the relatively large number of rules it can identify.

2. Association Rule Mining

Since the presentation of association rule mining by Agrawal, Imielinski and Swami in their paper "Mining association rules between sets of items in large databases" in 1993 [1], this area remained one of the most active research areas in machine learning and knowledge discovery.

Presently, association rule mining is one of the most important tasks in data mining. It is considered a strong tool for market basket analysis that aims to investigate the shopping behavior of customers in hoping to find regularities [1]. In finding association rules, one tries to find group of items that are frequently sold together in order to infer items from the presence of other items in the customer's shopping cart. For instance, an association rule may state that " 80% of customers who buy diaper and ice also buy cereal". This kind of information may be beneficial and can be used for strategic decisions like items shelving, target marketing, sale promotions and discount strategies.

Association rules is a valuable tool that has been used widely in various industries like supermarkets, mail ordering, telemarketing, insurance fraud, and many other applications where finding regularities are targeted. The task of association rules mining over a market basket has been described in [1], formally, let D be a database of sales transactions, and let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of binary literals called items. A transaction T in D contains a set of items called itemset, such that $T \subseteq I$. Generally, the number of items in an itemset is called length of an itemset. Itemsets that have a length k are denoted by k-itemsets. Each *i*temset is associated with a statistical threshold named *support*. The support of the itemset is number of transactions in D that contain the itemset. An association rule is an expression $X \Rightarrow Y$, where X, $Y \subseteq I$ are two sets of items and $X \cap Y = \phi$. X is called the *antecedent*, and Y is called the *consequent* of the association rule. An association rule $X \Rightarrow Y$ has a *measure of goodness* named *confidence*, which can be defined as, the probability a transaction contains Y given that it contains X, and is given as support(XY)/support(X).

Given the transactional database D, the association rule problem is to find all rules that have a support and confidence greater than certain user specified thresholds, denoted by *minsupp* and *minconf*, respectively.

The problem of generating all association rules from a transactional database can be decomposed into two subproblems [1].

Table 1 : Transactional Database

Transaction Id	Item	Time
I1	bread, milk, juice	10:12
I2	bread, juice, milk	12:13
I3	milk, ice, bread, juice	13:22
I4	bread, eggs, milk	13:26
I5	ice, basket, bread, juice	15:11

1. The generation of all itemsets with support greater than the *minsupp*. These itemsets are called *frequent* itemsets. All other itemsets are called *infrequent*.
2. For each frequent itemset generated in step1, generate all rules that pass *minconf* threshold. For example if item XYZ is frequent, then we might evaluate the confidence of rules $XY \Rightarrow Z$, $XZ \Rightarrow Y$ and $YZ \Rightarrow X$.

For clarity, consider for example the database shown below in Table 1, and let *minsupp* and *minconf* be 0.70 and 1.0, respectively. The frequent itemsets in Table 1 are {bread}, {milk}, {juice}, {bread, milk} and {bread, juice}. The association rules that pass *minconf* among these frequent itemsets are $milk \Rightarrow bread$ and $juice \Rightarrow bread$.

While the second step of association rule discovery that involves generation of the rules is considerably a straightforward problem given that the frequent itemsets and their *support* are known [1, 2, 18, 23]. The first step of finding frequent itemsets is relatively a resource consuming problem that requires extensive computation and large resource capacity especially if the size of the database and the itemsets are large [1, 28, 4]. Generally, for a number of distinct items m in a customer transaction database D, there are 2^m possible number of itemsets. Consider for example a grocery store that contains 2100 different distinct items. Then there are 2^{2100} possible different combinations of potential frequent itemsets, known by candidate itemsets, in which some of them do not appear even once in the database, and thus usually only a small subset of this large number of candidate itemsets is frequent. This problem has extensively being investigated in the last decade for the purpose of improving the performance of candidate itemsets generation [4, 28, 17, 23, 25, 40]. In this paper, we only considered a number of well known association rule mining algorithms that contributed improvement on the performance in the first step of the mining process. The second step, however, is not considered in this paper.

One of the first algorithms that has significant improvements over the previous association rule algorithms is the Apriori algorithm[2]. The Apriori algorithm presents a new key property named the "downward-closure" of the support, which states that if an itemset passes the *minsupp* then all of its subset must also pass the *minsupp*. This means that any subset of a frequent itemset have to be frequent, where else, any superset of infrequent itemset must be infrequent. Most of the classic association rule algorithms which have been developed after the Apriori algorithm such as [28, 4] have used this property in the first step of association rules discovery. Those algorithms are referred to as the Apriori-like algorithms or techniques.

Apriori-like techniques such as [28, 4, 25] can successfully achieve good level of performance whenever the size of the candidate itemsets is small. However, in circumstances with large candidate itemsets size, low minimum support threshold and long patterns, these techniques may still suffer from the following costs [17]:

- Holding large number of candidate itemsets. For instance, to find a frequent itemset of size 50, one needs to derive more than 2^{50} candidate itemsets. This significantly is costly in runtime and memory usage regardless of the implementation method in use.
- Passing over the database multiple times to check large number of candidate itemsets by pattern matching. The Apriori-like algorithms require a complete pass over the database to find candidate items at each level. Thus, to find potential candidate itemsets of size *n*+1, a merge of all possible combinations of frequent itemsets of size *n* and a complete scan of the database to update the occurrence frequencies of candidate itemsets of size *n*+1 will be performed. The process of repeatedly scan the database at each level is significantly costly in processing time.
- Rare items with high confidence and low support in the database will be basically ignored.

3. CLASSIFICATION IN DATA MINING

3.1 Literature Review

Classification presently is considered one of the most common data mining tasks [14, 24, 30, 39]. 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 [31], separate-and-conquer [15], covering and statistical approaches [24, 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 [21] 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 [32, 30], PART[14], RIPPER [7]and Prism[6].While single label classification, which assigns each rule in the classifier to the most obvious label, has been widely studied [30, 14, 7, 6, 19, 21], 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.

3.2 The Classification Problem

Most of the research conducted on classification in data mining has been devoted to single label problems. A traditional classification problem can be defined as follows: let *D* denote the domain of possible training instances and *Y* be a list of class labels, let *H* denote the set of classifiers for $D \rightarrow Y$, each instance $d \in D$ is assigned a single class y that belongs to *Y*. The goal is to find a classifier $h \in H$ that maximizes the probability that $h(d) = y$ for each test case (*d*, *y*). In multi-label problems, however, each instance $d \in D$ can be assigned multiple labels *y1*, *y2*, …, y_k for $y_i \in y$, and is represented as a pair (*d*, (*y1*, *y2*, …, y_k)) where (*y1*, *y2*, …, y_k) is a list of *ranked class labels* from *y* associated with the instance *d* in the training data. In this work, we only consider the traditional single class classification problem.

4. ASSOCIATIVE CLASSIFICATION

Generally, in association rule mining, any item that passes *minsupp* is known as a **frequent itemset**. If the frequent item consists of only a single attribute value, it is said to be a *frequent one-item*. For example, with *minsupp* = 20%, the frequent one-items in Table 4 are < (AT_1, z_1)>, < (AT_1, z_2)>, < (AT_2, w_1)>, < (AT_2, w_2)> and < (AT_2, w_3)>. Current associative classification techniques generate frequent items by making more than one scan over the training data set. In the first scan, they find the support of one-items, and then in each subsequent scan, they start with items found to be frequent in the previous scan in order to produce new possible frequent items involving more attribute values. In other words, frequent single items are used for the discovery of frequent two-items, and frequent two-items are input for the discovery of frequent three-items and so on.

When the frequent items have been discovered, classification based on association rules algorithms extract a complete set of class-association-rules (CAR) for those frequent items that pass *minconf*.

Table 2. Classification Data Set

Client #	Name	Current Job/year	Income	Criminal History	Loan
1	Sam	5	35K	No	Yes
2	Sara	1	40K	No	No
3	Smith	10	55K	Yes	No
4	Raj	5	40K	No	Yes
5	Omar	1	35K	No	No
6	Sandy	2	25K	No	No
7	Kamal	6	40K	No	Yes
8	Rony	5	34K	No	Yes

Table. 3. Sample of unclassified data set

Client #	Name	Current Job/year	Income	Criminal History	Loan
24	Raba	3	50K	No	?
25	Samy	3	14K	No	?
26	Steve	25	10K	Yes	?
27	Rob	0	45K	No	?

5. CLASSIFICATION BASED ON ASSOCIATION RULE PROBLEM

One of the first algorithms to merge classification with association rules was proposed in [22]. This classification approach consists of two main phases; phase one implements the famous Apriori algorithm [2] in order to discover frequent items. Phase two involves building the classifier. Experimental results indicated that the approach developed in [26] produced rules which are competitive to popular learning methods like decision trees [34].

Table 4: Training data 2

Row#	AT1	AT2	Class
1	Z1	W1	P1
2	Z1	W2	P2
3	Z1	W1	P2
4	Z1	W2	P1
5	Z2	W1	P2
6	Z2	W1	P1
7	Z2	W3	P2
8	Z1	W3	P1
9	Z2	W4	P1
10	Z3	W1	P1

Let T be the training data set with m attributes $AT_1, AT_2, \ldots, AT_m$ and $|T|$ rows. Let P be a list of class labels. An item is defined by the association of an attribute and its value (AT_i, a_i), or a combination of between 1 and m different attributes values. A rule r for classification is represented in the form: $(AT_{i1} = x_{i1}) \wedge (AT_{i2} = x_{i2}) \wedge \ldots \wedge (AT_{in} = x_{in}) \rightarrow p_{i1}$ where the antecedent of the rule is an item and the consequent is a class.

The appearance of the rule in the data set (*Appr*) of a rule r in T is the number of times the antecedent of the rule has been appeared in T. The support frequency (*SuppFreq*) of r is the number of cases in T that matches r*'s* antecedent, and belong to a class p_i. A rule r passes the minimum support threshold (*minsupp*) if for r, the $SuppFreq(r)/\ |T| \geq minsupp$, where $|T|$ is the number of instances in T. A rule r passes the minimum confidence threshold (*minconf*) if $SuppFreq(r)/Appr(r) \geq minconf$.

Any item in T that passes the *minsupp* is said to be a frequent item.

Consider for instance the training data set shown in Table 3 and assume that *minsupp* is set to 0.2 and *minconf* is 0.50. The support of rule $<(AT_1, z_1)>\rightarrow p1$ is 0.30, which satisfies the *minsupp* threshold. The confidence of rule $<(AT_1, z_1)>\rightarrow p1$ is 0.60, and thus this rule also satisfies the *minconf* threshold and therefore it is high potential rule in the classification system.

6. RELATED WORK

The problem of producing rules with multiple labels has been investigated recently in [37] in which the authors propose a new associative classification approach called multi-class, multi-label associative classification (MMAC). The authors also presents three measures for evaluating the accuracy of data mining classification approaches to a wide range of traditional and multi-label classification problems. Results for 28 different data sets show that the MMAC approach is an accurate and effective classification technique, highly competitive and scalable in comparison with other classification approaches.

One of the first algorithms to bring up the idea of using an association rule for classification was the CBA algorithms proposed in [26]. CBA implements the famous Apriori algorithm [2] that requires multiple passes over the training data set in order to discover frequent items. Once the discovery of frequent items finished, CBA proceeds by converting any frequent item that passes the *minconf* into a rule in the classifier. The frequent items discovery and rules generation are implemented in two separate phases by CBA [26].

A method based on association rule for medical image classification has been presented in [3]. It consists of three major phases, phase one involves cleaning up the medical images and extracting target features. Phase two learns rules which are used to build the classifier in phase three [3].

An associative classification algorithm that selects and analyses the correlation between high confidence rules, instead of relying on a single rule, has been developed in [22]. This algorithm uses a set of related rules to make a prediction decision by evaluating the correlation among them. The correlation measures how effective the rules based on their support values and class distributions are. In addition, a new prefix tree data structure named CR-tree to (i) handle the set of rules generated and to (ii) speed up the retrieval process of a rule has been introduced. The CR-tree is proven to be effective in saving storage since many condition parts of the rules are shared in the tree.

7. EMPIRICAL COMPARISON OF MMAC AND THE OTHER TRADITIONAL CLASSIFICATION TECHNIQUES

7.1 Experimental Setup

Experiments on *13* different data sets from the UCI data collection [27] were conducted using stratified ten-fold cross-validation. In cross-validation, the training data set is randomly divided into *10* blocks, each block is held out once, and the classifier is trained on the remaining *9* blocks; then its error rate is evaluated using the held-out block. Thus, the learning procedure is executed ten times on slightly different training data sets [38]. The selected data sets contain only text attributes since a desctitisation method (such as Fayyad and Iran found in [12] to treat continuous attributes) has not been implemented. Few of the selected data sets were reduced by eliminating their integer attributes. Several tests using ten-fold cross-validation have been performed to ensure that the removal of any integer attributes from some of the data sets does not significantly affect the classification accuracy.

Seven popular classification techniques have been compared in term of accuracy rate, runtime and number of rules produced. Namely, the algorithms that we chose are the CBA [26], MMAC [37], PART [14], C4.5 [30], OneR [19], Naïve Bayes [21] and RIPPER [7]. The choice of such methods is based on the multiplicity of strategies strategy they use to produce the rules. For example, C4.5 employs divide-and-conquer strategy to build a decision tree and then converts each path in the tree into a rule. RIPPER, on the other hand, uses separate-and-conquer strategy to build the rules in a greedy fashion. However, PART algorithm combines separate-and-conquer with divide-and-conquer to avoid global optimization and extracts rules faster than decision trees algorithms. CBA and MMAC are classification algorithms that are based on association rule mining techniques that use statistical constraints and depend on the co occurrence of items in the database. Last but not least, Naïve Bayes is a probabilistic algorithm.

To run the experiments; stratified ten-fold cross-validation was used to produce accuracy. Cross-validation is a standard evaluation measure for calculating error rate on data in machine learning.

7.2 Experimental Results and Observations

7.2.1 Observations on time requirement of the different algorithms

All experiments were conducted on a Pentium IV 1.7 Giga Hertz machine with 256 RAM. The experiments of PART, Naïve Bayes, C4.5, RIPPER and OneR algorithms were conducted using the *Weka* software system [42]. CBA experiments were performed on a VC++ implementation version provided by [41]. Finally, MMAC was implemented using Java under MS Windows platform.

Few studies have shown that the support threshold plays a major role in the overall prediction of the classification systems produced by classification based on association rules techniques [26, 22, 37]. If the user sets the support threshold too high, many good quality rules will be ignored, on the other hand, if the support value is set too low, the problem of over fitting arises and many redundant rules will be generated which consequently consumes more processing time and storage. The *minsupp* has been set to 3% for CBA and MMAC experiments since more extensive experiments which are reported in [26] is one of the rates that achieve a good balance between accuracy and the size of the classifiers. Moreover, since confidence threshold does not have larger impact on the behavior of any classification based on association rules algorithms as support value, and therefore it has been set to 30%.

7.2.2 Observations on time accuracy of the different algorithms

Table 5 represents the classification rate of the rules sets generated by all seven algorithms against 13 benchmark problems from the UCI data collection [27]. After analyzing the table, we found out that there is consistency between the decision tree C4.5 and PART algorithms in the sense that both of them outperformed OneR, RIPPER and Naïve Bayes traditional classification techniques. Particularly, decision tree C4.5 outperformed OneR, Naïve Bayes and RIPPER in the datasets with numbers 11, 9 and 9, respectively. On the other hand, PART outperformed OneR, RIPPER and Naïve Bayes in datasets with numbers 12, 9 and 7, respectively. In Fact, all algorithms outperformed OneR in classification accuracy. The consistency between C4.5 and PART algorithms in producing similar error rate figures supports the research done in [4].

The recently proposed classification based on association rules algorithm, i.e. MMAC, performed the best with regards to classification accuracy. In fact, the won-lost-ties scores of MMAC against decision tree C4.5 and PART are 6-6-1 and 8-5-0, respectively. Moreover, MMAC outperformed CBA in dataset number 11. In average, MMAC actually outperformed all the algorithms that we experimented as observed in figure 1. A possible reason for the accurate classification systems produced by MMAC is the fact that MMAC employs a more detailed rules' ranking method that looks for high confidence detailed rules to play part of the classification system.

In an example demonstrated by [37], CBA and MMAC were applied on the training data shown in Table 4 by using a *minsupp* of 20% and *minconf* of 40%

to illustrate the effectiveness of the rules sets derived by both algorithms. Table 6a represents the classification system generated by CBA, which consists of two rules and covers 8 training instances. The remaining two instances form a default class rule that covers 20% of the entire data.

Table 6b represents the classifier produced by MMAC on the same training data, in which more rules have been discovered, i.e. two more rules than the CBA rules set. The MMAC classification system covers nine training instances, and the remaining one forms the default class. Moreover, the default rule of our rules set produced by MMAC covers only 10% of the training data, and therefore it has less impact on the classification of unseen data that may significantly affect the accuracy in the classification, and could lead to deterioration in the overall error rate.

A comparison of the runtime of the decision tree C4.5, PART, RIPPER, Naïve Bayes and OneR algorithms on the *13* data sets from UCI Irvine data collection in order to compare scalability and efficiency has been performed.

After analyzing the chart, OneR and Naïve Bayes is the fastest algorithms to build the set of rules. This due the simplicity of such algorithms in the way they derive the classification accuracy and predict test instances. Moreover, RIPPER takes more time than the other C4.5 and PART to derive rules. In fact, RIPPER takes close to twelve times more processing time to build the classification system than C4.5 and close to ten times more than PART in the experiments. There are consistency on the runtime figures of PART and C4.5 due to that both construct trees as a general approach to derive the rules regardless of the different strategies they use to build the classification systems. This runtime results support the research done in [36, 4].

The runtime results of CBA and MMAC have been conducted separately due to the fact that both of the them use association rule mining as an approach to construct the classification system. Since the association rule approach often generates large number of rules, thus it is unfair to compare MMAC and CBA with the other classification technique in terms of runtime.

Our experiments indicate that MMAC is faster and can identify more rules than CBA in all cases (Table 7 and figure 2). The fast intersection method that MMAC employed to find frequent items reduces gradually its runtime.

Moreover, MMAC algorithm was implemented using Java, whereas CBA has been implemented using VC++. It is expected that runtime results of MMAC should significantly decrease in the case if VC++ had been used to implement it with more code optimization.

A deeper analysis of the rules produced by CBA, RIPPER, PART, C4.5 and MMAC has been conducted to compare the effectiveness of extracted classification systems. Furthermore, the features of the rules derived of such methods are also investigated.

Table 7 shows the size of the rules sets generated by CBA, RIPPER, PART, C4.5 and MMAC algorithms. The table indicates that classification based association rules methods, i.e. CBA and MMAC, often produce larger rules sets than traditional classification methods, i.e. C4.5, PART and RIPPER. This support the research conducted in [26, 39, 37]. RIPPER produces the smallest sized classification systems in all most all the experiments. This is due to the fact that RIPPER produces only general rules with minimal error rates. Analysis of the rules sets indicated that MMAC derives more rules than PART, RIPPER, C4.5 and CBA for the majority of the data sets. A possible reason for extracting more rules is based on the fact that MMAC always looks for high confidence details rules and employs a detailed ranking method to ensure that many detailed rules should take part of the classification. A possible enhancement to MMAC is a post pruning phase to reduce the number of rules derived.

There was consistency in the number of rules derived by decision trees C4.5 and PART for the majority of the data sets. After investigating the trees constructed by C4.5 algorithm at each iteration, we observed that it generates many empty rules which do not cover any single training instance. The reason for these many useless rules appear might be the way that C4.5 splits the training instances on an attribute, but only a small number of the possible values of this attribute occur in the training data.

Table 5. Classification accuracy of C4.5, OneR, PART, RIPPER, CBA , Naive Bayes and MMAC

Data Set No.	**Data Set**	**C4.5**	**OneR**	**PART**	**RIPPER**	**Naïve Bayes**	**CBA**	**MMAC**
1	Weather	50.00	42.85	57.14	64.28	57.14	85.00	71.66
2	Vote	88.27	87.35	87.81	87.35	87.12	87.39	89.21
3	Tic-tac	83.61	69.93	92.58	97.59	70.04	98.60	99.29
4	Sick	93.87	93.87	93.90	93.84	93.90	93.90	93.87
5	Primary-tumor	42.47	27.43	39.52	36.28	45.72	36.49	43.92
6	Mushroom	99.96	98.52	99.96	99.96	98.15	98.92	99.78
7	Lymph	83.78	75.00	76.35	77.70	81.75	75.07	82.20
8	led7	73.34	19.00	73.56	69.34	73.15	71.10	73.20
9	Kr-vs-kp	71.55	66.05	71.93	70.24	70.71	42.95	68.75
10	Heart-s	78.55	81.29	78.57	78.23	78.23	79.25	82.45
11	Heart-c_1	78.21	71.61	81.18	78.87	81.51	79.87	81.51
12	Contact-lenses	83.33	70.83	83.33	75.00	70.83	66.67	79.69
13	Breast-cancer	72.52	65.73	71.32	70.9	71.67	69.66	72.10

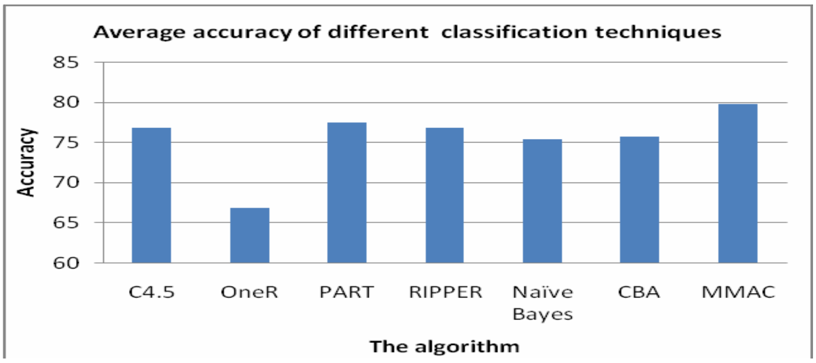

Figure 1: Average accuracy of classification for the algorithms: C4.5, OneR, PART, RIPPER, CBA , Naive Bayes and MMAC based on 13 different UCI data sets.

Table 6a. CBA classifier

RuleId	Frequent Item	Support	Confidence	Class Label
1	Z1	3/10	3/5	p1
3	W1	3/10	3/5	p1
default				p2

Table 6b. MMAC classifier

RuleId	Frequent Item	Support	Confidence	Class Label
1a	Z1	3/10	3/5	p1
1b	Z1	2/10	2/5	P2
2	Z2	2/10	2/4	p2
3	W1	3/10	3/5	p1
default				p1

Table 7. Number of rules of C4.5, PART, RIPPRR, CBA and MMAC algorithms

Data Set No.	Data Set	C4.5	PART	RIPPER	CBA	MMAC
1	Vote	4	13	4	40	84
2	tic-tac	95	50	14	25	26
3	Sick	1	8	2	10	17
4	Primary-tumor	23	22	5	1	28
5	Mushroom	33	18	11	45	48
6	Lymph	12	7	6	38	48
7	led7	37	31	19	50	192
8	Heart-s	2	8	2	22	31
9	Heart-c_1	12	11	4	44	72
10	contact-lenses	4	4	3	6	9
11	breast-cancer	4	20	3	45	71
12	balloon	5	2	2	3	3

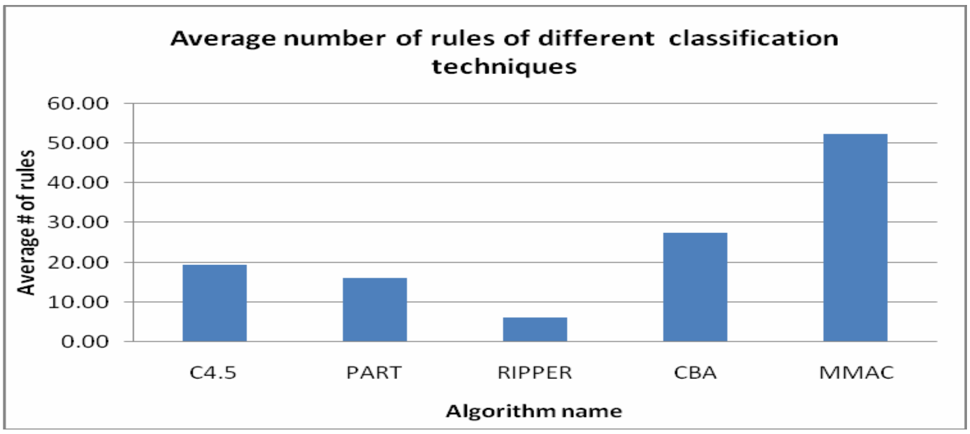

Figure 2: Average number of identified rules using the four algorithms:

C4.5, PART, PIPPER, CBA and MMAC.

8. CONCLUSIONS AND RESULTS

Prediction rate and runtime are important factors in association rule and classification tasks in data mining. In this paper, we investigated the topics of association rules mining algorithms, traditional classification techniques and classification based on association rule mining. The main contributions are:

- Theoretical survey on association rule mining algorithms. This includes the different strategies they use to produce frequent itemsets and rules generation.
- Comparison of traditional classification techniques such as decision trees, Naïve Bayes, RIPPER and others. This include the different strategies they employ to extract the rules from data sets such as divide-and-conquer, covering approach, separate-and-conquer, classification rules and statistical approaches.
- Survey on the use of association rule mining in classification framework. This resulted in a detailed look up into a recently proposed algorithm, i.e. MMAC.
- An extensive experimental comparison studies using several data sets between five popular traditional classification algorithms, i.e. OneR, decision trees C4.5, PART, Naïve Bayes, and RIPPER, and two well known classifications based on a association rule algorithms, i.e. CBA and MMAC, in term of classification rate, runtime and number of rules produced.
- Performance studies on several data sets indicated that there is consistency between C4.5 and PART with regards to error rate, rules features and size of the resulting classifiers. Moreover, OneR surprising, performed well on the real world data sets in which proves that performance of classification algorithms may vary depending on the application data. MMAC performed consistently well in term of classification accuracy on both artificial data sets, i.e. UCI data, and the optimization real world data sets.

The results also pointed out Naive Bayes and OneR algorithms are the fastest ones to construct the classification system due the simplicity of such methods in constructing the rules. RIPPER on the other hand, is the slowest algorithm in building the classification system due to the optimization phase it employs to reduce the size of the rules set.

9. FUTURE WORK

As a future work of this ongoing research, we will be looking at the problem of generating classification systems that predicts more than one class as the antecedent. To accomplish such task, an extensive study on text mining classification algorithms shall be made. Moreover, we will look at how to extract negative association rule which basically represents inferring items from the absence of other items in the customer shopping cart.

10. REFERENCES

[1] Agrawal, R., Amielinski, T., and Swami, A. (1993). *Mining association rule between sets of items in large databases*. In Proceeding of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207-216, Washington, DC, May 26-28.

[2] Agrawal, R. and Srikant, R. (1994). *Fast algorithms for mining association rule*. *Proceedings of the 20th International Conference on Very Large Data Bases***.** pp. 487 - 499.

[3] Antonie, M., Zaïane, O. R., Coman, A. (2003). *Associative Classifiers for Medical Images.* Lecture Notes in Artificial Intelligence 2797, Mining Multimedia and Complex Data, pp 68-83, Springer-Verlag.

[4] Blackmore, K. and Bossomaier, T. J. (2003). *Comparison of See5 and J48.PART Algorithms for Missing Persons Profiling*. Technical report. Charles Sturt University, Australia.

[5] Brin, S., Motwani, R., Ullman, J., Tsur, S. (1997). *Dynamic Itemset Counting and Implication Rules for Market Basket Data*. Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data.

[6] Cendrowska, J. (1987). *PRISM: An algorithm for inducing modular rules*. International Journal of Man-Machine Studies. Vol.27, No.4, pp.349-370.

[7] Cohen, W. W. (1995). Fast effective rule induction. *In the Proceeding of the 12 th International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, pp. 115-123.

[8] Cohen, W. W. (1993). *Efficient pruning methods for separate-and-conquer rule learning systems*. In the proceeding of the 13th International Joint Conference on AI, Chambry, France.

[9] Cowling, P. and Chakhlevitch, K. (2003). *Hyperheuristics for Managing a Large Collection of Low Level Heuristics to Schedule Personnel. Proceeding of 2003 IEEE conference on Evolutionary Computation*, Canberra, Australia, 8-12 Dec 2003.

[10] Dong, G., Li, J. (1999). *Efficient mining of frequent patterns: Discovering trends and differences*. In Proceeding of SIGKDD 1999, San Diego, California.

[11] Elmasri, R., and Navathe, S. B. (2000). *Fundamentals of Database Systems*.. Addison-Wesley, third Edition.

[12] Fayyad, U., and Irani, K. (1993). *Multi-interval discretisation of continues-valued attributes for classification learning*. *IJCAI-93*, pp. 1022-1027, 1993.

[13] Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P. (1996). *Advances in knowledge discovery and data mining*, MIT Press.

[14] Frank, E. and Witten, I. (1998). Generating accurate rule sets without global optimization. In Shavlik, J., ed., *Machine Learning: Proceedings of the Fifteenth International Conference*, pp. 144-151, Madison, Wisconsin, Morgan Kaufmann, San Francisco.

[15] Furnkranz, J. (1996). *Separate-and-conquer rule learning*. *Technical Report* TR-96-25, Austrian Research Institute for Artificial Intelligence, Vienna.

[16] Furnkranz, J. and Widmer, G. (1994). *Incremental reduced error pruning*. In *Machine Learning: Proceedings of the 11th Annual Conference*, New Brunswick, New Jersey, Morgan Kaufmann.

[17] Han, J., Pei, J., and Yin, Y. (2000) Mining frequent patterns without candidate generation.. *2000 ACM SIGMOD Intl. Conference on Management of Data*.

[18] Hipp, J., Güntzer, U., and Nakhaeizadeh, G. (2000). *Algorithms for association rule mining — a general survey and comparison*. SIGKDD Explor. Newsl. 2, 1 (Jun. 2000), 58-64. DOI= http://doi.acm.org/10.1145/360402.360421

[19] Holte, R.C. (1993). *Very simple classification rules perform well on most commonly used data sets*. Machine Learning, Vol. 11, pp. 63-91.

[20] Joachims, T. (1998). *Text categorization with Support Vector Machines: Learning with many relevant features*. Proceeding Tenth European Conference on Machine Learning, pp. 137-142.

[21] John, G. H. and Langley, P. (1995). *Estimating Continuous Distributions in Bayesian Classifiers*. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. pp. 338-345. Morgan Kaufmann, San Mateo.

[22] Li, W., Han, J., and Pei, J. (2001). *CMAR: Accurate and efficient classification based on multiple-class association rule*. In *ICDM'01*, pp. 369-376, San Jose, CA.

[23] Li, J., Zhang, X., Dong, G., Ramamohanaro, K. and Sun, Q. (1999). *Efficient mining of high confidence rules without support thresholds.* A book chapter in the book "Principles of Data Mining and Knowledge Discovery". LNCS.

[24] Lim, T. S., Loh, W. Y., and Shih, Y. S. (2000). *A comparison of prediction accuracy, complexity and training time of thirty-three old and new classification algorithms*. *Machine Learning*, 39.

[25] Liu, B., Hsu, W. and Ma, Y. (1999). *Mining association rules with multiple minimum supports*. In Proceeding of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining(KDD-99), August 15-18, 1999, San Diego, CA.

[26] Liu, B., Hsu, W., and Ma, Y. (1998). *Integrating Classification and association rule mining*. In *KDD '98*, New York, NY, Aug. 1998.

[27] Merz, C. J. and Murphy, P. M. (1996). *UCI Repository of Machine Learning Data- bases.* Irvine, CA, University of California, Department of Information and Computer Science.

[28] Park, J. S., Chen, M., and Yu, P. S. (1995). *An effective hash-based algorithm for mining association rules*. SIGMOD Rec. 24, 2 (May. 1995), 175-186. DOI= http://doi.acm.org/10.1145/568271.223813

[29] Pazzani, M., Mani, S., and Shankle, W. R. (1997). *Beyond Concise and colorful: learning intelligible rules*. In *KDD-97*.

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

[31] Quinlan, J. R. (1987). *Generating production rules from decision trees. Proceedings of the 10th International Joint Conferences on Artificial Intelligence*, pp. 304-307, Morgan Kaufmann, San Francisco.

[32] Quinlan, J. R. (1986). *Induction of Decision Trees*. Mach. Learn. 1, 1 (Mar. 1986), 81-106. DOI= http://dx.doi.org/10.1023/A:1022643204877

[33] Quinlan, J. R. (1987). *Simplifying decision trees, International Journal of Man-Machine Studies*, v.27 n.3, p.221-234, September.

[34] Quinlan, J. R. See5.0 (http://www.rulequest.com) viewed on May 2010.

[35] Savasere, A., Omiecinski, E. and Navathe, S.. (1995). *An efficient algorithm for mining association rules in large databases. In Proceeding of the 21st conference on VLDB.*

[36] Thabtah, F., Cowling, P. and Peng, Y. (2004). *Comparison of Classification techniques for a personnel scheduling problem. In Proceeding of the 2004 International Business Information Management Conference, Amman, July 2004*.

[37] 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 (ICDM'04).

[38] Witten, I. and Frank, E. (2000). *Data mining: practical machine learning tools and techniques with Java implementations*, Morgan Kaufmann, San Francisco.

[39] Yin, X., and Han, J. (2003). CPAR: Classification based on predictive association rule. In *SDM 2003*, San Francisco, CA.

[40] Zaki, M. J., Parthasarathy, S., Ogihara, M., and Li, W. (1997). *New algorithms for fast discovery of association rules*. *3rd KDD Conference*, pp. 283-286, August 1997.

[41] CBA. data mining tool. Downloading page. http://www.comp.nus.edu.sg/~dm2/p_download.html. Viewed on February 2010.

[42] Weka. Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/weka. Viewed on February 2010.