International Journal of Innovative Computing, Information and Control
Volume **7**, Number **12**, December **2011**
 ISSN 1349-4198
pp. 6815–6826

HYBRID ASSOCIATIVE CLASSIFICATION ALGORITHM USING ANT COLONY OPTIMIZATION

Waseem Shahzad and Abdul Rauf Baig

Department of Computer Science
National University of Computer and Emerging Sciences
Sector H-11/4, Islamabad, Pakistan
{ waseem.shahzad; rauf.baig }@nu.edu.pk

Received March 2010; revised July 2010

Abstract. *Classification rule discovery and association rules mining are two important data mining tasks. Association rules mining discovers all those rules from the training set that satisfies minimum support and confidence threshold while classification rule mining discovers a set of rules for predicting the class of unseen data. In this paper, we proposed a hybrid classification algorithm called ACO-AC, combining the idea of association rules mining and supervised classification using ant colony optimization. It is a class based association rules mining. The proposed technique integrates the classification with the association rule mining to discover high quality rules for improving the performance of classifier. Ant colony optimization is used to mine only the more appropriate subset of class association rules instead of exhaustively searching for all possible rules. First, strong association rules are discovered based on confidence and support and then, these rules are used to classify the unseen data. In proposed approach, we can mine association rules of each class parallel in distributed manner. We compared the proposed approach with eight other state of the art classification algorithms on twenty six data sets. Experiments results show that the hybrid classifier is more accurate and achieves higher accuracy rates when compared with other classification techniques.*
Keywords: Classification, Association rules mining, Ant colony optimization, Associative classification

1. **Introduction.** Almost all organizations collect a large number of data. There is potential business intelligence hidden in this data. The data mining has attracted a great deal of attention due to the wide availability of this huge number of data. The major question is how to turn these data into useful information and knowledge. The field of data mining answers this question. There are different data mining tasks including supervised classification, association rules mining or market basket analysis, unsupervised clustering, web data mining and regression. One technique of the data mining is classification. The goal of the classification is to build a set of models on the training data that can correctly predict the class of test objects. The input of these models is a set of objects (training data) containing the classes in which these objects belong to a set of variables describing different characteristics of the objects. There are several problems from a wide range of domains which can be casted into classification problems [1].

The aim of association rules mining is to find out the strong association between items in a given data base. Association rules that predict only a class attribute are called class association rules. Associative classification takes advantage of association rules mining for finding interesting relationship among items in the data set.

Ant colonies are distributed systems, and in spite of the simplicity of their individuals, they present a highly structured social organization. As a result of this organization, ant colonies can accomplish complex tasks that exceed the individual capabilities of a single ant. Examples are foraging, division of labor and cooperative transport. In all these examples, ants coordinate their activities via stigmergy. It is a form of indirect communication between ants by modifications of the environment [1].

In this paper, we proposed a new associative classification algorithm using ant colony optimization algorithm. The proposed approach used its evolutionary capability to efficiently find the more interesting subset of association rules. It does not exhaustively search for all possible association rules as conventional ARM approaches do. In each generation of the algorithm, a number of rules that satisfy minimum support and confidence threshold are selected for classifier.

The rest of the paper is organized as follows: in Section 2, we present the related work; Section 3 describes the ant colony optimization in detail; Section 4 describes the association rules mining and associative classification; Section 5 describes the proposed ACO-AC approach; Section 6 presents the experimentation results; and finally, Section 7 concludes the work and also gives some future directions.

2. **Related Work.** C4.5 is a decision tree classifier proposed by J. R. Quinlan [2]. It is a top down greedy algorithm. The aim of the algorithm is to build a tree that best fit the training data. The tree starts with a single node, if the all sample are of the same class, then the node become a leaf node and is labeled with the class. It used information gain as a heuristic for choosing the attributes that will best separate the sample into individual classes. The attribute with highest information gain is chosen. The selected attribute will become a test node. A branch is created for each value of the test attribute [3].

CN2 is a sequential covering algorithm that learns one rule at a time proposed by P. Clark et al. [4]. It finds a bet rule from the training data and then removes the training samples correctly classify by that rule. This process continues until there is no training sample in the training set. It used beam search to evaluate the promise of each node.

Ripper is a state of the art rule learner algorithm proposed by W. W. Cohen, for data classification [5]. It is a sequential learning algorithm. It builds rules until new rule results in too large error rate or training set is empty. An extra stopping condition is introduced that depends on the description length of the examples and rule set this is called minimum description length formula. It divides the training data randomly into growing set (2/3) and pruning set (1/3).

Ant Miner is a classification algorithm proposed by Parpinelli et al. [6]. An ant constructs a rule. It starts with an empty rule and incrementally constructs a rule by adding one item at a time. The selection of a item to be added is probabilistic and based on two factors: a heuristic quality of the item and the amount of pheromone deposited on it by the previous ants. The authors use the information gain as the heuristic value of an item until it is unable to continue its rule construction process. After the antecedent part of a rule has been constructed, the consequent of the rule is assigned by a majority vote of the training samples covered by the rule.

The extensions of the Ant Miner algorithm are proposed by Liu et al. in Ant Miner2 [7] and Ant Miner3 [8,9]. Ant Miner2 used same algorithm as original Ant Miner version. In Ant Miner2, the authors used density estimation as a heuristic function instead of information gain used by Ant Miner. They show that this simpler heuristic value does the same job as the complex one and hence Ant Miner2 is computationally less expensive than the original Ant Miner but has comparable performance. Another variant of Ant-Miner called Ant-Miner+ is proposed by Martens et al. [10]. Only the iteration-best ant is allowed to update the pheromone, the range of the pheromone trail is limited within an interval, class label of a rule is chosen prior to the construction of the rule and a different rule quality measure is used.

The first associative classification algorithm was proposed by B. Liu et al. [11]. It has three main steps rule discovery, rule selection and classification. Rule discovery process mines all rules from training data set where consequent of the rule is a class label. These rules are called class association rules. Rule selection process selects the subset of rules from all discovered rules on the basis of their predictive accuracy to make a classifier. They used confidence measure for selecting rules [12].

Another class based association rules mining algorithm called classification based on multiple class association Rules is proposed by W. Li et al. [13]. They used multiple rules for classifying an unseen data sample. For classify the test sample, it collects a small set of high confidence rules that matches with test sample and analyze correlation among those rules to assign the class label. They also used a tree structure for storing rules to improve the efficiency of rule retrieving process for classification purpose.

3. **Ant Colony Optimization.** Ant colonies or social insect societies are distributed systems. In spite of the simplicity of the behaviour of an individual, they present a highly structured social organization. As a result of this organization, ant colonies can achieve difficult tasks that exceed the individual capabilities of a single ant [1]. Ant colony optimization is a branch of swarm intelligence and a population based heuristic algorithm inspired by the behavior of natural ants. The first ACO algorithm was proposed by Marco Dorigo in 1992. ACO algorithm was developed by modeling some aspects of the food foraging behavior of ants. Ants pass the information on the trail they are following by placing a chemical substance called pheromone in the environment. Other ants that arrive in the journey are more likely to take the path with higher concentration of pheromone then the paths with lower concentrations of pheromones.

This phenomenon is modeled in the ACO algorithm. An artificial ant constructs a solution to the problem by adding solution components one by one to form a complete solution. When a solution is constructed its quality is determined and the components of the solution are assigned pheromone concentrations proportional to that quality. Subsequently other ants construct their solutions and guided by the pheromone concentrations in their search for components to be added in their solutions. Those components that have higher pheromone concentrations are identified as contributing to a good solution and repeatedly appear in the solutions. ACO has been successfully applied to solve many optimization problems such as job scheduling, subset problems, quadratic assignment and network routing.

4. **Associative Rules Mining and Associative Classification.** Association rules mining (ARM) is an important data mining problem. It finds strong and interesting relationships among large sets of data items. A typical example of ARM is market basket analysis, in which each record contains a list of items purchased by a customer. We are interested to find out the set of items that are frequently purchased together. It involves searching for interesting habits of customers. In ARM two factors are used to measure the importance of a rule, one is called support of the rule which is the ratio (or percentage) of transactions in which an item-set appears with respect to total number of transactions. Second factor is confidence, which is the ratio (or percentage) of the number of transactions that include all items in the consequent as well as the antecedent to the number of transactions that include all items in the antecedent. The support and confidence of a rule $X \Rightarrow Y$ are calculated according to Equations (1) and (2).

$$Support(X \Rightarrow Y = P(XUY)) \tag{1}$$

$$Confidence(X \Rightarrow Y = P(X|Y)) \quad (2)$$

where $P(XUY)$ means percentage of transaction contains X and Y together and $P(X|Y)$ is the probability of Y given X, means percentage of transaction that contains X also contain Y.

Associative classification is a specific kind of association rules mining, in which we are interested for finding class based association rules, in which consequent of the rule is always a class label. This is the problem tackled in the current work. Associative classification takes advantage of association rules mining for finding interesting relationship among items in the data set. There are different challenges of associative classification. First problem is rule generation is based on frequent item-set mining process and for large databases it takes a lot of time due to the large search space of items. Secondly, associative classification generates more rules. There may be redundant rules included in the classifier which increases the time cost when classifying objects. We have used redundant rule pruning procedure in our approach to tackle that problem. Second problem is rule generation which is based on frequent item-set mining process, for large data bases it takes a lot of time due to the large search space of items. We used ant colony optimization approach that mines only a subset of rules that cover the whole data set.

5. **Proposed Technique.** In this section, we will discuss our proposed ACO-AC approach in detail.

5.1. **General description.** The proposed approach finds a set of association rules from a training set to form a classifier. Each rule is in the form:

IF (item1 AND item2 AND) THEN class

Each item is an attribute-value pair. An example of item is "weather = cold". The proposed algorithm is shown in Figure 1. The algorithm for searching for the rules is ACO based. The search space is defined in the form of a graph, where each node of the graph represents a possible value of an attribute. Rules are discovered for each class separately. A temporary set of rules is discovered during each generation of the algorithm and inserted in a set of rules reserved for the selected class label. This process continues until coverage of the set of rules of selected class is greater than or equal to a minimum coverage threshold specified by the user.

At the start of the algorithm, discovered rule set is empty and user defined parameters are initialized that include minimum support, minimum confidence, minimum coverage and number of ants used by the algorithm. As we mine the association rules of each class separately, therefore, the first step is to select a class from the set of remaining classes. The pheromone values and heuristic values on links between items (attribute-value pairs) are initialized. The pheromone values on incoming links to all those items are set to zero that do not satisfy the minimum support threshold so that ants are not able to choose these items. The generation count "g" is set to 1. Generation count controls how many maximum numbers of items can be added by an ant in antecedent part of rule which it is constructing. For example when g = 2 an ant can add a maximum of two items in its rule antecedent part.

The algorithm discovers association rules for a class based on support and confidence measures. In DO WHILE loop, each ant constructs a rule. When all ants have constructed their rules, then support and confidence of each rule is calculated and those rules are selected which meet minimum support and confidence threshold. Then, we sort all selected rules in decreasing order on the basis of confidence and then on the basis of support before trying to insert them in a rule list of the selected class. The sorted rules are inserted one by one in the rule list of the selected class, according to a criterion described below, until

```
Discovered_RuleList = {};                    /* initialize the rule list with empty set */
TrainingSet = {all training samples};
Initialize min_support, min_confidence, min_coverege, /* minimum threshold for support confidence
                                                         and coverge */
Initialize No_ants;        */ initialize the maximum number of ants */
FOR EACH CLASS C IN THE TRAINING SET
        Rule_Set_Class = {}; /* initialize the rule set of the selected class with empty set */
        Initialize pheromone value of all trails;
        Initialize the heuristic function;
        Calculate the support of all 1-itemset (item => C) of the training set;
        IF(support(item) < min_support)
                Set the pheromone value 0 of all those items;
        END IF
        g = 1; /* generation count */
        WHILE(g != no_attributes && coverege < min_coverege)
                t = 1;    /* counter for ants */
                Set the class C as consequent of Ant_t;
                DO
                        Ant_t incrementally construct class based association rule by adding one item
                        at a time and it can only add maximum g no of items in its rule;
                        t = t + 1;
                WHILE(t <= no_ants);

                        For each rule constructed by all ants
                                IF(support(Rule) >= min_support)
                                        Calculate the confidence of the rule;
                                        IF(confidence(Rule) >= min_confidence)
                                                Insert the rule in the list of Rule_Set_Class;
                                        END IF
                                END IF
                        END FOR
                        Sort all the rules in Rule_Set_Class according to confidence and support;
                        Insert the rule one by one from Rule_Set_Class into Discovered_RuleList
                until coverage of Rule_Set_Class is less then min_coverage threshold;
        END WHILE
        Update pheromones;
        g = g + 1;         /* increment generation count */
END FOR
Pruning discovered rule set;
Output: Final classifier;
```

FIGURE 1. Proposed ACO-AC technique

coverage of the rule set is greater than or equal to a minimum coverage threshold specified by user. If minimum coverage criterion is not met then pheromone values are updated and a new generation starts. When minimum coverage criterion is met then the WHILE loop is exited and the rules of the selected class are copied in a final discovered rule list. Then rules are built for another remaining class and this process continues until there is no more remaining class left. When the rules of all classes have been built, then rule set pruning procedure tries to remove redundant rules from the discovered rule set and the remaining set is the final classifier.

There are different advantages of proposed approach. It does not generate all possible subsets of the association rules, which are computationally very expansive. It generates only most valuable association rules by using ant colony optimization. Another problem of the associative classification is that it finds large number of rules and most of them are redundant, we used a redundant rule pruning procedure to remove the redundant rules, which speedup the classification process. Association rules of each class can be discovered parallel because there is no dependency between association rules of each class.

5.2. **Pheromone initialization.** The pheromone values on all edges are initialized before the start of WHILE loop for each new class. The pheromone values on the edges between all items are initialized with the same amount of pheromone. The initial pheromone is:

$$\tau_{ij}(t=1) = \frac{1}{\sum_{i=1}^{a} b_i} \tag{3}$$

where a is the total number of attributes in training set excluding the class attribute and b_i is the number of possible values in the domain of an attribute a_i. The pheromone values of all those items are set to zero which do not satisfy a minimum support threshold.

5.3. **Selection of an item.** An ant incrementally adds an item in the antecedent part of the rule that it is constructing. When an item (i.e. an attribute-value pair) has been included in the rule then no other value of that attribute can be considered. The probability of selection of an item for current partial rule is given by the Equation (4):

$$P_{ij}(t) = \frac{\tau_{ij}(t)\eta_{ij}(g)}{\sum_{i=1}^{a} x_i \sum_{j=1}^{b_i}[\tau_{ij}(t)\eta_{ij}(g)]} \tag{4}$$

where $\tau_{ij}(g)$ is the amount of pheromone associated between $item_i$ and $item_j$ in current generation. Furthermore, $\eta_{ij}(c)$ is the value of the heuristic function on the link between $item_i$ and $item_j$ for the current selected class. The total number of attributes in training data set is a, and x_i is a binary variable that is set to 1 if the attribute A_i was not used by current ant and otherwise set to 0, and b_i is the number of possible values in the domain of attribute A_i.

5.4. **Heuristic function.** The heuristic value of an item indicates the quality of that item for the rule. We use a correlation based heuristic function that calculates correlation of next possible items with the last item (attribute-value pair) chosen by current ant in order to guide the selection of next item. The heuristic function is:

$$\eta_{ij} = \frac{|item_i, item_j, class_k|}{|item_i, class_k|} . \frac{|item_j, class_k|}{|item_j|} \tag{5}$$

The most recently chosen item is $item_i$ and $item_j$ is the item being considered for adding in the rule. $|item_i, item_j, class_k|$ is the number of uncovered training samples having $item_i$, and $item_j$ with class label k for which ants are constructing rules. This value is divided by the number of uncovered training samples that have $item_i$ with $class_k$ to find the correlation between the items $item_i$ and $item_j$.

The other component of the heuristic function indicates the oveall importance of $item_j$ in determining the $class_k$. The factor $|item_j, class_k|$ is the number of uncovered training samples having $item_j$ with $class_k$ and is divided by the factor $|item_j|$ is the number of uncovered training samples having $item_j$.

We used Laplace-corrected confidence for calculating the heuristic value of the first item of the rule antecedent and is given in Equation (6):

$$\eta_j = \frac{|item_j, class_k + 1|}{|item_j| + No\_classes} \tag{6}$$

where, $No\_classes$ is the total number of classes present in the data set.

5.5. **Rule construction stoppage.** An ant continues to add items in the rule in every generation. The rule construction process can stop in two cases, one if value of generation counter is equal to total number of attributes present in the data set except class attribute that means all attributes have tested and second if in any generation if coverage of the rule set of that particular class reached minimum coverage threshold.

5.6. **Quality of a rule and pheromone update.** The quality of a rule is calculated on the basis of confidence of the rule which is calculated as:

$$Q = \frac{TP}{Covered} \tag{7}$$

In which covered is the number training samples that matches with the rule antecedent part constructing by an ant and TP is the number of training samples whose antecedent is same as the antecedent of the rule of ant and whose consequent is also same as consequent of the rule of ant.

The pheromone values are updated after each generation so that in next generation ants can make use of this information in their search. The amount of pheromone on links between items occurring in those rules which satisfy minimum support threshold but whose confidence was below the minimum required confidence (and hence, they were removed from the temporary rule set) are updated according to the Equation (8):

$$\tau_{ij}(g+1) = (1-\rho)\tau_{ij}(g) + \left(1 - \frac{1}{1+Q}\right)\tau_{ij}(g) \tag{8}$$

where $\tau_{ij}(g)$ is the pheromone value between $item_i$ and $item_j$ in current generation, ρ represents the pheromone evaporation rate and Q is the quality of the rule constructed by an ant. The pheromones of these rules are increased so that in next generation ants can explore more search space instead of searching around those rules which are already inserted in the discovered rule set. This pheromone strategy increases the diversity of the search by focusing on new unexplored areas of search space. Each pheromone value is normalized by dividing it by the summation of all pheromone values of its competing items.

5.7. **Rule selection process.** When all ants have built their rules during a generation, these rules are placed in a temporary set. These rules are checked for minimum support and confidence criteria and those which do not fulfill them are removed. The next step is to insert these rules in the rule set reserved for the discovered rules of the current class. A rule is moved from the temporary rule set to the rule set of the current class only if it is found to enhance the quality of the later set. For this purpose the top rule from the temporary rule set, called R1, is removed. This rule R1 is compared, one by one, with all the rules already present in the discovered rule set of the selected class. The comparison continues until a rule from the discovered rule set satisfies a criterion described below, or until there are no more rules left in the discovered rule set with which R1 can be compared. In the later case, when no rules in the discovered rule set are able to fulfill the criterion, R1 is inserted into the discovered rule set. If a rule in the discovered rule set fulfills the criterion then the rule R1 is rejected and further comparison of R1 is stopped. The criterion is as follows. Let the compared rule of discovered rule set be called R2. If R2 is more general than R1 and confidence of R2 is higher than or equal to R1 then R2 satisfies the criterion to reject the inclusion of R1. If R2 is exactly the same as R1 then also the criterion is satisfied.

The logic of this criterion is that since R2 is already in the rule set any data sample that matches with R1 is also matched with R2 and since we assign the class label of highest confidence rule therefore transaction will always be classified by R2 and R1 will not increase the coverage of rule set.

5.8. **Discovered rule set and pruning of set.** When the coverage of discovered rule set of the selected class reaches a coverage threshold, then we stop the rule discovery process for that class. This process is repeated for all classes. A final discovered rule list contains discovered rules of all classes.

The discovered rule list may contain a large number of rules and there may be redundant rules. Redundant rules are those rules which do not fire for any single training sample. We remove these rules from the final list. The discovered rule list of rules is first sorted on the basis of confidence. Then, it is applied to classify the samples of the training set. For each sample, the rules in the discovered rule list are tested one by one in order of their sorting. If a rule fires for the test sample then the rules below it are not tested. The rule pruning process flags all those rules which are fired for at least one training sample. In this way, it discovers those rules which are never used. All such rules are deleted from the rule set. The remaining rule set becomes the final classifier and used to predict unseen test cases. This pruning process increases the comprehensibility of classifier because a small number of rules can be easily understood by a domain expert. It also makes the classification process fast because for classifying a test case we check each rule one by one.

A new test case unseen during training is assigned the class label of the rule that covers the test sample and also has the highest confidence among any other rules covering it.

6. **Experiments and Analysis.** We have implemented the proposed algorithm in MatLab 7.0. We conduct experiments on a machine that has 1.75GHZ dual processors with 1GB RAM. We compare the results of proposed approach with other state of the art, well known classification algorithms which are AntMiner, AntMinerC, C4.5 a decision tree builder, Ripper, SVM (Support Vector Machine), Logistic Regression, K-nearest Neighbour and Naïve Bayes. The performance measures for comparison are predictive accuracy, number of rules, and number of terms per rule. The experiments are performed using a ten-fold cross validation procedure.

We use twenty six data sets of UCI repository [25] for comparing different techniques. The data sets sorted on the basis of number of classes are shown in Table 1. These data sets include binary and multi-class problems. They also have diversity in terms of number of attributes, number of transactions, number of classes and types of the attributes. As our proposed approach works only for categorical attribute, therefore, we discretize the continuous attributes in a preprocessing step by using the unsupervised discretization filter.

The values of user defined parameters are given in Table 2. The parameters are: number of ants used, evaporation rate, the value of alpha and beta parameters that indicate the relative importance of pheromone and heuristic, minimum support, confidence and coverage used in the algorithm.

Our performance metrics are average predictive accuracy that indicates the predictive power of classifier, number of rules (#R) and number of terms/rule (#T/R).

Predictive Accuracy: The predictive accuracy is defined as the ability of the classification model to correctly classify the unseen data. It is reported as the average after performing tenfold cross validation.

Number of Rules: This is the average of number of rules in the rule sets after performing tenfold cross validation.

Number of Terms per Rule: This is the average of items (conditions) per rule in all the rule sets obtained by tenfold cross validation.

The best performance is shown in bold. The experiment results indicate that the ACO-AS achieves significantly higher accuracy rates then the compared algorithms on almost all the datasets. For example on breast-cancer data set the proposed approach has 99.29 average accuracy rates. However, the numbers of rules are larger than the other rule based classifier. The numbers of terms per rule are all higher than the Ant-Miner but less then C4.5, Ant-Miner-C and Ripper. These experiment results indicate for constructing

Table 1. Data sets used in the experiment. The data sets are sorted on the basis of attributes, instances and classes.

Data Sets	No.Attributes	No.Samples	No.Classes
Haberman	3	307	2
Transfusion	4	748	2
Mammographic-Mass	5	961	2
Pima Indian Diabetes	8	768	2
WBC	9	683	2
Tic-tac-toe	9	958	2
Heart	13	270	2
Credit (Australia)	15	690	2
Congress House Votes	17	435	2
Credit (Germany)	19	1000	2
Hepatitis	19	155	2
SPECT (Heart)	22	267	2
WDBC	31	569	2
Ionosphere	34	351	2
Iris	4	150	3
Balance-scale	4	625	3
TAE	6	151	3
Hayes Roth	6	132	3
Wine	13	178	3
Car	6	1728	4
Vehicle	18	282	4
Dermatology	33	366	6
Glass	9	214	7
Zoo	16	282	7
Image_Seg	19	210	7
Ecolli	7	336	8

Table 2. Parameters used in experiments

Parameter	Value
Number of Ants	1000
Evaporation Rate	0.15
Alpha	1
Beta	1
Minimum Support	0.01
Minimum Confidence	0.50
Min Coverage	0.98

classifier if we used association rules mining approach for discovering classification rules finds more accurate and compact rules from training set.

The reason of achieving higher accuracy rate is that when we used association rules mining approach for discovering classification rules, it finds more interesting rules that increased the predictive power of the classifier. Ant colony optimization better search the whole search space in order to find valuable patterns from whole training set.

TABLE 3. Average predictive accuracies with standard deviations obtained after tenfold cross validation

Data Sets	ACO-AC	Ant-Miner-C	Ant-Miner	C4.5	KNN	Log Reg	Naïve Bayes	Ripper	SVM
BC-W	99.29 ± 1.01	97.85 ± 1.69	94.64 ± 2.74	94.84 ± 2.62	96.42 ± 1.54	96.56 ± 1.21	96.13 ± 1.19	95.57 ± 2.17	96.70 ± 0.69
Wine	100.0 ± 0.0	99.44 ± 1.76	90.0 ± 9.22	96.60 ± 3.93	96.08 ± 4.59	96.60 ± 4.03	98.30 ± 2.74	94.90 ± 5.54	98.30 ± 2.74
Credit(Aus)	98.84 ± 1.14	87.54 ± 3.21	86.09 ± 4.69	81.99 ± 7.78	82.44 ± 7.31	85.77 ± 4.75	79.37 ± 4.57	86.07 ± 2.27	85.17 ± 2.06
Credit(Ger)	97.80 ± 1.48	72.46 ± 5.13	71.62 ± 2.71	70.73 ± 6.71	74.43 ± 7.87	75.82 ± 4.24	74.87 ± 5.96	70.56 ± 5.96	75.11 ± 3.63
Car	91.43 ± 3.31	98.03 ± 1.17	82.38 ± 2.42	96.0 ± 2.13	93.75 ± 1.87	93.22 ± 2.10	86.04 ± 2.32	89.17 ± 2.52	93.74 ± 2.65
Tic-tac-toe	99.06 ± 0.92	100 ± 0.0	74.95 ± 4.26	94.03 ± 2.44	98.75 ± 0.66	98.23 ± 0.50	70.09 ± 5.78	97.57 ± 1.44	98.33 ± 0.53
Iris	98.67 ± 2.81	98.0 ± 4.50	95.33 ± 4.50	94.0 ± 6.63	95.33 ± 4.50	97.33 ± 5.62	95.33 ± 3.22	94.76 ± 5.26	96.67 ± 3.52
Bal-scale	91.53 ± 2.11	87.49 ± 6.34	75.32 ± 8.86	83.02 ± 3.24	89.26 ± 1.59	88.30 ± 2.69	91.04 ± 2.55	80.93 ± 3.35	87.98 ± 1.80
TAE	86.75 ± 10.89	81.38 ± 11.72	50.67 ± 6.11	51.33 ± 9.45	64.67 ± 7.73	53.33 ± 11.33	57.33 ± 10.91	44.67 ± 10.35	58.67 ± 10.98
Glass	91.60 ± 6.46	82.27 ± 6.67	53.33 ± 4.38	68.90 ± 8.98	70.95 ± 5.83	63.65 ± 6.72	51.69 ± 8.31	70.48 ± 8.19	57.70 ± 8.10
Heart	99.25 ± 1.56	80.74 ± 9.37	80.74 ± 4.94	78.43 ± 6.26	80.71 ± 6.17	77.0 ± 5.05	85.19 ± 8.27	73.59 ± 9.57	80.32 ± 6.25
Hepatitis	98.0 ± 4.50	87.17 ± 7.81	80.67 ± 8.67	68.25 ± 11.63	67.62 ± 9.06	64.25 ± 8.87	74.79 ± 6.90	73.46 ± 8.21	75.37 ± 8.62
Zoo	97.0 ± 4.83	96.0 ± 5.16	81.36 ± 11.30	94.0 ± 9.17	97.0 ± 6.75	97.0 ± 4.83	97.0 ± 6.75	90.0 ± 11.55	95.0 ± 7.07
Haberman	86.31 ± 4.57	83.05 ± 7.80	71.99 ± 7.57	73.88 ± 4.66	70.11 ± 11.95	73.08 ± 3.99	74.05 ± 4.76	72.47 ± 6.74	73.44 ± 0.97
Ecolli	97.80 ± 1.48	83.64 ± 6.11	47.52 ± 11.32	82.99 ± 7.72	80.86 ± 6.45	86.31 ± 5.38	85.66 ± 2.83	79.18 ± 2.22	82.69 ± 5.04
Vehicles	85.80 ± 6.73	85.91 ± 5.62	56.79 ± 9.56	65.86 ± 6.76	67.62 ± 5.63	77.18 ± 9.35	42.79 ± 2.54	66.52 ± 11.86	66.57 ± 7.40
Mam-Mass	97.92 ± 2.40	78.67 ± 4.65	78.25 ± 3.48	82.44 ± 4.56	78.07 ± 5.99	82.92 ± 3.91	82.69 ± 3.82	83.52 ± 4.52	80.32 ± 4.37
Ind-Diabetes	96.88 ± 4.47	80.26 ± 6.19	74.63 ± 6.65	72.11 ± 6.96	69.75 ± 7.13	77.98 ± 5.91	75.49 ± 4.50	77.96 ± 7.47	77.06 ± 4.09
Dermatology	97.28 ± 3.83	94.27 ± 4.51	58.72 ± 7.36	95.07 ± 2.80	95.62 ± 3.22	96.71 ± 2.83	96.98 ± 2.75	93.96 ± 3.63	97.54 ± 2.99
Ionosphere	98.29 ± 2.41	89.71 ± 7.56	68.0 ± 11.09	89.98 ± 5.25	86.0 ± 6.38	86.57 ± 7.63	82.75 ± 4.94	90.29 ± 6.90	87.14 ± 5.75
WDBC	98.95 ± 1.89	87.33 ± 5.46	85.26 ± 4.22	93.31 ± 2.72	95.06 ± 2.88	95.24 ± 2.07	93.48 ± 4.49	94.03 ± 3.72	97.72 ± 1.45
Image-Seg	91.43 ± 7.71	98.10 ± 2.61	70.48 ± 10.95	88.82 ± 8.04	86.57 ± 7.99	85.71 ± 9.28	78.48 ± 5.29	83.74 ± 7.72	88.52 ± 5.70
Spec-Heart	98.49 ± 1.94	87.41 ± 4.97	75.38 ± 5.50	80.03 ± 51.85	81.20 ± 16.77	81.94 ± 15.92	76.87 ± 12.72	79.29 ± 19.43	87.20 ± 11.48
Transfusion	85.71 ± 2.74	79.57 ± 3.04	77.30 ± 6.37	77.71 ± 12.32	71.01 ± 14.98	77.17 ± 14.49	72.37 ± 17.22	76.11 ± 14.54	76.24 ± 15.33
Hayes Roth	95.66 ± 11.29	91.65 ± 8.37	75.05 ± 10.62	80.93 ± 8.24	70.93 ± 9.59	53.52 ± 11.18	77.20 ± 11.08	78.57 ± 14.47	58.13 ± 15.33
House-Votes	99.30 ± 1.11	95.86 ± 3.71	94.54 ± 2.27	95.31 ± 2.57	92.28 ± 3.86	95.14 ± 3.0	90.54 ± 3.59	95.66 ± 2.75	96.09 ± 3.06

Table 4. Average number of rules per discovered rule set and average number of terms per rule. The results are obtained using tenfold cross validation.

Datasets	#R/Rule Set					#T/R				
	ACO-AC	AMC	AM	C4.5	Ripper	ACO-AC	AMC	AM	C4.5	Ripper
BC-W	33.90	20.40	11.0	10.50	5.10	1.0	1.42	1.02	2.32	1.79
Wine	12.0	6.0	5.50	5.30	3.90	1.0	1.25	1.04	1.41	1.62
Credit(Aus)	26.80	5.50	3.90	74.80	4.60	1.56	1.57	1.0	3.22	1.81
Credit(Ger)	81.80	13.50	8.50	73.60	4.20	2.16	1.88	1.13	3.21	2.36
Car	32.10	58.0	11.40	80.26	41.10	2.68	2.50	1.03	2.59	4.01
Tic-tac-toe	30.90	12.30	6.60	38.60	10.30	1.82	2.74	1.09	2.64	2.82
Iris	18.90	12.80	9.20	5.50	3.90	1.19	1.05	1.0	1.22	1.03
Bal-scale	13.50	108.7	17.70	40.10	11.10	1.0	2.43	1.0	2.85	2.91
TAE	43.30	48.70	20.90	18.30	3.90	1.82	1.44	1.0	2.69	1.64
Glass	58.60	42.20	15.50	15.40	7.20	1.92	1.93	1.01	2.83	2.33
Heart	23.10	13.10	5.60	12.60	5.60	1.0	1.79	1.08	1.73	1.86
Hepatitis	25.80	11.30	3.90	11.60	4.60	1.07	2.41	1.11	1.70	1.0
Zoo	11.80	7.0	5.10	7.60	6.80	1.05	1.59	1.11	1.60	1.67
Haberman	35.10	58.20	20.70	3.40	2.30	1.32	1.56	1.0	1.58	2.0
Ecolli	81.80	57.60	8.60	14.0	9.10	2.16	1.75	1.01	2.84	2.98
Vehicles	78.50	43.0	14.20	20.10	8.40	1.87	1.82	1.02	3.13	1.92
Mam-Mass	31.30	20.30	15.90	8.90	4.30	1.39	1.82	1.0	2.47	2.15
Ind-Diabetes	86.40	45.60	15.30	7.80	3.90	1.84	1.93	1.0	2.18	2.59
Dermatology	39.10	20.0	10.40	9.30	8.70	1.66	2.69	1.07	2.23	2.99
Ionosphere	50.60	6.67	4.20	9.30	5.80	1.0	1.15	1.0	2.54	2.33
WDBC	45.60	15.40	8.40	7.20	4.70	1.0	1.67	1.0	1.75	2.60
Image-Seg	58.50	26.60	16.20	10.60	9.80	1.34	1.41	1.03	1.99	2.83
Spec-Heart	19.0	25.40	5.60	12.50	2.40	1.66	7.17	1.28	3.02	3.23
Transfusion	17.10	24.80	10.10	4.20	3.10	1.36	1.45	1.0	1.35	2.07
Hayes Roth	18.90	19.80	8.0	11.70	7.20	1.46	1.76	1.02	2.56	2.40
House-Votes	5.80	7.30	3.0	7.40	2.60	1.0	1.95	1.0	1.84	2.05

7. **Conclusion and Future Work.** In this paper, we proposed an ACO based associative classification algorithm by combining the two important data mining paradigms, classification and association rules mining. It is a supervised learning approach for discovering association rules. Ant colony optimization is used to find the most suitable set of association rules. ACO search only a subset of association rules to form an accurate classifier instead of massively searching all possible association rules from data set hence proposed approach avoids exhaustive search in rule discovery process, used its evolutionary capability to do that. It has the ability of efficiently dealing with complex search space. We compared our ACO-AC approach with eight other popular classification techniques on a large number of data sets. The experiment results indicate that proposed ACO-AC approach performs significantly better than the state of the art classification approaches.

In future, we will try to thoroughly investigate the various parameters used in the algorithm specially support, confidence and coverage. This approach can be extended to find out the general association rules instead of class association rules.

REFERENCES

[1] A. G. Ramos, Swarm intelligence in data mining, in *Studies in Computational Intelligence, vol.34*, A. Abraham, C. Grosan and V. Ramos (eds.), Springer, 2006.

[2] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

[3] J. R. Quinlan, Generating production rules from decision trees, *Proc. of Int. Joint Conf. Artificial Intelligence*, San Francisco, USA, 1987.

[4] P. Clark and T. Niblett, The CN2 induction algorithm, *Machine Learning*, vol.3, no.4, pp.261-283, 1989.

[5] W. W. Cohen, Fast effective rule induction, *Proc. of the 12th International Conference on Machine Learning*, 1995.

[6] R. S. Parpinelli, H. S. Lopes and A. A. Freitas, Data mining with an ant colony optimization algorithm, *IEEE Transaction on Evolutionary Computation*, vol.6, no.4, pp.321-332, 2002.

[7] M. Dorigo and L. M. Gambardella, Ant colony system: A cooperative learning approach to the travelling salesman problem, *IEEE Transaction on Evolutionary Computation*, vol.1, no.1, 1997.

[8] B. Liu, H. A. Abbass and B. McKay, Classification rule discovery with ant colony optimization, *Proc. of IEEE/WIC Int. Conf. Intell. Agent Technol.*, pp.83-88, 2003.

[9] C.-J. Lin, C. Chen and C. Lee, Classification and medical diagnosis using wavelet-based fuzzy neural networks, *International Journal of Innovative Computing, Information and Control*, vol.4, no.3, pp.735-748, 2008.

[10] D. Martens, M. de Backer, R. Haesen, J. Vanthienen, M. Snoeck and B. Baesens, Classification with ant colony optimization, *IEEE Transactions on Evolutionary Computation*, vol.11, no.5, 2007.

[11] B. Liu, H. Hsu and Y. Ma, Integrating classification and association rule mining, *Proc. of the 4th Int. Conf. Knowledge Discovery Data Mining*, pp.80-86, 1998.

[12] B. Liu, Y. Ma and C.-K.Wong, Classification using association rules: Weaknesses and enhancements, in *Data Mining for Scientific and Engineering Applications*, R. L. Grossman, C. Kamath, P. Kegelmeyer, V. Kumar and R. R. Namburu (eds.), Berlin, Germany, Springer-Verlag, 2001.

[13] W. Li, J. Han and J. Pei, CMAR: Accurate and efficient classification based on multiple class-association rules, *Proc. of 2001 IEEE Int. Conf. Data Min.*, pp.369-376, 2001.

[14] S. Hettich and S. D. Bay, *The UCI KDD Archive*, http:// kdd.ics.uci.edu., 1996.

[15] T.-P. Hong, C.-W. Lin and Y.-L. Wu, An efficient FUFP-tree maintenance algorithm for record modification, *International Journal of Innovative Computing, Information and Control*, vol.4, no.11, pp.2875-2887, 2008.

[16] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd Edition, Morgan Kaufmann Publishers, 2006.

[17] B. Liu, H. A. Abbass and B. McKay, Classification rule discovery with ant colony optimization, *IEEE Computational Intelligence Bulletin*, vol.3, no.1, 2004.

[18] S. Swaminathen, *Rule Induction Using Ant Colony Optimization for Mixed Variable Attributes*, Master Thesis, Texas Technology University, 2006.

[19] L. Wang, X. Wang and Y. Liu, Combination of multiple real-valued nearest neighbor classifiers based on different feature subsets with fuzzy integral, *International Journal of Innovative Computing, Information and Control*, vol.4, no.2, pp.369-379, 2008.

[20] M. Dorigo, V. Maniezzo and A. Colorni, Ant system: Optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernatics, Part B*, vol.26, no.1, 1996.

[21] Y. Kusunoki and M. Inuiguchi, Rule induction via clustering decision classes, *International Journal of Innovative Computing, Information and Control*, vol.4, no.10, pp.2663-2677, 2008.

[22] J. Han, J. Pei and Y. Yin, Mining frequent patterns without candidate generation, *Proc. of 2000 ACM SIGMOD Int. Conf. Manage. Data*, pp.1-12, 2000.

[23] J. L. Koh and P. Yo, An efficient approach for mining fault-tolerant frequent patterns based on bit vector representations, *Proc. of the 10th International Conference DASFAA 2005*, Beijing, China, 2005.

[24] Y.-W. Wang, P.-C. Chang and C.-Y. Fan, Database classification by integrating a case-based reasoning and support vector machine for induction, *Journal of Circuits, Systems, and Computers*, vol.19, no.1, pp.31-44, 2010.

[25] W. Shahzad and A. R. Baig, Compatibility as a heuristic for construction of rules by artificial ants, *Journal of Circuits, Systems, and Computers*, vol.19, no.1, pp.297-306, 2010.

[26] C.-J. Chu, V. S. Tseng and T. Liang, Mining temporal rare utility itemsets in large databases using relative utility thresholds, *International Journal of Innovative Computing, Information and Control*, vol.4, no.11, pp.2775-2792, 2008.