

Application of a New m *Ant-Miner*_{PR} Algorithm in Classification Rule Mining

L. Yang^{1*}, K.S. Li¹, W.S. Zhang², Y. Wang³, Z.X. Ke¹

¹College of Mathematics and Informatics
South China Agricultural University, Guangzhou
510642, China

²Institute of Automation, Chinese Academy of Sciences
Beijing, 100190, China

³Department of Computer Science and Information Systems
Bradley University, Peoria
Illinois, 61625, USA
yanglei_s@scau.edu.cn



ABSTRACT: A new classification algorithm based on multi-ant and pheromone repulsion principles is studied and proposed in this paper to improve the prediction accuracy of the classification rule based on the traditional Ant-Miner algorithm. The proposed algorithm uses multi-ant colony construction method to reduce dependence on random initial term. The volatile coefficient of pheromone update is added to prevent premature convergence. The new state transition function based on pheromone repulsion principle is also presented to avoid falling into local optimum, whereas a rule quality formula is improved. Several aspects of the proposed algorithm are investigated by experimenting of benchmark data sets from the University of California at Irvine repository. We study the performance of our proposed approach and compare it with several commonly used classification algorithms, such as Ant-Miner and Ant-Miner with graphical user interface. Experimental results indicate that the proposed approach builds more accurate models than the compared algorithms. The high accuracy supplemented by the comprehensibility of the discovered rule sets is the main advantage of this method.

Subject Categories and Descriptors

K.2.8 [Database Applications]: Data mining; **B.2.4** [High-Speed Arithmetic]: Algorithms

General Terms: Computational Swarm Intelligence, Ant Colony Optimization, Data Mining, Classification

Keywords: Data Mining, Classification Rule, Ant Colony Algorithm, Pheromone Repulsion

Received: 9 June 2015, Revised 29 July 2015, Accepted 8 August 2015

1. Introduction

Data mining is a process of mining interesting patterns and knowledge from a large amount of data [1]. The patterns and knowledge can be used for information management, query optimization, decision support, and process control. This mining process can also be used to maintain the data themselves. Data mining is the combination of artificial intelligence, machine learning, and database. Classification is the main task of data mining [2–4]. Classification rule mining is an important part of classification method [5]. In the field of data mining, classification rule discovery is defined as a classification model constructed by a certain classification method based the characteristics of a data set, which can be used to map the data record to a given category. Thus, this data set can be used to predict data categories.

Ant colony optimization (ACO)[6,7] is a family of metaheuristics, which was initially proposed by Dorigo based on the food-foraging behavior of biological ants. The optimal solution or approximate optimal solution can be determined in a large search space through cooperation among ants. This widely used algorithm is robust, self-organizing, and parallel, and it can be integrated easily with other algorithms. The classification task of data mining encounters massive data sets, and search spaces are huge. Thus, ACO is used to classify problems, and a

famous Ant-Miner algorithm [8] was proposed by Parpinelli et al.

In this paper, a new ant colony classification mining algorithm based on the principles of multi-ants and pheromone repulsion, namely, m -Ant-Miner_{PR}, is proposed according to the classical Ant-Miner algorithm. Improvements are attained in four aspects, such as ant colony construction, state transition probabilities, pruning strategy, and pheromone update method. The proposed algorithm is implemented with Java language. Then, our experiments using six publicly available data sets show that the predictive accuracy obtained with m -Ant-Miner_{PR} algorithm implementing the proposed multi-ant and pheromone repulsion strategies is statistically and significantly higher than the predictive accuracy of other rule induction classification algorithms, such as Ant-Miner and graphical user interface (GUI) Ant-Miner [9]. Moreover, the rules discovered with m -Ant-Miner_{PR} algorithm are considerably simple.

The remainder of this paper is organized as follows. Section 2 explains the basic concepts of ACO and gives an overview of existing ACO algorithms for classification rule discovery. Section 3 describes our proposed approach. The experimental results and discussion are presented in Section 4. Section 5 concludes this paper and gives some future directions of research.

2. ACO and Classification Rule Discovery

2.1 ACO

Ant colony algorithm, which was derived from the foraging behavior of real ants, was first proposed by Dorigo et al. [10] in 1991. Ants can determine the shortest path between food sources and nests without using visual information, and they can adapt to environment changes. Research indicated that ants leave a chemical hormone (called pheromone in ant colony algorithm) in the path that they had followed, and the route information is communicated with other ants via the concentration of chemical elements. The higher the number of ants that pass through a path is, the higher the concentration of the chemical hormone on this path becomes, thereby attracting a large number of ants to choose this path. This process can be described as a positive feedback mechanism, that is, the probability of an ant to choose a path that is proportional to the number of ants that have passed over the path. Given that ants prefer to choose the path with high concentration of chemical element, all ants eventually focus on the shorter path.

The definition of ACO in the book written by Dorigo and Stützle [11] in 2004 provided a broad description. Given an optimization problem, we needed to first define the solution space of a finite set, which could be used to compile optimization problem solutions. Then, we needed to define a set of pheromone values; these values are usually called pheromone models. A pheromone model is a parametric probability model, which is a core

module of ant colony algorithm; it is used to generate solutions of a problem based on probability. The candidate solution is established after iteration based on the pheromone and heuristic information. Then, an ant colony eventually converges to the optimal solution or near-optimal solution.

2.2 ACO for Classification Rule Discovery

Ant colony algorithm is a bionic optimization algorithm inspired by the foraging behavior of ants. It is widely used in solving complex combinatorial optimization problems because of its ability to determine optimal solutions and its strong robustness. In recent years, several attempts have been made to apply ACO for the discovery of classification rules. The first ACO-based algorithm for this purpose, called Ant-Miner, was proposed by Parpinelli et al. in 2002 based on an ant colony mechanism. Classification rules were constructed by simulating the foraging behavior of ants in this algorithm. The process of each ant search is equivalent to the discovery process of rules. The process of ants in choosing a direction along the way is equivalent to the process of ants in choosing an attribute node to add to rules. In Ant-Miner, given some data, we conduct a heuristic search for finding rules governing the data. The primary objective of Ant-Miner algorithm is the incremental construction of a classification rule of the type "IF <conditions> THEN <class>" by an ant. Each term is a triple<attribute, operator, value>, where the value belongs to the domain of attribute. The operator element in the triple is a relational operator. The current version of classification algorithm copes only with categorical attributes, so that the operator element in the triple is always "=". If the properties of an instance meet the <conditions> part of rules, the rules of <class> part are used to predict the class of instances.

The basic idea of Ant-Miner [12] is as follows: At first, the list of discovered rules is empty, and the training set consists of all training cases. Each search of the algorithm completes the three basic tasks, namely rule construction, rule pruning, and pheromone updating. Then, a classification rule is found. This rule is added to the list of discovered rules, and the training cases covered correctly by this rule are removed from the training set. This process is performed iteratively, whereas the number of uncovered training cases is greater than a user-specified threshold, called maximum uncovered. Algorithm eventually obtains a set of classification rules. The heuristic function in Ant-Miner is based on ant colony mechanism and information theory. Compared with other classification algorithms, Ant-Miner is a global search classification rule resulting from the introduction of an ant colony operation mechanism. It is different from the greedy and local search of decision tree and rule reduction algorithm. Pheromones and the heuristic factor are the two aspects of the decision factor of Ant-Miner, and the effective combination of these two factors determines the operating efficiency of Ant-Miner.

Ant-Miner is the first algorithm that applied ACO to data classification problem. Studies [13–15] showed that

compared with the classical classification algorithms C4.5 [16] and CN2 [17,18], Ant-Miner algorithm simplifies the discovered rules and improves the comprehensibility and prediction accuracy of classification. In recent years, numerous researchers of ant colony classification algorithm have proposed many expansion and improvement methods for Ant-Miner. In the construction of ant colony, Meyer [9] proposed GUI Ant-Miner, which used multi-ant colony construction method. Jin et al. [19] presented multi-swarm ant colony construction method. In the state transition rule, Liu et al. [20] proposed the new state transfer rule, which extended the exploration ability for Ant-Miner state transition rules and reduced the discovered rule preferences. Wang et al. [21] simplified state transfer probability calculation method based on the research foundation of Liu. In heuristic strategy, Liu et al. [22] presented the evaluation method based on density. Martens [23] produced effective ant-based assessment of the selected class of equations. Wang et al. [21] proposed a simple and effective heuristic function for the problem of prematurity. In terms of rule pruning, Chan et al. [24] proposed a hybrid pruning approach to improve the pruning speed of Ant-Miner in dealing with large data set of attributes. Pheromone updating was discussed in [21, 22] to control the pheromone evaporation rate of old trail. Liu et al. introduced the pheromone evaporation rate for the pheromone updating formula of Ant-Miner. Wang et al. proposed the adaptive pheromone update rule method based on the method of Liu. In terms of dealing with continuous attributes, Otero et al. [25,26] presented the processing method to discretize continuous attributes in construction rules. Then, they proposed dynamic discretization method based on the minimum description length. In addition, Chan et al. [27] proposed an improved algorithm to solve multilabel classification tasks. Smaldon et al. [28] proposed an improved algorithm to generate a random classification rule.

We studied GUI Ant-Miner [9] and determined that its heuristic function value (followed Ant-Miner heuristic function) contains local information and varies with the operation of the algorithm. Therefore, GUI Ant-Miner easily falls into local optima. We referenced the ant colony construction approach of GUI Ant-Miner and proposed new state transition probabilities, rule pruning strategy, and pheromone updating method. Then, a new ant colony classification mining algorithm based on the principles of multi-ants and pheromone repulsion, namely, ${}_m\text{Ant-Miner}_{PR}$, was proposed. The primary objective of this research is to determine the improved method of the proposed algorithm in this paper, which can improve the prediction precision and stability of the algorithm and the simplicity of rules. In this manner, the performance of classification mining algorithm can be improved, and the improved algorithm can obtain advantages in massive data classification.

3. New Ant Colony Classification Mining Algorithm:

${}_m\text{Ant-Miner}_{PR}$

Extracting rules from the data set is the first step in the process of large-scale data mining, which can be used as a discrete optimization problem. The main objective is to find the symbol rule sets of the type “IF <conditions> THEN <class>”, which can accurately describe the data sets, and predicting the classification with the minimum error. This section mainly introduces the relationship of ${}_m\text{Ant-Miner}_{PR}$ rule extraction problem with the subset problem and the process of using ${}_m\text{Ant-Miner}_{PR}$ to extract accurate rule sets and to predict the class of data.

3.1 Problem Definition

Given the data set $D = \{(x_p, d_p) | p = 1, \dots, n_p\}$, where x_p represents the aggregate of input values, and d_p is the ideal classification, the goal is to extract a set of rules to predict the corresponding class of the input vector accurately. That is to say, the goal is to minimize errors.

$$f(\pi) = \frac{\sum_{p=1}^{n_p} (o_p - d_p)^2}{n_p} \quad (1)$$

where o_p is the predictive class of rule set for input vector x_p .

3.2 Problem Description

In Ant-Miner[29], if attributes $x_i (i = 1, \dots, n_x)$ are discrete values, each attribute has a finite set $dom(x_i) = \{v_{i1}, \dots, v_{im}\}$, and each attribute value pair $x_i = v_{ij}$ is expressed by describing the one node in the graph, then the figure is fully connected. Each rule has connection conditions as antecedent; each condition is an attribute value pair. Corresponding to a rule, a solution can be regarded as a condition set, i.e., $\pi = \{x_i = v_{ij} : x_i \in X, v_{ij} \in dom(x_i)\}$; X is the input attribute values sets. Thus, the process of searching the solution can be considered as a subset problem, and a subset of all the attribute value pair is selected, thereby making the data vectors covered as much as possible by the corresponding rules.

3.3 Construction of Solution

3.3.1 Ant Colony Construction Method and Pheromone Initialization

Ant-Miner draws on the idea of using single-ant, GUI Ant-Miner proposed by Meyer [9], which is based on the idea of using multi-ants. However, only an ant (ant colony optimal) is used for pheromone updating, thereby making the computational complexity of GUI Ant-Miner lower than that of the method of pheromone updating by every ant. The approach of GUI Ant-Miner can be considered as a compromise and trade-off of single ant and standard ACO ant colony. In this study, we used the idea of GUI Ant-Miner and defined the algorithm for ${}_m\text{Ant-Miner}$, which can reduce the dependence of the algorithm on random initial term. First, the algorithm was initialized, and the parameter values were required to set. The list of discovered rules was empty, and the training set consists of all the training cases. The attribute node path was established by

simulating ants in ant colony algorithm optimization model. Every node was initialized with the same amount of pheromone. Thus, when the first ant has started its search, all nodes would have the same amount of pheromone. The initial amount of pheromone deposited at each path position is inversely proportional to the number of values of all attributes and is defined by Eq. (2).

$$\tau_{ij}(t=0) = \frac{1}{\sum_{l=1}^{|X|} n_l} \quad (2)$$

where X is the total number of attributes, and n_l is the number of possible values that can be acquired by attribute x_l .

3.3.2 Improvement of State Transfer Rule

Each ant constructs a rule by starting with an empty rule, that is, $\pi^k(0) = \phi$, and by incrementally adding one condition at a time, thereby gradually constructing a single rule. The m -Ant-Miner_{PR} is a sequential covering algorithm. It discovers a rule, and the training samples correctly covered by this rule (i.e., samples that satisfy the rule antecedent and have the class predicted by the rule consequent) are removed from the training set. Subsequently, the algorithm discovers another rule through the reduced training set, and then the training set is further reduced by removing the training samples covered by the newly discovered rule. This process continues until the training set is empty or nearly empty. The probability that $x_i = v_{ij}$ is chosen to be added to the current partial rule is

$$p_{ij}^k(t) = \frac{\tau_{ij}(t)\eta_{ij}}{\sum_{x_l \in N_i^k(t)} \sum_{v_{lm} \in v(x_l)} \tau_{lm}(t)\eta_{lm}} \quad (3)$$

where $N_i^k(t)$ is the set of attributes that ant k unused, $v(x_l)$ contains all the values of the attribute x_l , and $\tau_{ij}(t)$ is the amount of pheromone associated with $term_{ij}$ at iteration t , corresponding to the amount of pheromone currently available in the position i, j of the rule being followed by the current ant. The better the quality of the rule constructed by an ant is, the higher the amount of pheromone added to the trail segments visited by the ant becomes. Therefore, as time passes, the best trail segments to be followed, i.e., the best terms (attribute-value pairs) to be added to a rule eventually obtain increasing amounts of pheromone, thereby increasing probability of being chosen. η_{ij} is the value of a problem-dependent heuristic function for $term_{ij}$. The higher the value of η_{ij} is, the more relevant for classification the $term_{ij}$ becomes, thereby increasing its probability of being chosen. The calculation method of the amount of pheromone and problem-dependent heuristic function is organized as follows.

In this paper, we proposed a new calculation method of pheromone based on pheromone repulsion, and this method was applied to the new transition probability. Pheromone contained attractive parts and repulsion parts,

and the released pheromone from different ants could be distinguished with the algorithm. The ants were attracted by their own released pheromones and excluded by pheromones released by other ants. Therefore, we defined a representation to the attractive weight of Λ_{ij}^k and a representation of the exclusive weight of Φ_{ij}^k for each attribute node $term_{ij}$ of the ant k . The attractive weight of representation is defined as follows:

$$\Lambda_{ij}^k(t) = \frac{\tau_{ij}^k(t)}{\sum_{u \in N_i^k(t)} \tau_{iu}^k(t)} \quad (4)$$

where $\tau_{ij}^k(t)$ is the pheromone of $term_{ij}$ for ant k , which is normalized with the sum of pheromones left on the other nodes by the ant k . The weight of representation of exclusion is defined as

$$\Phi_{ij}^k(t) = \frac{\Pi_{ij}^k(t)}{\sum_{u \in N_i^k(t)} \Pi_{iu}^k(t)} \quad (5)$$

where

$$\Pi_{ij}^k(t) = \sum_{m=1, \dots, n_k (m \neq k)} \tau_{ij}^m(t) \quad (6)$$

The formula represents the amount of repulsion pheromone of $term_{ij}$ for ant k , namely, the sum of pheromone quantity released by other ants. According to the preceding definition, the method for calculating the pheromone of $term_{ij}$ is

$$\tau_{ij}(t) = \frac{(\Lambda_{ij}^k(t)/\Phi_{ij}^k(t))^\alpha}{\sum_{u \in N_i^k(t)} (\Lambda_{iu}^k(t)/\Phi_{iu}^k(t))^\alpha} \quad (7)$$

The ratio of attraction and repulsion in the formula balances the relationship between exploration and exploitation. The algorithm initially tends to explore and then, in the search of the latter, to exploit. This balance is achieved by constantly increasing the weights of attraction and decreasing the weights of repulsion. At the beginning of the search, ants tend to walk along the road that has never been visited because the amount of pheromone released by other ants is more than that released by themselves, thereby aiding in exploring a new path and discovering new rules. However, the continued increase of the released pheromones of the ants themselves results in the increase of weight of attraction, thereby making the approaching ants conduct their own exploitation, which converges to the optimal solution in finding the optimal rules.

In m -Ant-Miner_{PR}, the problem-dependent heuristic function is based on the amount of information corresponding to each attribute value pair $x_i = v_{ij}$.

$$\eta_{ij} = \frac{\log_2 n_c - I(D_{ij})}{\sum_{l=1}^{|X|} \sum_{m=1}^{\eta_l} \log_2 n_c - I(D_{lm})} \quad (8)$$

where n_c is the number of category, D_{ij} is all data types of the attribute value pair $x_i = v_{ij}$, and $I(D_{ij})$ is the corresponding amount of information.

$$I(D_{ij}) = -\sum_{q=1}^{n_c} \frac{f_{ij,q}}{|D_{ij}|} \log_2 \frac{f_{ij,q}}{|D_{ij}|} \quad (9)$$

where $f_{ij,q}$ is the number of patterns in class q of D_{ij} .

3.3.3 Improved Method of Rule Pruning

Ant-Miner uses sensitivity multiplied by specificity, as quality is measured with Eq. (10). This formula includes unreasonable places. For example, when $FP + TN = 0$, the right fraction denominator is zero, and the calculation produces errors. If TN is 0, then the calculation result is 0. However, the actual situation is TP representation of the number of examples covered by the rule, that is, TP is not 0, and the quality of Q should not be 0. Thus, the formula to calculate the quality of the rules should be improved.

$$Q = \left(\frac{TP}{TP + FN}\right) \times \left(\frac{TN}{FP + TN}\right) \quad (10)$$

The quality of a rule in this paper is presented by the following formula:

$$Q = \begin{cases} 1 & (FP + FN + TN = 0) \\ \frac{TP}{TP + FN} & (FP + TN = 0) \\ \frac{TP + FP}{TP} & (TN + FN = 0) \\ \frac{TP + FN}{TP + FN} \times \frac{TN}{FP + TN} & (ELSE) \end{cases} \quad (11)$$

where TP is the number of cases covered by the rule that has the class predicted by the rule, FP is the number of cases covered by the rule that has a class different from that predicted by the rule, TN is the number of cases that are not covered by the rule and do not have the class predicted by the rule, and FN is the number of cases that are not covered by the rule but have the class predicted by the rule. The larger the value of Q is, the higher the quality of the rule becomes.

3.3.4 Improvement on the Method of Pheromone Update

In Ant-Miner, the amount of pheromone is increased because the set of terms occurring in the rule constructed by the ant is overly dependent on rule quality Q, and the amount of pheromone associated with each $term_{ij}$ that does not occur in the rule determined by the current ant decreases quickly. This mechanism can easily lead to premature convergence of the algorithm, which leads to a low prediction accuracy [30]. We introduced pheromone volatilization coefficient ρ to simulate pheromone evaporation in real ant colonies. Pheromone evaporation is implemented according to Eq. (12) for the set of terms occurring in the rule constructed by the ant, and the

amount of pheromone of an unused term is computed by dividing its current value by the total summation of pheromone for all terms based on Eq.(13). The final effect is the reduction of the normalized amount of pheromone for each unused term.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \left(1 - \frac{1}{1+Q}\right)\tau_{ij}(t) \quad (12)$$

$$\tau_{ij}(t+1) = \frac{\tau_{ij}(t)}{\sum_i^a \sum_j^b \tau_{ij}}, \forall i, j \notin Rule \quad (13)$$

The pheromone update is close to the foraging behavior of real ants in this approach. In Ant-Miner, pheromone volatilization was performed only for each unused term, not for the used terms, which was not consistent with the actual situation. This method aims only to strengthen the discovered solution utilization rate, and to accelerate the convergence speed, thereby reducing the exploration capability of the new and improved solutions. The new method reduces the utilization of the original solution and improves the exploration ability of the new solution.

4. Experiments and Analysis

We evaluated the performance of $m_Ant-Miner_{PR}$, which was implemented by comparing it with that of the commonly used rule induction classification algorithms, such as Ant-Miner and $m_Ant-Miner$. The results of the comparison were obtained via six sets of public data sets through an AMD(R) A6 CPU with clock rate of 1.5 GHZ and main memory of 4 G. $m_Ant-Miner_{PR}$ was developed in Java language in MyEclipse-integrated environment.

4.1 Parameter Setting of $m_Ant-Miner_{PR}$

$m_Ant-Miner_{PR}$ has the following six user-defined parameters: evaporation factor, ρ ; colony size, No_of_ants; minimum number of cases per rule, Min_cases_per_rule; maximum number of uncovered cases in the training set, Max_uncovered_cases; number of rules used to test convergence of the ants, No_rules_converge; and the maximum number of iterations, No_of_iterations. The parameter settings are shown in Table 1.

4.2 Experimental Method

We performed the experiments through six publicly available data sets from the University of California Irvine (UCI) [31] Machine Learning Repository. UCI database was proposed by UCI for machine learning database. The data set is a commonly standard test data set. The six publicly available data sets involved binary (two class values) and multiclass (more than two class values) classification problems. Table 2 presents a summary of the data sets used in the experiments. The first column of this table presents the data set name, whereas the other columns indicate the number of cases, the number of attributes, and the number of classes of the data set.

The data format of this algorithm is based on WEKA

Parameter	Ant-Miner	$_m$ Ant-Miner	$_m$ Ant-Miner _{PR}
ρ	0.9	0.9	0.9
No_of_ants	1	5	5
Min_cases_per_rule	5	5	5
Max_uncovered_cases	10	10	10
No_rules_converge	10	10	10
No_of_iteration	1000	1000	1000

Table 1. Parameters Used in Experiments

Dataset	Samples	Attributes	Classes
breast-cancer	286	10	2
car	1728	7	4
soybean	683	36	19
sponge	76	46	3
tic-tac-toe	958	10	2
vote	435	17	2

Table 2. Suite of Data sets Used in Our Experiments

Dataset	Average Predictive Accuracy		
	Ant-Miner	$_m$ Ant-Miner	$_m$ Ant-Miner _{PR}
breast-cancer	74.37±2.55	73.85±2.19	77.26±1.05
car	84.90±0.90	83.80±1.38	85.88±0.83
soybean	87.91±1.80	87.58±1.12	89.81±1.02
sponge	91.67±3.88	90.48±2.82	91.25±2.67
tic-tac-toe	72.30±2.82	71.28±2.37	74.57±1.46
vote	94.95±1.11	94.47±0.78	95.20±1.28

Table 3. Average Predictive Accuracies Obtained through 10-fold Cross-Validation for the Algorithms Proposed in This Paper

program. The format of WEKA program is attribute-relation file format, which is a kind of ASCII text file.

We performed a 10-fold cross-validation procedure [32] to test the accuracy of the algorithm. This procedure consisted of dividing the data set into 10 partitions and maintaining a similar number of examples and class distribution across all partitions. For each partition, the classification algorithm ran with the remaining nine partitions, as the training set and its performance were

evaluated with the unseen (hold-out) partition, which also typically requires multiple 10-fold cross-validations (for example, 10-fold cross-validation for 10 times). Then, its average was obtained as the accuracy of the estimation. The comparison was performed across three criteria: 1) the predictive accuracy of the discovered rule lists, namely, the correct classification rate of test data samples after the classification experiment by the algorithm; 2) the simplicity of the discovered rule list, measured by the

number of discovered rules and the average number of terms (conditions) per rule; and 3) the time of the classification algorithms ran under 10-fold cross-validation, which is used to measure the efficiency of different algorithms.

4.3 Comparisons Based on Predictive Accuracy

The results are reported in Table 3 by comparing the predictive accuracy of m Ant-Miner_{PR} and that of other classification algorithms. The numbers at right of “±” symbol are the standard deviations of the corresponding predictive accuracy rates. As shown in this table, m Ant-Miner_{PR} discovered rules with a better predictive accuracy than those of other classification algorithms in five data sets, namely, breast-cancer, car, soybean, tic-tac-toe, and vote. In the sponge data set, Ant-Miner discovered rules with a better predictive accuracy than that of other classification algorithms. However, the difference was relatively small.

The value of the most accurate algorithm for a given data set is shown in bold.

4.4 Comparisons on Rule Simplicity

We now turn to the results concerning the simplicity of the discovered rule list, which were measured through the number of discovered rules and the average number of terms per rule. The results are reported in Table 4 and Table 5 by comparing the simplicity of the rule lists discovered via m Ant-Miner_{PR} and other classification algorithms. The numbers immediately after the “±” symbol are the standard deviations of the relative number. The experimental results show that m Ant-Miner_{PR} can obtain the shortest condition number and the shortest number of rules in the concentration of six publicly available data sets, where four groups of data sets exist, namely, breast-cancer, soybean, sponge, and vote. In the car data set, discovered a compact rule list with 14.10 ± 0.31 rules, and m Ant-Miner discovered the simplicity of the average number of terms with 20.30 ± 1.39 terms per rule. In the tic-tac-toe data set, Ant-Miner discovered the optimal average rule number and the average number of terms per rule, which are 7.50 ± 0.52 and 8.00 ± 1.27 , respectively.

Dataset	Average Number of Rules		
	Ant-Miner	m Ant-Miner	m Ant-Miner _{PR}
breast-cancer	6.20±0.20	6.30±0.21	5.80±0.29
car	15.70±0.42	15.30±0.52	14.10±0.31
soybean	23.00±0.52	22.70±0.26	22.40±0.27
sponge	2.20±0.20	2.50±0.34	2.20±0.13
tic-tac-toe	7.50±0.52	8.60±0.52	8.22±0.37
vote	5.60±0.22	5.40±0.31	4.80±0.13

Table 4. Average Number of Rules (Average ± Standard Error) Measured through 10-fold Cross-Validation

Dataset	Average Number of terms		
	Ant-Miner	m Ant-Miner	m Ant-Miner _{PR}
breast-cancer	8.50±0.40	8.50±0.34	7.50±0.81
car	21.00±1.15	20.30±1.39	22.40±0.99
soybean	48.60±1.61	49.10±1.00	47.20±1.68
sponge	1.50±0.40	2.20±0.73	1.50±0.34
tic-tac-toe	8.00±1.27	10.50±1.53	9.30±1.49
vote	6.60±0.76	5.70±0.63	4.20±0.25

Table 5. Average Number of Terms (Rule Conditions) in the Discovered List (Average ± Standard Error) Measured through 10-fold Cross-Validation

Category	Average Rank		
	Ant-Miner	$_m$ Ant-Miner	$_m$ Ant-Miner _{PR}
Predictive accuracy	1.83	3.00	1.17
Model size	2.33	2.50	1.17
Number of rules	2.17	2.33	1.50

Table 6. Statistical Test Results According to Nonparametric Friedman Test with Holm's Post Hoc Test

Table 6: Presents the results of the statistical tests for predictive accuracy, size of the classification model, and total number of discovered rules, based on the nonparametric Friedman test with the Holm's post hoc test [33].

Figure 1 illustrates the average predictive accuracy rank versus the average model size rank and average predictive accuracy rank versus the average number of rules rank of the algorithms used in our experiments. The lower the average rank is, the better the algorithm performance is.

4.5 Comparison of the Algorithm Efficiency

This section calculates the time of classification algorithms that run 10-fold cross-validation, which is used to measure the efficiency of different algorithms. Table 7 presents the average computational time (in seconds) to complete 10-fold cross-validation for Ant-Miner, $_m$ Ant-Miner, and $_m$ Ant-Miner_{PR} algorithms for 10 runs. In general, the computational time taken by $_m$ Ant-Miner_{PR} is slower than the time taken by Ant-Miner. We improved the structure of the algorithm and increased the calculation method of the pheromone of repulsion based on basic ant colony classification mining algorithm. Thus, $_m$ Ant-Miner_{PR} algorithm running time is slightly longer than that of the other algorithms.

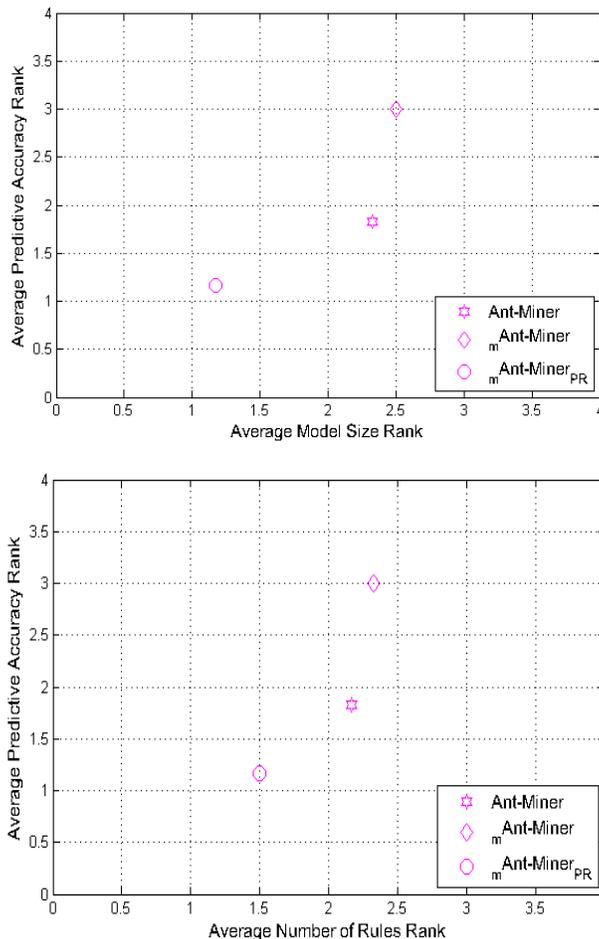


Figure 1. Average predictive accuracy rank versus average model size rank and average predictive accuracy rank versus the average number of rules rank of the algorithms

Dataset	Average Computational Time		
	Ant-Miner	$_m$ Ant-Miner	$_m$ Ant-Miner _{PR}
breast-cancer	0.109	0.734	1.731
car	0.203	2.091	5.431
soybean	14.906	636.671	339.067
sponge	0.484	32.776	4.54
tic-tac-toe	0.125	0.827	0.328
vote	0.141	1.264	3.666

Table 7. Average Computational Time in Seconds Taken by Ant-Miner, $_m$ Ant-Miner, and $_m$ Ant-Miner_{PR} to Complete 10-Fold of the Cross-Validation

In many data mining applications, the computational time taken by the algorithm to induce a classification model has a minor importance. However, it needs to have a high prediction accuracy and concise rules. In addition, $_m$ Ant-Miner_{PR} algorithms can be easily parallelized, given that each ant builds and evaluates a candidate solution independent from all the other ants, which may also increase the speed of the algorithm. Thus, we aim to focus on computational time in future work.

5. Conclusion and Future Work

5.1 Conclusion

In this paper, a new ant colony classification mining algorithm based on the principles of multi-ants and pheromone repulsion, namely, $_m$ Ant-Miner_{PR} is proposed according to classical Ant-Miner algorithm. Improvements are attained in four aspects. The algorithm uses multi-ant colony construction method to reduce dependence on the random initial term. The volatile coefficient of pheromone update is added to prevent premature convergence. The new state transition function based on pheromone repulsion principle is also presented to avoid falling into local optimum, whereas the rule quality formula is improved. The improved algorithm has a significant improvement in the prediction accuracy, whereas the list of rules from the improved algorithm is more concise than that of the original algorithm.

5.2 Future Work

No serious attempt was made to optimize the setting of the parameters in $_m$ Ant-Miner_{PR}. This optimization can be attempted in upcoming research. Improving the understanding and scalability of classification knowledge is an important task of classification mining. It is also the direction of our future research. In addition, obtaining a large speed up by running a parallel version of $_m$ Ant-Miner_{PR} is an interesting topic to study.

6. Acknowledgment

This work was partially supported by Science and Technology Project of Guangdong Province of China (Grant No. 2014A020208087), Fund of Natural Science Foundation of Guangdong Province of China (Grant No. 2013040015755), National Spark Technology Project (Grant No. 2014GA780051), and National Spark Technology Project (Grant No. 2013GA780044). The authors would like to thank the anonymous reviewers for their useful comments and suggestions.

References

- [1] Jiawei, H., Micheline, K., Jian, P (2012). *Data Mining: Concepts and Techniques*, 3rd ed. 1-25.
- [2] Salama, K.M., Abdelbar, A.M., Freitas, A.A (2011). Multiple pheromone types and other extensions to the Ant-Miner classification rule discovery algorithm. *Swarm Intelligence*, 5 (3-4) 149-182.
- [3] Baig, A. R., Shahzad, W., Khan, S. (2013). Correlation as a Heuristic for Accurate and Comprehensible Ant Colony Optimization Based Classifiers. *IEEE Transactions on Evolutionary Computation*, 17 (5) 686-704.
- [4] Otero, F. E. B., Freitas, A. A., Johnson, C. G. (2013). A New Sequential Covering Strategy for Inducing Classification Rules with Ant Colony Algorithms. *IEEE Transactions on Evolutionary Computation*, 17 (1) 64-76.
- [5] Baig, A.R., Shahzad, W. (2012). A correlation based Ant Miner for classification rule discovery. *Neural Computing & Applications*, 21 (2) 219-235.
- [6] Yang, J., Luo, Z. (2007). Coalition formation mechanism in multi-agent systems based on genetic algorithms. *Applied Soft Computing*, 7 (2) 561-568.
- [7] Jiang, J., Xia, N., Qi, M., Mu, C. (2005). An Ant Colony Algorithm Based Multi-task Coalition Serial Generation

- Algorithm. *Acta Electronica Sinica*, 33 (12) 2178-2182.
- [8] Zhang, G., Jiang, J., Xia, N., Su, Z. (2007). Solutions of Complicated Coalition Generation Based on Discrete Particle Swarm Optimization. *Acta Electronica Sinica*, 35 (2) 323-327.
- [9] Meyer, F., Parpinelli, R.S (2012). ACO: Public Software, <http://www.aco-metaheuristic.org>.
- [10] Colorni, A., Dorigo, M., Maniezzo, V (1991). Distributed Optimization by Ant Colonies. In: Proceedings of the ECAL91-European Conference on Artificial Life, Elsevier Publishing, 134-142.
- [11] Dorigo, M., Stützle, T. (2004). Ant Colony Optimization. 1-328.
- [12] Liu, B., Pan, J. (2007). Research of classification algorithm based on Ant Colony Optimization. *Computer applications and software*, 24 (4) 50-53.
- [13] Parpinelli, R.S., Lopes, H.S., Freitas, A.A (2002). Data mining with an ant colony optimization algorithm. *IEEE Transactions on Evolutionary Computation*, 6 (4) 321-322.
- [14] Parpinelli, R.S., Lopes, H.S., Freitas, A.A. (2002). An ant colony algorithm for classification rule discovery. *Data Mining: A Heuristic Approach*, 191-208.
- [15] Martens, D., Baesens, B., Fawcett, T. (2011). Editorial survey: Swarm intelligence for data mining. *Machine learning*, 82 (1) 1-42.
- [16] Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. 1-107.
- [17] Clark, P., Niblett, T. (1989). The CN2 induction algorithm. *Machine learning*, 3(4) 261-283.
- [18] Clark, P., Boswell, R. (1991). Rule induction with CN2: Some recent improvements. In: Proceedings of the European Working Session on Learning (EWSL-91), Heidelberg, Berlin: Springer, p. 151-163.
- [19] Jin, P., Zhu, Y., Hu, K., et al. (2006). Classification rule mining based on ant colony optimization algorithm. *Intelligent Control and Automation*, 654-663.
- [20] Liu, B., Abbass, H.A., McKay, B. (2003). Classification rule discovery with ant colony optimization. In: *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology*, Halifax, Canada, 83-88.
- [21] Wang, Z., Feng, B. (2005). Classification rule mining with an improved ant colony algorithm. *AI 2004: Advances in Artificial Intelligence*, 357-367.
- [22] Liu, B., Abbass, H. A., McKay, B. (2002). Density-based heuristic for rule discovery with Ant-Miner. In: *Proceedings of the 6th Australasia-Japan Joint Workshop on Intelligent and Evolutionary System*, Canberra, Australia, 180-184.
- [23] Martens, D (2012). Classification with Ant Colony Optimization, <http://www.antminerplus.com>.
- [24] Chan, A., Freitas, A. (2006). A new classification-rule pruning procedure for an ant colony algorithm. *Artificial Evolution*. 25-36.
- [25] Otero, F. E. B., Freitas, A. A., Johnson, C. G. (2008). cAnt-Miner: An Ant Colony Classification Algorithm to Cope with Continuous Attributes. *Ant colony optimization and swarm intelligence*, 48-59.
- [26] Otero, F. E. B., Freitas, A.A., Johnson, C. G (2009). Handling continuous attributes in ant colony classification algorithms. In: Proceedings of the Computational Intelligence and Data Mining, Nashville, TN: IEEE, 225-231.
- [27] Chan, A., Freitas, A. A. (2006). A new ant colony algorithm for multi-label classification with applications in bioinformatics. In: *Proceedings of the 8th annual conference on Genetic and evolutionary computation (GECCO'06)*, New York, USA: ACM, 27-34.
- [28] Smaldon, J., Freitas, A. A (2006). A new version of the Ant-Miner algorithm discovering unordered rule sets. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO'06), New York, USA: ACM, 43-50.
- [29] Parpinelli, R. S., Lopes, H. S., Freitas, A. A (2001). An Ant Colony Based System for Data Mining: Applications to Medical Data. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), Morgan Kaufmann, San Francisco, 791-798.
- [30] Xiao, J., Liang, Y. (2012). Classification rule mining based on improved Ant-Miner algorithm, *Computer Engineering*, 38 (17) 162-165.
- [31] Asuncion, A., Newman, D (2010). UCI Machine Learning Repository, <http://archive.ics.uci.edu/ml/>.
- [32] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1137-1143.
- [33] Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research*, 7. 1-30.