

# An Adaptive Ant Colony Algorithm for Classification Rule Mining

Xiaomeng Zhang and Wensheng Sun\*

School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, XiTucheng Road  
No. 10, Haidian District, Beijing, China

\*Corresponding author

**Abstract**—Ant-Miner algorithm is a typical classification rule mining algorithm which can improve the classification accuracy and generate simple rules. However, it also has a few disadvantages, such as complicated computing method for heuristic factor, long calculation time, slow evolution and so on. Based on the Ant-Miner algorithm, this paper adjusted the probability of the deterministic selection and the volatility coefficient dynamically and adaptively. This not only guarantees the convergence speed but also improves the global search ability. To verify the effectiveness of the algorithm, we used public database UCI datasets for algorithm simulation. Compared with the Ant-Miner algorithm, the proposed algorithm improves the classification accuracy rate and gives more concise rules.

**Keywords**—data mining; adaptive ant colony algorithm; classification rule; pheromone

## I. INTRODUCTION

According to [1], data classification is a kind of data mining form that constructs the classifier by taking the training sample dataset as input. Reference [2] pointed out a rule-based classifier is a technique for classifying data by a set of rules. Reference [3] proposed that ant colony algorithm is a combinatorial optimization algorithm which is inspired by ant foraging behavior. In recent years, its application has been expanded to achieve good experimental results in many fields, such as combinatorial optimization, neural network, artificial intelligence. Reference [4] introduced that the algorithm had been applied in the aspects of grid service reliability assurance. Reference [5] applied the ant colony algorithm to settle the problem of sequence coverage. Ant-Miner algorithm is proposed by [6] based on ant colony algorithm for classification rule mining algorithm. This algorithm improves classification accuracy and receives good classification results in [7].

Ant-Miner algorithm established the relationship between ant colony algorithm and classification rule mining for the first time. After analyzing the characteristics of the information evaporation factor, an adaptive mechanism is built based on normal distribution function, and then that is added to the Ant-Miner algorithm to form a new algorithm. In the proposed algorithm, the probability of the deterministic selection is adjusted dynamically. The improved algorithm not only accelerates the convergence speed, but also produces higher classification accuracy and more concise rules.

## II. AN ALGORITHM OF CLASSIFICATION RULE MINING BASED ON ADAPTIVE ANT COLONY ALGORITHM

Reference [8] pointed out the rule-based classifier is an important technique to construct the classification model. The model obtained by learning can be represented by a set of "if ... then ..." rules. Each classification rule can be expressed as an expression of the form:

**IF** condition **THEN** consequent

where, the left side of the rule is called the rule condition and the right side the rule consequent.

### A. Adaptive Ant Colony Optimization Model

For a given classification problem, suppose that there are  $k$  pieces of data and  $a$  attribute variables  $A_i$  ( $i = 1, 2, \dots, a$ ) in the data set.  $V_{ij}$  represents the  $j$ th attribute value of the attribute  $A_i$ .  $b_i$  denotes the number of the attribute values of the attribute  $A_i$ . Attribute  $term_{ij}$  indicates  $A_i = V_{ij}$ . Similar to traveling salesman problem, the algorithm needs to calculate the probability of being selected for all the cities not in tabu list. Also, in the sequential covering algorithm based on rule classification, it is necessary to select the best rule in the current class. A  $term_{ij}$  is chosen to join in the current rule each time. The probability of the  $term_{ij}$  is given as

$$p_{ij}(t) = \frac{\tau_{ij}(t)^\alpha \cdot \eta_{ij}(t)^\beta}{\sum_i^a \sum_j^{b_i} \tau_{ij}(t)^\alpha \cdot \eta_{ij}(t)^\beta} \quad (1)$$

We combine stochastic selection strategy with deterministic selection strategy to adjust the selection strategy dynamically [9]. Specifically, after a certain number of iterations are evolved, it is necessary to narrow the gap between the best and the worst case of the path pheromone concentration values and increase the probability of the random selection, thus using the following path selection probabilities:

$$term_{ij} = \begin{cases} \arg \max \{ \tau_{ij}(t)^\alpha \cdot \eta_{ij}(t)^\beta \}, & \text{if } r \leq p_0 \\ \text{select } term_{ij} \text{ according to probability } p_{ij}, & \text{else} \end{cases} \quad (2)$$

where  $p_0 \in (0,1)$ ,  $r$  is a uniformly distributed random number in  $(0,1)$  interval.

1) *Rule generation*:  $\tau_{ij}(t)$  is the pheromone concentration of the  $j$ th attribute item of attribute  $A_i$  at time  $t$ . The initial pheromone concentration is set to

$$\tau_{ij}(0) = \frac{1}{\sum_{i=1}^a b_i} \quad (3)$$

After an ant constructs a rule, the pheromone of all attribute items is updated. For the attribute item in the obtained rule, the pheromone update is performed as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \tau_{ij}(t) \cdot Q \quad (4)$$

where  $\rho$  represents information evaporation factor.  $1-\rho$  denotes the pheromone residual factor.  $Q$  indicates the rule validity.

For attribute items that are not included in the rule, the pheromone update strategy is

$$\tau_{ij}(t+1) = \frac{\tau_{ij}(t)}{\sum_i \sum_j \tau_{ij}(t)} \quad (5)$$

$\eta_{ij}(t)$  is the heuristic factor of the attribute  $term_{ij}$  at the moment  $t$ . This paper does not intend to calculate the information entropy of attribute via Ant-Miner algorithm, because the method is computationally complicated and the associated computation time is long. We employ density-based heuristic factors instead, that is

$$\eta_{ij}(t) = \frac{|T_{ij}(t)|}{|T(t)|} \quad (6)$$

where  $|T_{ij}(t)|$  is the number of data in the current data set containing attribute  $term_{ij}$ , and  $|T(t)|$  is the total number of data in the current data set.

2) *Pruning strategy*: For rules containing a large number of attribute items but a small amount of training records, it will inevitably lead to over-fitting. Therefore, it is necessary to prune the rules to avoid the results. In order to determine whether pruning is to be performed, rule validity  $Q$  can be

applied to measure the effectiveness of the rule before and after pruning:

$$Q = \frac{TP}{TP+FN} \cdot \frac{TN}{TN+FP} \quad (7)$$

$TP$  is the number of instances that both rule condition and rule consequent match with data;  $FP$  is the number of instances that rule condition matches but rule consequent mismatch; the scenario of  $FN$  is just the opposite of  $FP$ ;  $TN$  is the number of instances that neither of them matches.

If  $Q$  increases after pruning, the rule will be selected. A simple pruning strategy is to remove the attribute item in the current rule that can maximize the validity of the rule. After the pruning, the rule with the maximum rule validity is obtained.

3) *Adaptive design of  $\rho$* : In the process of constructing rules, when the search comes to a certain extent, the rules obtained by ants are exactly the same, as a result, the solution space cannot be explored further, and the ant colony algorithm emerges a local stagnation phenomenon that is not conducive to discover better rules. Reference [10] updated the quantity of information based on the uniformity of pheromone. In [11], the pheromone was updated according to the weight of the difference between the path taken by individual ants and the mean value of paths taken by all ants. In [12], the authors pointed out that too-small  $\rho$  (the pheromone volatility) would make the pheromone that was not searched on the path reduce to 0 approximately. If  $\rho$  is too large, then the pheromone on selected path will reduce sharply. According to the experience of the past references,  $\rho(t)$  needs to meet the following three conditions:

$$(1) 0 < \rho < 1;$$

(2) In the initial stage of the path search, in order to speed up the convergence speed, we need to select a smaller  $\rho$  value. When the search comes to a certain extent, we need to increase the value of  $\rho$  to avoid falling into the local stagnation;

(3) The  $\rho$  value cannot be too large (close to 1). If  $\rho$  is too large, it will reduce the convergence rate too much.

Based on the above analysis of the characteristics of the  $\rho$  value, we establish the following adaptive adjustment mechanism for  $\rho(t)$ :

$$\rho(t) = \frac{3}{2} \int_0^t f(\tau) d\tau \quad (8)$$

where  $f(t)$  is the normal distribution function when  $\mu=0$ , the formula is given as follows

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-t^2/2\sigma^2} \quad (9)$$

According to the basic knowledge of probability,  $\rho(t)$  in formula (8) has the following properties:

- (1)  $0 < \rho \leq 0.75$  ;
- (2) As  $t$  increases, the value of  $\rho(t)$  will gradually increase, but the growth will slow down.
- (3) The maximum value of  $\rho(t)$  is 0.75.

Obviously, the adaptive mechanism established for  $\rho$  is in accordance with the selection of  $\rho$  value.

Figure I illustrates the function images of  $f(t)$  when the value of  $\sigma$  is 4 and 10, respectively. The decreasing speed at  $\sigma=10$  is much slower than that at  $\sigma=4$ . When  $t=5$ , the integral area of  $f(t)$  at  $\sigma=4$  is obviously larger than that at  $\sigma=10$ . This will cause the volatility  $\rho$  to be too large and lead to a reduction of the convergence rate in a short time. Hence, we select  $\sigma=10$  to make the trend of volatility  $\rho$  increase moderately.

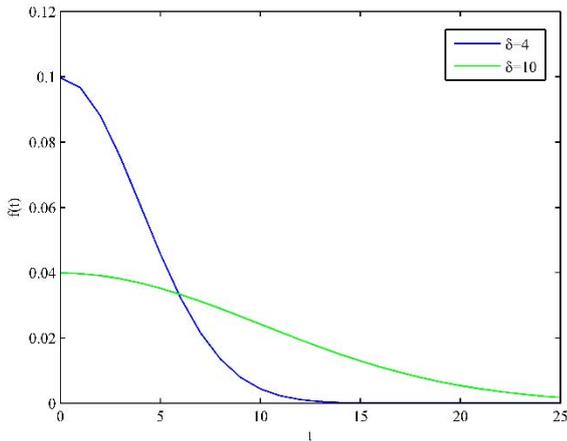


FIGURE I.  $f(t)$  FUNCTION IMAGE COMPARISON WHEN  $\sigma=4$  AND  $\sigma=10$

### B. Algorithm Description

Algorithm: adopt adaptive ant colony algorithm to learn a set of IF-THEN classification rules.

Input: a set of data marked with a class.

Output: IF-THEN rule set.

Adaptive ant colony optimization algorithm steps are listed as follows:

- (1) Initialization
  - (1.1) Initialize an empty rule list **ruleBestList** that can store obtained rules.
- (2) Iterative process

While (The number of samples in dataset is greater than the maximum number of uncovered samples)

For  $k=1, 2, \dots, m$  (The circulation of  $m$  ants)

(2.1) Define an empty rule list named **ruleTempList**, and store the rules for each ant.

(2.2) Initialize time  $t=0$ . In accordance with the formula (3) and (6), initialize the pheromone concentration and heuristic factor of each attribute value in all attributes, and initialize an empty rule named **ruleTemp**.

(2.3) Calculate and compare the probability of each property item based on the formula (1), to find the value of the property  $term_{ij}$  with maximum probability, then add  $term_{ij}$  to the rule **ruleTemp**. After those operations, we filter out all the data that contains the attribute  $term_{ij}$  in the current training set and remove the attribute  $term_{ij}$  from the current attribute set.

(2.4) If **ruleTemp** contains all the attributes, then performing step (2.5). If not, applying formula (6) to calculate the heuristic factor of the current attribute set of all the attribute items, then performing step (2.3).

(2.5) Prune the rule with the pruning strategy described in A.2). Get a rule after pruning. Put the rule into **ruleTempList** and update the pheromone on the attribute items via formulas (4) and (5). Number of ants  $k = k + 1$ .

end for  $k$

(3) Compare the rule validity  $Q$  of all the rules in **ruleTempList**. Select the largest rule named **ruleBest** and store it in the **ruleBestList**. All data that satisfy **ruleBest** are removed in the training set and a new training set **Data** is obtained.

end while

Specific flow chart is illustrated in Figure II.

### III. SIMULATION EXPERIMENT AND RESULT ANALYSIS

Tic\_tac\_toe and Breast cancer datasets were selected from the UCI public database. 90% of the datasets were randomly selected as the training data set and the remaining 10% as the testing data.

Ten experiments were carried out using 10 folds cross validation. Table I shows the details of the data set, including the number of samples in the dataset, the number of attributes and the number of categories.

The parameters of the algorithm are set as follows: the total number of ants is 30, and the maximum number of uncovered samples is 5. Table II shows the experimental results of the proposed algorithm. As for the dataset Tic\_tac\_toe, we find that  $\alpha=3$  and  $\beta=2$  are the optimal combination in all the results, and for the dataset Breast cancer,  $\alpha=4$  and  $\beta=4$  are the best combination among the results. These optimal combinations in two cases will be utilized to compare the performance of different algorithms.

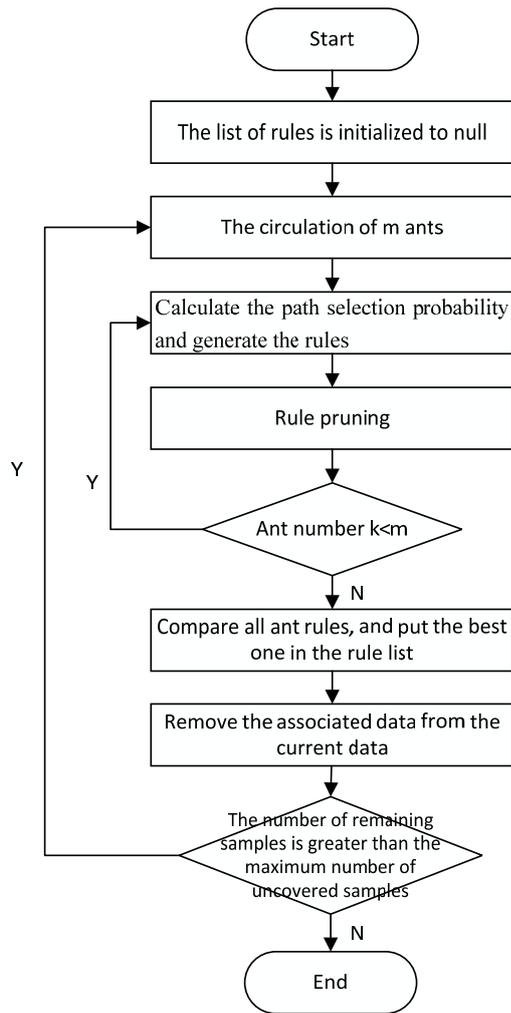

**FIGURE II. FLOW CHART OF THE PROPOSED ALGORITHM**

Table III reveals that the experimental results of the dataset Tic\_tac\_toe. Table IV shows that the experimental results of the dataset Breast cancer. The figures in Table III and IV illustrate the classification accuracy and the average number of rules in three algorithms, namely C4.5, Ant-Miner and this paper's algorithm. As can be seen from Table III, the classification accuracy of three algorithms, in decreasing order, is this paper's algorithm (89.66%), C4.5 (83.18%), Ant-Miner algorithm (73.04%). While the average number of rules of them behaves opposite patterns, that is, this paper's algorithm (8.3), C4.5 (8.3) and Ant-Miner algorithm (8.5). It is clear from Table IV that the classification accuracy of this paper's algorithm is considerably higher than that of others, achieving at 83.02%. While its average number of rules is lower than the Ant-Miner algorithm. In contrast, the C4.5 experiences the poorest classification accuracy of all, and followed closely by Ant-Miner algorithm.

Overall, it is proved that the improved algorithm of this paper is better than the other algorithms. Specially, the rules obtained by this paper's algorithm are more concise than Ant-Miner algorithm.

**TABLE I. EXPERIMENTAL DATA SET**

	Number of samples	Number of attributes	Number of categories
Breast ancner	286	9	2
Tic_tac_toe	958	9	2

**TABLE II. THE EXPERIMENTAL RESULTS OF DIFFERENT  $\alpha, \beta$  VALUES**

$\alpha$	$\beta$	Tic_tac_toe		Breast Cancer	
		Accuracy of classification	Average number of rules	Accuracy of classification	Average number of rules
1	1	86.59%	8.0	82.02%	6.7
	2	86.37%	7.7	78.70%	6.7
	3	86.80%	7.6	80.66%	6.6
	4	88.91%	8.4	82.59%	6.5
2	1	87.50%	7.9	80.55%	6.7
	2	87.81%	8.4	81.21%	6.8
	3	86.92%	8.2	82.29%	6.8
	4	88.14%	8.4	81.95%	6.6
3	1	88.29%	8.2	80.53%	6.8
	2	89.66%	8.3	76.85%	6.7
	3	87.61%	8.1	80.89%	6.9
	4	87.77%	7.9	82.06%	6.9
4	1	87.76%	8.1	80.38%	6.8
	2	89.29%	8.5	80.17%	6.6
	3	87.39%	8.0	78.92%	6.8
	4	88.87%	8.3	83.02%	6.9

**TABLE III. EXPERIMENTAL RESULTS OF THE DATASET TIC\_TAC\_TOE**

	C4.5	Ant-Miner	This paper's algorithm
Accuracy of classification	83.18%	73.04%	89.66%
Average number of rules	8.3	8.5	8.3

**TABLE IV. EXPERIMENTAL RESULTS OF THE DATASET BREAST CANCER**

	C4.5	Ant-Miner	This paper's algorithm
Accuracy of classification	73.34%	75.42%	83.02%
Average number of rules	6.2	7.2	6.9

#### IV. CONCLUSION

This paper proposes an improved adaptive ant colony algorithm based on Ant-Miner algorithm. It combines the strategy of stochastic selection with deterministic selection and provides the method of adaptively adjusting the evaporation coefficient. Meanwhile, the updating strategy of heuristic factor pheromone is improved and the calculation is also simplified. The effectiveness of the proposed algorithm is illustrated in datasets from the UCI public database. The experimental results show that the improved adaptive ant colony algorithm is better than Ant-Miner algorithm and C4.5, this is most evidence in their classification accuracy and the average number of rules.

## V. ACKNOWLEDGEMENTS

It is supported by national natural science foundation of china under grant 61302080.

## REFERENCES

- [1] T. Pang-ning, M. Steinbach, and V. Kumar, Introduction to data mining, 1st ed. Beijing: China Machine Press, 2010.
- [2] H. Jia-wei, K. Micheline, Data mining: concept and techniques, 2nd ed. Beijing: China Machine Press, 2006.
- [3] M. Dorigo, and L.M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," IEEE T. Evolut. Comput, vol. 1, pp. 53-66, April 1997.
- [4] X. Ya-mei, C. Bo, C. Jun-liang, "Optimizing services composition based on improved ant colony algorithm," Chin. J. Comput, vol. 35, pp. 270-281, February 2012.
- [5] M. An-xiang, Z. Chang-sheng, Z. Bin, "Adaptive artificial bee colony algorithm for classification problem," J. JILIN. U: TECHNO. ED, vol. 46, pp. 252-258, January 2016.
- [6] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, "An ant colony algorithm for classification rule discovery," (chapter 10 in Data mining: A heuristic approach), pp. 191-208, 2002.
- [7] H. Zhi-yuan, Z. Zheng-yin, T. Ke-zong, and G. Shang, "Distributed classification rule mining based on colony algorithm," Sci. Technol. Eng, vol. 7, pp. 199-202, January 2007.
- [8] M. Xian-ming, and L. Pei-liang, "Data mining model s based on adaptive ant colony optimization algorithm," Chin. J. Construct. Mach, vol. 7, pp. 275-279, September 2009.
- [9] Z. Ji-hui, G. Qi-sheng, and X. Xin-he, "A self-adaptive ant colony algorithm," Control. Theory. & APPL, vol. 17, pp. 1-3, February 2000.
- [10] C. Ling, S. Jie, Q. Ling, and C. Hong-Jian, "An adaptive ant colony algorithm based on equilibrium of Distribution," J. Software. Vol. 14, pp. 1379-1387, July 2008.
- [11] G. Shi-wei, G. Lei, D. Ya-qin, Y. Ning, and C. Liang, "Adaptive ant colony algorithm based on dynamic weighted rule," Comput. Appl, vol. 27, pp. 1741-1743, July 2007.
- [12] W. Ying, and X. Jian-ying, "An adaptive ant colony optimization algorithm and simulation," J. Syst. Simul, vol. 14, pp. 31-33, January 2002.