

Modified Ant Colony Optimization Algorithm (MAnt-Miner) for Classification Rule Mining

Sarbeswara Hota, Pranati Satapathy and Alok Kumar Jagadev

Abstract Classification rule mining is an important task of data mining. Ant colony optimization (ACO) algorithms are applied successfully to various optimization problems. Earlier Ant-Miner, an ACO algorithm was used to discover the classification rules and predictive accuracy was determined. In this paper, modified ant colony optimization (MAnt-Miner) is proposed to generate the classification rules and to enhance the predictive accuracy. This method is applied on breast cancer data set, and the experimental result showed that the predictive accuracy of MAnt-Miner is better than Ant-Miner.

Keywords Classification · Heuristic · Pheromone · Swarm intelligence

1 Introduction

Out of the several data mining tasks, classification is an important task. Classification problems can be viewed as optimization problems, where the goal is to find the best model that represents the predictive relationships in the data [1]. Actually, a classification problem consists of generating a small set of rules with predefined

S. Hota (✉) · A.K. Jagadev
Siksha O Anusandhan University, Bhubaneswar, India
e-mail: sarbeswarahota@soauniversity.ac.in

A.K. Jagadev
e-mail: alokjagadev@soauniversity.ac.in

P. Satapathy
Department of IMCA, Utkal University, Bhubaneswar, India
e-mail: satapathy.pranati@gmail.com

class labels. The rules are generated from the training data set. The generated rules are then used to classify, i.e., to predict the class labels of unknown instances. The classification rules are of the form

$$\text{IF } \langle \text{condition} \rangle \text{ THEN } \langle \text{class} \rangle.$$

where condition is expressed as term1, term2, and so on, and each term is a triplet $\langle \text{attribute, operator, value} \rangle$.

The classification rule generation problem is NP-hard because for n number of attributes, the number of conditions is $O(2^n)$ that is exponential in nature. So many researchers have applied biologically inspired algorithms on these types of NP-hard problems to get the solutions.

Ant colony optimization (ACO) algorithm, one of the biologically inspired algorithms has been used extensively in classification task of data mining. Parpinelli et al. [2] are the first to propose ACO algorithm for discovering classification rules, known as Ant-Miner. Later on, this algorithm has been studied extensively by many researchers and modifies in various forms. The objective of our work is to propose a new heuristic value and pheromone updating strategy in the Ant-Miner.

The remainder of this paper is organized as follows. Section 2 introduces the ACO algorithms. Section 3 introduces the Ant-Miner. The proposed model of our work and the modifications in the Ant-miner are presented in Sect. 4. The experimental results obtained on one benchmark data set are presented in Sect. 5, and Sect. 6 concludes this paper.

2 Ant Colony Optimization Algorithm

ACO [3] is a branch of newly developed form of artificial intelligence called swarm intelligence. Swarm intelligence is a field that studies the collective intelligence of a swarm, i.e., a group of insects, which live in colonies such as ants, bees, etc. A single ant can do some simple tasks on its own, but the intelligent behavior shown by the colony lies in their cooperative work. Dorigo and Maniezzo presented the cooperative behavior of ant colonies. ACO algorithms [3] are based on the following ideas:

- The path traversed by an ant refers to a solution of a problem.
- The pheromone amount deposited on a path represents the strength of solution.
- The probability of choosing a path depends on the amount of pheromone laid by an ant.

In brief, the design of an ACO algorithm involves the following specification:

- Formulation of a problem based on a problem-dependent heuristic function and probability calculation of the edges between the vertices.
- A heuristic function (η) that represents the quality of the sub-solutions.
- The pheromone updating method, which increases or decreases pheromone trail (τ).
- A probability function that build the solution of the problem.

The basic pseudocode of ACO is as follows.

```

Procedure ACO
{
    While (termination condition is not reached)
    {
        Construct_Solution ()
        Perform_Actions ()
        Update_Pheromone ()
    }
}

```

3 Ant-Miner

The purpose of Ant-Miner is to discover classification rules from data set [2].

Algorithm Ant-Miner

```

{
    Training_set= all training cases;
    DiscoveredRuleList=[ ];
    WHILE (No. of cases in the training set > max_uncovered_cases)
    {
        i=0;
        REPEAT
        i=i+1;
        Anti incrementally constructs a classification rule;
        Prune the just constructed rule;
        Update the pheromone of the trail followed by Anti;
    UNTIL (i>= No_of_Ants) or (Anti constructed the same rule as the previous
    No_Rules_Converg-1 Ants)
    Select the best rule Rbest among all constructed rules;
    Add rule Rbest to DiscoveredRuleList;
    Remove the cases correctly covered by the selected rule from the training set;
    }
}

```

In Ant-Miner, initially the DiscoveredRuleList is empty and the training set is all the training cases. On each execution of WHILE loop that incorporates a number of executions of nested REPEAT-UNTIL loop, one classification rule is

generated. The generated rule is appended to the DiscoveredRuleList, and the training cases that are correctly classified by this rule (i.e., instances satisfying the rule conditions and having the class predicted by the rule conclusion) are removed from the training set. This process is iterated while the number of uncovered training cases is greater than a user-specified threshold, called Max_uncovered_cases.

Each execution of the REPEAT-UNTIL loop of Ant-Miner algorithm comprises of these basic processes,

- Pheromone Initialization (τ) and pheromone updating strategy
- Problem-dependent heuristic value calculation (η)
- Rule construction using probability calculation of adding a term

4 Proposed Model and Modified Ant-Miner

The primary purpose of this paper is to discover the classification rules in a particular data set and classify with test data. Thus, it computes the classification accuracy. It performs the rule mining with the modified Ant-Miner. The general model of this work is described with the following steps as:

Step 1: Identification of categorical attributes and number of classes of a training data set

Step 2: Study of ACO algorithm for classification rule and its modifications

Step 3: Discover the classification rule list on the given data set

Step 4: Calculate the predictive accuracy of the rule list on the test data set and compare with Ant-Miner

In Ant-Miner, the ants select terms on the basis of pheromone amount and heuristic function, which measures the predictive power of a term. But in this method, the pheromone of each term is changed after an ant constructs a rule, while heuristic function is always the same, so that the next ant tends to choose terms used in the previous rule, whose pheromone is increased and is unlikely choose unused terms, whose pheromone is decreased. Consequently, the ants converge to a single constructed rule too quickly. This prevents to produce other potential rules.

Hence, in this paper, we use the existing pheromone initialization and probability calculation functions as that of Ant-Miner but a different heuristic function and pheromone updating method to find the most predictive terms in a rule.

4.1 Pheromone Initialization

Each term_{ij} (term corresponding to attribute *i* and value *j*) corresponds to a segment in some path that can be followed by an ant. At each iteration of the WHILE loop of the Ant-Miner algorithm, all term_{ij} are initialized with the same amount of pheromone, so that when the first ant starts its search, all paths 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 as

$$\tau[i,j](t = 0) = \frac{1}{\sum_{i=0}^a b[i]} \quad (1)$$

where *a* is the total number of attributes, and *b*[*i*] is the number of possible values that can be taken on by attribute *A_i*.

4.2 Heuristic Function

Heuristic function $H(WA_i = V_{ij})$ of term_{ij} computation of Ant-Miner [2] is based on information entropy and normalization, but this value is always the same irrespective of the contents of the rule in which the term occurs. Consequently, the ants likely converge to a single constructed rule too quickly. This avoids producing alternative potential rules. In order to overcome these shortcomings, we use a simple heuristic function [4], which is defined as follows:

$$\eta[i,j] = \frac{-\sum_{w=0}^k \text{freq } T_{ij}^w \log_2 \text{freq } T_{ij}^w}{|T_{ij}|} \quad (2)$$

where *k* is the number of classes;

$|T_{ij}|$ is the total number of cases in partition *T_{ij}*;

freq *T_{ij}^w* is the number of cases in partition *T_{ij}* with class *w*.

4.3 Probability Calculation

As specified in the Ant-Miner, the probability that term_{ij} is chosen to be added to the current partial rule is given as

$$\text{Prob}[i, j] = \frac{\eta[i, j] \cdot \tau[i, j](t)}{\sum_{i=0}^n x[i] \cdot \sum_{j=1}^{b_i} (\eta[i, j] \cdot \tau[i, j](t))} \quad (3)$$

where

- a is the total number of attributes,
- x_i is set to 1 if the attribute A_i was not yet used by the current ant, or to 0 otherwise.,
- b_i is the number of values in the domain of the i th attribute.

4.4 Pheromone Updating Strategy

After the rule construction by an ant, the amount of pheromone associated with each term that occurs in the constructed rule is updated [5] and the pheromone of unused terms is updated by normalization. For pheromone updating, we use the following method

$$\tau[i, j](t) = (1 - \rho) + \left(1 - \frac{1}{1 + Q}\right) \cdot \tau[i, j](t - 1) \quad (4)$$

where ρ is the pheromone evaporation rate and Q is the quality of the constructed rule. In this paper, we take $\rho = 0.1$. The quality of a rule is computed by the formula: $Q = \text{sensitivity} \cdot \text{specificity}$, defined as:

$$Q = \frac{\text{TP}}{\text{TP} + \text{FN}} \cdot \frac{\text{TN}}{\text{FP} + \text{TN}} \quad (5)$$

where:

- TP (true positives) is the number of cases covered by the rule that has the class predicted by the rule.
- FP (false positives) is the number of cases covered by the rule that has a class different from the class predicted by the rule.
- FN (false negatives) is the number of cases that are not covered by the rule but that has the class predicted by the rule.
- TN (true negatives) is the number of cases that are not covered by the rule and that do not have the class predicted by the rule

The pheromone of unused terms is updated by normalization given as

$$\tau[i, j](t) = \frac{\tau[i, j](t - 1)}{\sum \tau[i, j](t - 1)} \quad (6)$$

5 Experimental Results

The performance of the modified Ant-Miner was evaluated using one public-domain data set from the UCI repository, i.e., the breast cancer data set having 659 cases, 9 categorical attributes, and 2 class labels. In our experiment, the parameters were set as follows:

- $No_of_ants = 200$.
- $Min_cases_per_rule = 30\%$ of the total training data set.
- $Max_uncovered_cases = 20\%$ of the total training data set.
- $No_rules_converg = 10$.

The comparison between Ant-Miner and modified Ant-Miner was carried out based on the predictive accuracy of the discovered rule lists. The number right after the “±” symbol is the standard deviation of the corresponding predictive accuracies rates (Tables 1 and 2).

From the above comparison, it is concluded that proposed Ant-Miner discovered rules with a better predictive accuracy than Ant-Miner in the data set. In addition, we can use the confusion matrix for analyzing the performance of both the algorithms. The confusion matrices for the two classes are shown in Tables 3 and 4.

Using confusion matrix as a tool for analyzing the performance, the accuracy rate can be calculated as

$$Accuracy\ rate = \frac{TP + TN}{TP + FN + FP + TN} \tag{7}$$

By using confusion matrix tool for analyzing the performance of the above two algorithms, it is concluded that the accuracy rate of the modified Ant-Miner is

Table 1 Test runs for predictive accuracy

Run number	Ant-Miner	Modified Ant-Miner
1	92.3695	97.1888
2	93.1727	98.7952
3	93.9759	98.7952
4	92.3695	97.1888
5	94.7791	96.7791
6	95.5823	96.3855
7	96.3855	97.1888
8	93.9759	97.9920
9	94.7791	97.1888
10	92.3695	95.5823

Table 2 Mean accuracy rate with standard deviations

Valuation item	Ant-Miner	Modified Ant-Miner
Accuracy rate (%)	94.2168 ± 1.6481	97.5100 ± 1.2672

Table 3 Confusion matrix for Ant-miner

	Class1	Class2	Total
Class1	194	5	199
Class2	9	42	51
Total	203	47	250

Table 4 Confusion matrix for modified Ant-Miner

	Class1	Class2	Total
Class1	195	4	199
Class2	2	49	51
Total	197	53	250

Table 5 Accuracy rate comparison

	Ant-Miner (%)	Modified Ant-Miner (%)
Accuracy Rate	94.4	97.6
Misclassification Rate	5.6	2.4

higher than that of the Ant-Miner and the misclassification rate of the modified Ant-Miner is lower than that of the Ant-Miner as shown in Table 5.

6 Conclusion

In this paper, we have modified the Ant-Miner with different heuristic function and pheromone updating method. The modified Ant-Miner was used to discover the classification rules from the breast cancer data set, and the predictive accuracy was compared with the predictive accuracy of Ant-Miner. It is inferred that for the given data set, the modified Ant-Miner works better than Ant-Miner. This classification rule mining problem can be designed as a multi-objective problem and multi-objective ACO algorithm can be used to get the set of Pareto solutions.

References

1. Fayyad, U.M., Piatetsky Shapiro, G., Smyth P.: From data mining to knowledge discovery: an overview. In: *Advances in Knowledge Discovery and Data Mining*, pp. 1–34. AAAI/MIT, Cambridge (1996)
2. Parpinelli, R.S., Lopes, H.S., Frietas, A.A.: Data mining with an Ant Colony Optimization Algorithm. *IEEE Trans. Evol. Comput.* **6**(4), 321–332 (2002)
3. Dorigo, M., Colomi, A., Maniezzo, V.: The Ant System: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B* **26**(1), 29–41 (1996)
4. Jiang, W.J., Xu, Y.H., Xu, Y.S.: A Novel Data Mining Algorithm based on Ant Colony System. In: *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, pp. 18–21 (2005)
5. Liu, B., Abbass, H.A., Mckay, B.: Classification rule discovery with Ant Colony optimization. *IEEE Comput. Intell. Bull.* **3**(1) (2004)