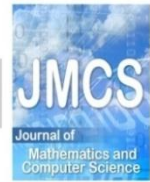




Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: www.tjmcs.com



Designing a new version of Ant-Miner using genetic algorithm

Kayvan Azaryuon¹

¹Department of Computer Engineering, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran
k.azaryoun@yahoo.com

Article history:

Received March : July 2013

Accepted May : September 2013

Available online : April 2014

Abstract

the current article seeks to design and implement a new algorithm for data mining based on ant colony optimization algorithm, which is called Ant-Miner. Ant-Miner extracts classification rules from databases. In our article, we have presented a new version of Ant-Miner which is more efficient than its previous versions. The new version has been dubbed "Ant-Miner 4". We have modified the structure of the heuristic function used in Ant-Miner, implemented it based on the correction function of Laplace, and changed pheromone trail synchronization process in order to enable the redesigned system to produce rules with higher prediction power. In the proposed algorithm, we have tried to employ genetic algorithm to avoid local minimum points, produce a general optimized response, and determine the best values for the parameters. We tested Ant-Miner 4 and Ant-Miner 3 on four data sets, finding out that the new Ant-Miner has a better performance than the older version in terms of the accuracy of the extracted rules.

Keywords: ant colony optimization algorithm, classification rules, data mining, databases.

1.Introduction

Ant-Miner chiefly seeks to extract knowledge from databases. Data mining is a science developed through combining various sciences related to machine learning, statistical concepts and databases. Emphasis should be put on the fact that data mining merely seeks to extract knowledge contrary to

statistical methods. By knowledge we mean the kind of knowledge that is not only accurate but also comprehensive for the user [1] [2]. The data mining process is consisted of different tasks, including classification, regression, independent modeling, clustering, etc. [1].

In this article, we intend to focus on classification, which is aimed at discovering a set of rules from among the training set (a set of records or cases whose class has already been determined) and employing them to predict the class of the test set (a set of records or samples whose class has not been determined yet). We intend to modify Ant-Miner algorithm so as to enable it to produce more accurate rules.

Rule discovery is one of the most important tasks in the data mining process. During this process, a symbolic set of rules are presented to describe each class of family of data [12]. This method of demonstrating knowledge for the user enjoys higher understanding than other methods.

Parpinelli for the first time developed a system called "Ant-Miner 1" based on the ant colony optimization method to extract classification rules from databases [13]. An updated version of Ant-Miner (Ant-Miner 2) presented a new method to calculate the heuristic function. The new version calculated the heuristic function based on the frequency of a variable-value (term) in each class. Ant-Miner 3 provided a new strategy to synchronize trail pheromones [11]. In this article, we intend to use the correction function of Laplace to develop Ant-Miner's heuristic function. This method is capable of discovering rules with higher prediction accuracy without increasing the implementation time of the algorithm. We have dubbed our new algorithm "Ant-Miner 4". We applied our algorithm to four data sets and compared its efficiency with that of Ant-Miner 3, finding out that our Ant-Miner is more accurate than Ant-Miner 3 in terms of the accuracy of the extracted rules.

2. Ant Colony System (ACS)

The ant colony system is a branch of artificial intelligence known as swarm intelligence. Ants are independent insects that work together [6]. They are capable of finding the closest path from the nest to the food without being assisted by a central control system, centralized monitoring or imaging information. Ants communicate with each other indirectly with the help of a substance called "pheromone", sharing information about paths with each other. Each ant encounters different paths when solving a problem. When passing through a path, ants increase the amount of pheromone existing in the path proportional to the path's quality. Therefore, ants are more likely to choose a path with higher amount of pheromone when facing several paths. As a result, ants always search the shortest path to find food. Cooperation and consistence are the main elements observed in the behavior of ants. This system can help develop a metaheuristic method to solve NP optimization problems [4]. Defining the problem in the form of a graph is the main requirement for solving it based on the ant colony method. Then, the ant colony optimization (ACO) algorithm could be employed to find the shortest trail, which is the solution to the problem. The general performance of ants in selecting nodes in the trail depends on a probability function based on equation (1), which will be examined in section 3.

3. Ant-Miner Outline

Ant-Miner is a supervised learning method aimed at extracting classification rules from databases. The supervised learning method is consisted of three stages: 1- selection of a training set, 2- provision of a method for data mining and analysis (learning), and 3- prediction rules synthesis. Training set selection is one of the key stages of the data mining process. The training set should be as small, simple and comprehensive as possible [5] [12]. Ant-Miner output is a set of rules illustrated in figure 1. Figure 2 illustrates outlines of an Ant-Miner.

IF (term1 AND term2 AND ...) THEN <class>

Figure 1- Ant-Miner output

Each term is in the form of Term $ij \equiv A_i = V_{ij}$

A_i : I variable

V_i : I value of A I range.

```
Training set = all training cases;
WHILE (No. of uncovered 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 among all constructed rules;
Remove the cases correctly covered by the selected rule from the training set;
END WHILE
[8]
```

Figure 2- Ant-Miner outline

In Ant-Miner, an ant starts with a blank term (if then) and selects a term in each stage. Put simply, the ant adds a new term to the front part of the rule by selecting a trail. Thus, the ant moves through the graph of

the problem's structure and adds new terms to the current rule by passing by different nodes. Once a rule is completed, its quality is measured in terms of prediction power and is listed temporarily. Then, the second ant starts moving and constructs a new rule based on the amount of pheromone deposited on the trails using function (1). Ant number is one of the parameters that could be adjusted by the user in this stage (repeat loop). Finally, the best rule with the highest quality is selected from among the constructed rules and is added to the list of discovered rules. Each discovered rule covers cases belonging to a certain class. The above-mentioned process is applied to produce rules that will cover cases belonging to other classes as well. This loop (while loop) is repeated and repeated until all cases existing in the training set are covered by the discovered rules. As mentioned before, selection of trails or terms to be added to the rule under construction takes place based on equation (1), the problem dependent heuristic function and the amount of pheromone existing in the trail.

(1)

$$P_{ij} = \frac{\eta_{ij} \cdot \tau_{ij}}{\sum_{i=1}^a \sum_{j=1}^{b_i} (\eta_{ij} \cdot \tau_{ij}(t))}$$

η_{ij} : value of problem dependent heuristic function for term ij

τ_{ij} : amount of pheromone existing in i - j trail or term ij

β : parameter for determining the value of the heuristic function

Ant-Miner calculates the value of η_{ij} for each term ij that could be used in each rule so as to help the ants with the term selection process. This function is developed based on the problem's theoretical structure [3, 9].

In the original version of Ant-Miner, the value of the heuristic function is obtained from equation (2) (entropy).

$$(2) \quad \eta_{ij} = \frac{\text{Log}(K) - \text{Info}T_{ij}}{\sum_i^a \sum_j^{b_i} \text{Log}_2^{(k)} - \text{Info}T_{ij}}$$

$$\text{Info}T_{ij} = - \sum_{w=1}^k \left[\frac{\text{freq}T_{ij}}{|T_{ij}|} \right] \times \text{Log}_2 \left[\frac{\text{freq}T_{ij}}{|T_{ij}|} \right]$$

(3)

K: number of classes

$|T_{ij}|$: Total number of cases in component T_{ij} (a component is consisted of a number of cases, where $A_{ij} = V_{ij}$)

$freq T_{ij}^w$: Total number of term repetition in T_{ij} existing in class W

a: total number of attributes (variables)

bi: total number of attributes belonging to the domain of attribute i.

In the above equations, the more is the value of $InfoT_{ij}$ the lower will be the likelihood that ants choose $term_{ij}$ when constructing a rule, because the value $InfoT_{ij}$ varies from zero to $\log_2 K$. If the value of V_i does not take place for A_i in the training set, the value of $InfoT_{ij}$ will be equal to the highest value or $\log_2 K$. So, $term_{ij}$ enjoys the lowest prediction power. The value of $InfoT_{ij}$ will be zero if all the cases belong to the same class. It means that the possibility of selecting $term_{ij}$ increases. The heuristic function developed by Ant-Miner is similar to the heuristic function of the C4.5 decision tree algorithm [7].

3.1. Pruning the rules

In Ant-Miner, the pruning process takes place after each rule is constructed. Pruning the rules is a common operation in data mining [8] aimed at further simplifying the rules and increasing their prediction power. Pruning takes place when an ant completes construction of a rule. The main idea in the pruning process is that this operation starts with rules consisted of a number of terms. One term of the rule is omitted each time that the pruning process is repeated and the quality of the rule is assessed by equation (4). This process continues as long as the quality of the rule increases by omitting its terms. It is noteworthy that the process might result in the displacement of the class predicted by a rule [10].

(2)

$$Q = \left(\frac{TruePos}{TruePos + FalseNeg} \right) \times \left(\frac{TrueNeg}{FalsePos + TrueNeg} \right)$$

TruePos: number of cases covered by the rule whose class has been correctly predicted by the rule.

FalsePos: number of cases covered by the rule whose class is different from the class predicted by the rule.

FalseNeg: number of cases that have not been covered by the rule, but belong to the class predicted by the rule.

TrueNeg: number of cases that have not been covered by the rule and belong to a class different from the one predicted by the rule.

3.2. Pheromone synchronization

When an ant is done with constructing a rule, the amount of pheromone in that trail is synchronized using equation (5).

$$(3) \quad \tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t).Q \forall i, j \in R$$

To simulate pheromone trail evaporation, the amount of pheromone is decreased based on the number of terms not used during the rule construction process. Equation (6) is employed in this connection [12].

$$\tau_{ij} = \frac{\tau_{ij}}{\sum \tau_{ij}} \forall i, j$$

τ_{ij} : pheromone amount in trails i and j , or for each term ij .

4. Ant-Miner 2 performance evaluation

In Ant-Miner 2, also known as concentration based Ant-Miner [11], the heuristic function is calculated using equation (7).

(4)

$$\eta_{ij} = \frac{\text{majority-class } T_{ij}}{|T_{ij}|}$$

Majority-class T_{ij} : to calculate this value, all the samples whose value is $A_i = V_{ij}$ are separated. Then, the samples are examined to determine to which class most of them belong. Calculation of the heuristic

function through the use of equation (7) improves time complexity of Ant-Miner, but does not affect rule prediction accuracy.

5. Ant-Miner 3 performance evaluation

Ant-Miner 3 presented a new method for pheromone synchronization. In this algorithm, the amount of pheromone increases depending on the number of terms used in the rule construction process based on equation (8). Also, the amount of pheromone is normalized based on the terms that have not been used [12].

(5)

$$\tau_{ij}(t) = (1-p)\tau_{ij}(t-1) + (1 - \frac{1}{1+Q})\tau_{ij}(t-1) \cdot \forall i, j \in R$$

Q: quality of the constructed rule

P: parameter used to determine pheromone evaporation rate.

The value of Q ranges between 0 and 1. The highest value of Q means the highest pheromone increase for the terms that have been used, while the lowest value of Q means pheromone decrease for the terms that have not been used.

The main characteristic of parameter P is that it enables controlling the effectiveness of the passage of time on the rule construction process [12]. The value of this parameter ranges between 0 and 1.

Ant-Miner 3 also offers a new method for selecting terms to be added to the front part of a rule. In the new method, each term is selected based on algorithm (1).

```

if    q ≤ φ
    Loop
        if    qk ≤ ∑j ∈ R Pij    then
            choose termij
        end-Loop
    else
        choos termij With man Pij

```

Algorithm (1)

q_1, q_2 : random numbers

φ : a parameter between 0 and 1

J_i : number of i value for A_j

P_{ij} : calculated in function (1).

Random numbers q_1 and q_2 have been used to create balance between new knowledge and knowledge extracted by previous ants. Therefore, new knowledge has been extracted if $q_1 \leq \varphi$. Otherwise, knowledge extracted by ants in previous stages has been used. This method paves the way to examine paths not covered by previous ants and discover any hidden knowledge existing there.

6. Developing the new algorithm (Ant-Miner 4)

When constructing a rule, the ant starts the process with a blank term. Since each rule is designed to cover the cases of a certain class, selecting a proper $A_i = V_{ij}$ that could appropriately cover the cases of that class is of great importance. In other words, the question remains which $A_i = V_{ij}$ term enjoys high informational value to cover cases belonging to class C ? In Ant-Miner 3, all the terms have equal amounts of pheromone. So, term selection solely depends on the value of the heuristic function for that term in the first step. Since the next ant is more probability to select a trail covered by the previous ant, the heuristic function could play a very important role in directing the ants toward the proper trail.

In Ant-Miner 4, we have used the correction method of Laplace (equation 9) to develop the heuristic function.

(6)

$$\eta_{ij} = \frac{|term_{ij}, k| + 1}{|term_{ij}| + No_of_classes}$$

$|term_{ij}, k|$: determines the number of training cases that include $term_{ij}$ in the current class of K .

$|term_{ij}|$: determines the number of training cases that include $term_{ij}$.

$No_of_classes$: determines the number of the varying values of the class.

Use of the correction method of Laplace paves the way for Ant-Miner 4 to reduce overfitting and manage the subgroups in addition to increasing the accuracy of the rules. Overfitting occurs when a case in the training set is either covered by more than one rule belonging to different classes or is not covered by any rules at all. For example, let's assume that there is a term with a number of rules belonging to a positive class. The number of classes is also two. In this condition, the heuristic function of Ant-Miner 3 (equation 7) measures the informational value of the term in question as $1 / 1 = 100\%$. The value 100% eliminates

the universalizability of the rule constructed by the term in question. Equation (7) does not pay attention to the number of data sets when calculating the informational value of a term, while the method of Laplace calculates the value $(1 + 1) / (1 + 2) = 67\%$ for the term by considering the number of classes. This value is more consistent with realities and enjoys better universalizability.

7. Pheromone synchronization in Ant-Miner 4

As explained before, each rule covers a number of cases given the nature of Ant-Miner. In some cases, the number of positive cases (TP) covered by the rule might be less than the number of negative cases (FP) covered. A rule with such characteristics does not enjoy good prediction powers.

In Ant-Miner 3, the amount of pheromone increases for terms used in the rule construction process proportional to the quality of the rule, something that might lead to the deviation of the next ants and construction of low quality rules in some cases.

In Ant-Miner 4, the amount of pheromone increases for terms used in the rule construction process only after the rule is accepted based on a threshold value. The relation related to the threshold value has been presented in equation (10).

(7)

$$Rule_Threshold = \text{Max}(0.5, \frac{|k|}{|trainingset|})$$

$|k|$: determines the number of positive training sets for the current class.

$\frac{|k|}{|trainingset|}$: determines the total number of the cases belonging to the training sets. Thus, $\frac{|k|}{|trainingset|}$ indicates the probability of predicting cases belonging to class K by a rule. The Max value in this relation guarantees that the constructed rule will have higher prediction accuracy.

For example, let's assume that the value of prediction probability $\frac{|k|}{|trainingset|}$ is 70% for each case belonging to class K in a training set. In that case, the rule will not be accepted because of falling under the threshold value if Ant-Miner discovers a rule with 60% accuracy to predict the cases of class K. this fact corresponds with the reality hidden in the data as well. Also, when prediction probability stands at 10% for class K in a training set (in this case, the threshold value stands at 0.5) and Ant-Miner creates a rule with 15% accuracy to predict cases belonging to class K, this rule is not still accepted because of falling under the threshold value in spite of having a higher prediction accuracy than the prediction value

$\frac{|k|}{|trainingset|}$, because it does not have a proper accuracy rate in total (15%). In Ant-Miner 4, the

amount of pheromone for terms used in the rule construction process increases based on equation (8) only after the rule corresponds with the threshold value.

8. Ant-Miner 4 optimization

In this stage, we intend to optimize Ant-Miner algorithm through the use of genetic algorithm. Falling into the local optimization trap is one of the main problems ahead of ant colony-based algorithms. To avoid this trap, we seek to optimize the research process by properly determining parameters with the help of genetic algorithm. The proposed algorithm is consisted of a population of genetically modified, and sometimes mutated, ants. In this stage, we seek to find the best possible value for parameters β , ρ and q_0 . In the latest version of Ant-Miner, the values of these parameters were fixed and equal for all the ants. In the new method, however, each ant implements a version of the original algorithm with exclusive values. Each genetic algorithm is consisted of different tasks such as cryptography, parent selection, crossover and mutation. For cryptography, a 12-bit string was applied to the three parameters so as to encode each parameter with 4 bits. The numerical range for each parameter stood at -15-0 for β and 0-1 for ρ and q_0 .

Four ants are selected from among all the ants using a competitive function. The selection process is repeated at the beginning of each cycle. After each cycle ends, the selected ants preserve the value of their parameters as well as the value of the fitness function. For crossover, a single-point operator is used and crossover point is determined in a way that the parameters will be regarded as inseparable units. As mentioned before, parameters β and q_0 are used for the next selection process, while parameter ρ is used for local optimization.

9. Comparison between Ant-Miner 3 and Ant-Miner 4

Four data sets presented in table 2 are employed to evaluate the performance of Ant-Miner 4.

Data set	#cases	#categ attrib.	Attrib. #contin.	#classes
Breast cancer	683	-	9	2
Tic-tac-toe	953	9	-	2
dermatology	366	33	1	6
hepatitis	155	13	6	2

Table 2- data sets used in the testing process

These data sets have been recovered from UCI database [14]. Table 2 presents the name of data sets, number of cases in each data set, number of parameters, number of values and number of classes.

Having applied Ant-Miner 3 and Ant-Miner 4 to each defined data set, the performance of each system was evaluated in terms of prediction power. The results have been presented in table 3. It is noteworthy that reference [13] has compared Ant-Miner 1 with CN2 to establish its efficiency, and reference [11] has compared Ant-Miner 3 with Ant-Miner 1 to establish its efficiency. Table 3 shows that the prediction power of Ant-Miner 4 is higher than that of Ant-Miner 3 for two parameters, is equal with it for one parameter, and is lower for one parameter.

Data sets	Prediction power of Ant-Miner 3	Prediction power of Ant-Miner 4
breast cancer	94.32 ± 0.94	95.98 ± 0.21
Tic – tac –toe	76.58 ± 2.53	78.51 ± 0.24
Dermatology	95.05 ± 1.25	94.55 ± 0.45
Hepatitis	91.05 ± 2.35	91.20 ± 0.50

Table 3- comparison between the new version of Ant-Miner and its older version

10. Conclusion

The deductive decision tree is one of the known methods for discovering classification rules in databases. Parpinelli has established that Ant-Miner 1 constructs simpler and more accurate rules compared to the decision tree C 4.5. The current article evaluated Ant-Miner 1, Ant-Miner 2 and Ant-Miner 3 and examined ways of developing a better version of Ant-Miner. The new version increases the accuracy of the extracted rules through adopting the correction function of Laplace and offering a new method for pheromone trail synchronization, something that improves Ant-Miner's capability to extract real word data. Future research on Ant-Miner could focus on subgroup management. Needless to say, a genetic algorithm was employed to configure the parameters and optimize their values in Ant-Miner 4 in order to avoid local optimization and extract rules with higher accuracy.

11. References

- [1] S. M. Weiss and C. A. Kulikowski, *Computer Systems that Learn*, San Francisco, CA: Morgan Kaufmann, 1991.
- [2] A. A. Freitas and S. H. Lavington, *Mining Very Large Databases with Parallel Processing*, London, UK: Kluwer, 1998.
- [3] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, New York, NY: John Wiley & Sons, 1991.
- [4] M. Dorigo, G. Di Caro and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artificial Life*, vol. 5, no. 2, pp. 137-172, 2000.

- [5] Dorigo, M., & Maniezzo, V. "*The ant system: optimization by a colony of cooperating agents*". IEEE Transactions on Systems, Man, and Cybernetics, 26(1), 1-13 1996.
- [6] Ziqiang Wang, Boqin Feng, Classification Rule Mining with an Improved Ant Colony Algorithm, Lecture Notes in Computer Science, Volume 3339, Jan 2004.
- [7] M. Dorigo, A. Coloni and V. Maniezzo, "*The Ant System: optimization by a colony of cooperating agents*," *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, vol. 26, no. 1, pp. 29-41, 1996.
- [8] M. P. Oakes, "Ant Colony Optimization for Stylometry: The Federalist Papers." International Conference on Recent Advances in Soft Computing, November 2004.
- [9] Bo Liu , Hussein A. Abbass , and Bob McKay " *Classification Rule Discovery with Ant Colony Optimization* "IEEE Computational Intelligence .Bulletin February 2004
- [10] Lotti Admane, Karima Benatchba, Mouloud Koudil, Habiba "*Using ant colonies to solve data-mining problems*" DFUAS IEEE International Conference on Systems BP 68M, 16270, Oued Smar, Algérie. 2003.
- [11] Parepinelli, R. S., Lopes, . '*An Ant Colony Algorithm for Classification Rule Discovery*'. In H.A. a. R. S. a. C. Newton (Ed.), Data Mining: Heuristic Approach: Idea Group Publishing. 2002
- [12] Hettich, S. and Bay, S.D. (1999). The UCI KDD Archive, Retrieved September, 2002, from the World Wide Web: <http://kdd.ics.uci.edu>. 2002.
- [13] M. P. Oakes, "Ant Colony Optimisation for Stylometry: The Federalist Papers." International Conference on Recent Advances in Soft Computing, November 2004.
- [14] R. Shakerian, S. H. Kamali, M. Hedayati, M. Alipour , "Comparative Study of **Ant Colony** Optimization and Particle Swarm Optimization for Grid Scheduling", Journal of mathematics and computer Science (**JMCS**) **2011 Issue: 3** **Pages:** 469-474
- [15] Rouhollah Maghsoudi, Arash Ghorbannia Delavar, Somayye Hoseyny, Rahmatollah Asgari, Yaghub Heidari , "Representing the New Model for Improving K-Means **Clustering** Algorithm based on Genetic Algorithm", Journal of mathematics and computer Science (**JMCS**) **2011 Volume: 2** (2011) **Issue: 2** , **Pages:** 329 - 336