Journal of mathematics and computer science 9 (2014), 352-361

Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: www.tjmcs.com

Classification Rule Discovery with Ant Colony Optimization

Kayvan Azaryuon[1] , Babak Fakhar[2], Ali Daghaieghi[3]

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

[2]Department of Computer Engineering, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran
fakharbabak@yahoo.com

[3]Information &Communication Department National Iranian Drilling Company IRAN
a.daghaieghi@nidc.ir

Article history:
Received : August 2013
Accepted : September 2013
Available online : January 2014

Abstract

This paper proposes an algorithm for data mining called Ant-Miner (ant-colony-based data miner). The goal of Ant-Miner is to extract classification rules from data. The algorithm is inspired by both researches on the behavior of real ant colonies and some data mining concepts as well as principles. Recently research shows that ant colony optimization algorithm have been applied successfully to combinatorial optimization problems. In this paper we present an improvement to Ant-Miner. We compare the performance of new algorithm with before algorithm in two public domain data sets.

Keyword : Ant Colony, Classification Rules, Data mining, Extract knowledge, Database.

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]. The general performance of each task is based on searching in large databases, a search that will lead to the discovery of new authentic and significant patterns that could contribute to extracting and illustrating knowledge hidden inside data.

Discovery of rules is one of the most important tasks in the data mining process since it provides a symbolic set of rules for each data category or family [3].

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 [8]. The ant colony method is more flexible and powerful than traditional methods in solving problems. An updated version of Ant-Miner (Ant-Miner 2) presented a new method to calculate the heuristic function. The new version calculates 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]. This article seeks to reveal the setbacks of Ant-Miner 3 and introduce a new version of Ant-Miner.

2.Ant Colony System

The ant colony system is a branch of artificial intelligence known as swarm intelligence. Ants are independent insects that work together. They are capable of finding the closest path from the nest to the food without being assisted by a central control system, centralized monitoring and 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] [3]. Figure 2 illustrates outlines of an Ant-Miner. Ant-Miner output is a set of rules illustrated in figure 1.

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

Figure1- Ant-Miner output

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

A_i: I variable

V_i: I value of A I range.

In Ant-Miner, an ant starts moving 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 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. Trail selection or selection of terms to be added to the rule under construction is done by ants based on equation (1), the problem dependent heuristic function and the amount of pheromone existing in the trail.

```
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]
```

Figure2- Ant-Miner outline

(1)

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

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

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

Ant-Miner calculates the value of τ_{ij} for each $term_{ij}$ that could be used in each rule so as to help the ants to select the terms. This function is developed based on the problem's theoretical structure [6].

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

(2)

$$\eta_{ij} = \frac{Log(K) - InfoT_{ij}}{\sum_{i}^{a}\sum_{j}^{bi} Log_2^{(k)} - InfoT_{ij}}$$

(3)

$$InfoT_{ij} = -\sum_{w=1}^{k}\left[\frac{freqT_{ij}}{|T_{ij}|}\right] \times Log_2\left[\frac{freqT_{ij}}{|T_{ij}|}\right]$$

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)

$freqT_{ij}^{w}$: Total number of term repetition in Tij existing in class W.

a: total number of attributes (variables)

b: 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 termij when constructing a rule, because the value InfoTij 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 termij increases. The heuristic function developed in Ant-Miner is similar to the heuristic function of decision tree algorithm C4.5 [7].

3.1. Pruning rules

In Ant-Miner, the pruning process takes place after each rule is constructed. Pruning the rules is a common operation in data mining [7] [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.

(4)

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

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 rue.

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

In Ant-Miner, equal amounts of pheromone primarily exist in each trail or term. The amount of pheromone is calculated by equation (5) [8].

(5)

$$\tau(t=0) = \frac{1}{\sum_{i=1}^{a} b_i}$$

a: number of attributes (variables)

b_i: number of possible values for a.

When an ant is done with constructing a rule, the amount of pheromone in that trail is synchronized using equation (6).

(6)
$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t).Q \forall i, j \in R$$

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 (7) is employed in this connection [6].

(7)
$$\tau_{ij} = \frac{\tau_{ij}}{\sum \tau_{ij}} \forall i, j$$

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

4.Ant-Miner1 analysis

It is reasonable to see that ants select terms based on equation (1), in which terms are selected based on the amount of pheromone and value of the heuristic function. In this equation, the amount of pheromone changes each time that a term is selected by an ant. This amount increases for the terms used in the rule construction process and decreases for the other terms. At the same time, the value of the heuristic function does not change when a rule is under construction [9]. These conditions lead to the quick convergence of ants into a specific trail, eliminating the capability to extract new knowledge in other trails. In Ant-Miner2, also called concentration-based Ant-Miner [11], the heuristic function is calculated using equation (8).

(8)
$$\eta_{ij} = \frac{majority - classT_{ij}}{\left|T_{ij}\right|}$$

Majority-classTij: to calculate this value, all the samples whose value is A i = Vij 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 (8) improves time complexity of Ant-Miner, but does not change preciseness of rule prediction.

5.Ant-Miner2 analysis

In Ant-Miner 2, the problem related to the original Ant-Miner (quick convergence of ants into a trail) was not solved. In the next version of Ant-Miner dubbed "Ant-Miner 3", a new method was used to synchronize pheromones based on the terms used in the rule construction process. In Ant-Miner 3, the

pheromone synchronization process takes place through the use of equation (9). Thus, the amount of pheromone is normalized based on the terms that have not been used.

(9)

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

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 [9]. The value of this parameter ranges between 0 and 1. In Ant-Miner 3, selection of terms to construct a rule takes place through the use of algorithm 1 in order to solve the problem related to the quick convergence of ants into a trail.

if $q \le \varphi$
  Loop
    if $q_2 \le \sum_{jeji} Pij$ *then*
    choose termij
  end–Loop
else
    choos termij With man Pij

Algorithm 1: Ant-Minier2

q_1,q_2: random numbers

φ: a parameter between 0 and 1

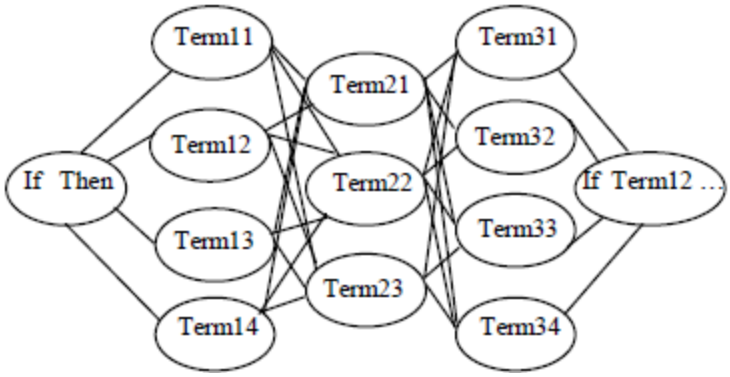
J_i: number of i value for Aj

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

Random numbers q1 and q2 have been used to create balance between new knowledge and knowledge extracted by previous ants. Therefore, new knowledge has been extracted if $q_1 \le \varphi$. Otherwise, knowledge extracted by ants in previous stages has been used.

6.Presentation of a new algorithm and comparison of it with Ant-Miner 3 (our propose)

Use of random numbers presented in Ant-Miner3 might lead to the extraction of new knowledge from the problem, but it also increases the implementation time of the algorithm and results in the discovery of low quality rules. The implementation time of the algorithm increases because ants might move through new trails and discover low quality rules. So, the algorithm has to omit such rules. This process leads to an increase in the implementation time of the algorithm. For example, if there are three attributes (a=3) in a problem, ants will explore the graph illustrated in figure 3 in the worst scenario to construct a rule based on Ant-Miner3.



If $Term_{12}$ and $Term_{22}$ and $Term_{35}$ then class=A

Figure 3- space search graph in Ant-Miner 3

We suggest that in order to optimize selection of terms by ants, all the terms for whom the value of function (1) is between q_1 and q_2 are determined first instead of using random numbers or function (1). Then, one term is randomly selected and added to the rule under construction. q_1 and q_2 are parameters that could be determined and controlled by both Ant-Miner and user. q_1 value is lower than q_2 value, with Pij standing between them. The higher the difference between q_1 and q_2 is the higher will be the probability to extract new knowledge and assess the other terms. Contrarily, the lower is the difference between these parameters the more the new term selection process will be based on the value of function (1). Therefore, ants' movements could be controlled by the user, and the speed and precision of Ant-Miner could be increased. This fact should be taken into consideration that the higher is the difference between q_1 and q_2 the longer will be the implementation time of the algorithm. One characteristic of this method is that it does not select terms based on a fully random basis. It rather determines the main trail through the use of function (1) and randomly selects one term from among the terms that are closer to the function. In this method, the graph of the problem is explored according to figure 4.

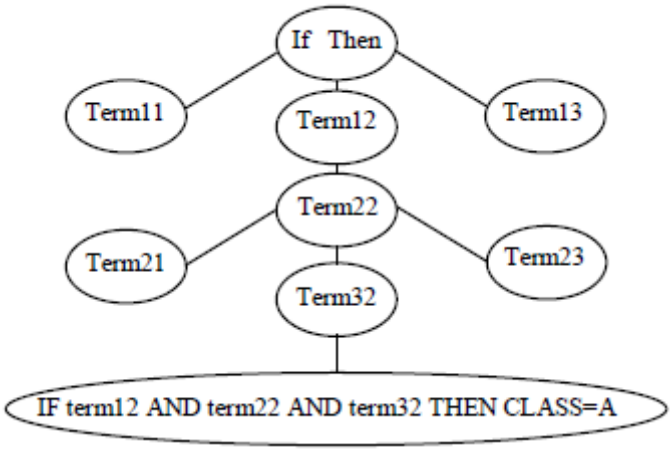


Figure 4- space search graph in the new Ant-Miner

Let's assume that the value of function (1) is the highest value for term13 in figure 2. In this condition, Ant-Miner is more likely to select it to be added to the current rule. But our method considers $term_{12}$ and term11 since they are closer to the value of function (1) and are located between q_1 and q_2 values, selecting one randomly. Finally, the quality of the constructed rule is calculated and the rule is pruned. A comparison between our version of Ant-Miner and Ant-Miner3 shows that the quality of the rule constructed by our Ant-Miner is higher than the one constructed by Ant-Miner3 in some cases. The results of this comparison have been presented in table 1. Also, rule construction takes place quicker in our Ant-Miner compared to Ant-Miner3 since our Ant-Miner allocates a shorter period of time to the pruning process thanks to the smaller size of the rules.

Table (2):
Comparison between classification algorithms

.	Breast Cancer		Tic-tac-toe	
Run Num	new Ant- Miner	Ant- Miner3	new Ant- Miner	Ant- Miner3
1	94.05	94.32	71.84	82.97
2	94.15	93.15	73.12	72.34
3	96.12	91.67	78.81	78.94
4	97.16	97.06	79.88	80
5	96.01	92.75	78.24	72.63
6	94.5	95.65	81.27	80

7.Conclusion

The deductive decision tree is one of the known methods for discovering classification rules in databases. Parpinelli has established that Ant-Miner1 constructed simpler and more accurate rules compared to the decision tree C4.5. The current article assessed Ant-Miner1, Ant-Miner2 and Ant-Miner3, and examined ways of developing a better version of Ant-Miner. The new version enables demonstrating ant behavior in a more controllable way thanks to the parameters mentioned in the article. Knowledge extraction takes

place quicker and more accurately in the new version, something that improves application of Ant-Miner in large systems. Ant-Miners that have been presented so far are capable of discovering a regular list of rules. Therefore, future research on Ant-Miner should focus on the capability to provide an irregular list of rules. An irregular list of rules enables independent interpretation of each rule, making it easier to understand extracted knowledge.

8.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] U. M. Fayyad, G. Piatetsky-Shapiro and P. Smyth, "*From data mining to knowledge discovery: an overview*," In: *Advances in Knowledge Discovery & Data Mining*, U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy (Eds.) Cambridge, MA: AAAI/MIT, pp. 1-34, 1996.

[6] Dorigo, M., & Maniezzo, V. "*The ant system:optimization by a colony of cooperating agent"s*. EEE Transactions on Systems,Man, and Cybernetics, 26(1), 1-13 1996.

[7] Ziqiang Wang, Boqin Feng, Classification Rule Mining with an Improved Ant Colony Algorithm, Lecture Notes in Computer Science, Volume 3339, Jan 2004.

[8] Parepinelli, R. S., Lopes, . '*An Ant Colony Algorithm for Classification Rule Discovery'*. In H. A. a. R. S. a. C. Newton (Ed.), Data Mining

[9] M. P. Oakes, "Ant Colony Optimisation for Stylometry: The Fedaralist Papers." International Conference on Recent Advances in Soft Computing, November 2004.

[10] 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

[11] 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