

# Review on Ant Miners: Algorithms for Classification Rules Extraction using Ant Colony Approach

Safeya Rajpiplawala  
Student (ME CE IV)

Parul Institute of Engg. & Tech.,  
Vadodara, Gujarat (India)

Dheeraj Kumar Singh  
Assistant Professor (IT Dept.)

Parul Institute of Engg.  
& Tech., Vadodara, Gujarat (India)

## ABSTRACT

Mining classification rules from data is a key mission of data mining and is getting great attention in recent years. This paper presents a study of various Ant-miner algorithms for classification Rule extraction and their relative study by reflecting their advantages individually. Ant Colony based algorithms have been successfully implemented in different fields such as remote sensing problems, combinatorial problems, scheduling problems and the quadratic assignment problems. No single algorithm is efficient enough to crack problems from different fields. Hence, in this study some algorithms are presented which can be used according to one's requirement. Modification and extension done to the ant colony based Ant-miner algorithm is discussed here.

## General Terms

Ant Colony Optimization, Classification rules, data mining, Classification

## Keywords

ACO, SI, Rule Classifier, Ant-Miner

## 1. INTRODUCTION

There is a extensive variety of algorithms that have come out from the actions of social bugs. Social insects are typically differentiated by their self union and with the negligible contact or the absence of it. Every social insect separately is superior. They can get information regarding surroundings and cooperate with the remote insects in some way, by Stigmergy. All these features describe Swarm Intelligence. In sections 2, 3, 4, 5, 6, 7, 8 we have presented the Ant\_Miner, cAnt\_Miner, dAnt\_Miner, Parallel Ant\_Miner2, ant miner3, TACO\_Miner and cAntMiner<sub>PB</sub> algorithm for the rule extraction for classification. We have divided dAnt\_Miner2 into two subparts one is dAnt\_Miner known as density based Ant\_Miner2 and other one is Parallel Ant\_Miner2.

## 2. ANT COLONY SYSTEM AND ANT MINER ALGORITHM

Ant colony optimization (ACO) [1] is a stem of a recently developed form of artificial intelligence known as swarm intelligence. Swarm intelligence (SI) is "the property of a structure whereby the combined activities of (simple) agents work together locally with their surroundings cause simple functions overall patterns to appear" [2]. In cluster of insects, which exist in colonies, such as ants a creature cannot do a work on its own; colony's supportive work is the key motivation shaping the intelligent behavior. The majority of

actual ants are blind. Real ants whilst walking arbitrarily leaves a chemical substance known as pheromone [1].

Pheromone magnetizes other ants to stay close to other ants. The pheromone vanishes over time to allow search evaporation. In many experiments presented by Dorigo and Maniezzo give details about the complex behavior of colonies [3], where ants always prefer shortest path.

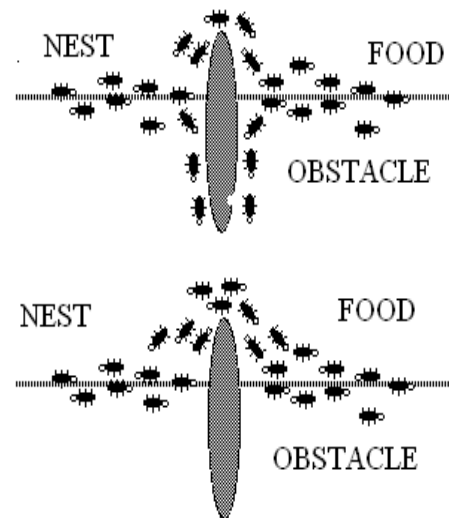


Figure 1. Overview of Dorigo and Maniezzo experiment<sup>[3]</sup>

Further Parpinelli, Lopes and his associates were the earliest to propose Ant Colony Optimization (ACO) for learning classification rules, called Ant-Miner. They discover that an ant-based search is more flexible, robust and optimized than conventional approaches. Their technique employs a heuristic value based on entropy measure. The chief goal of Ant-Miner is to mine classification rules from data [5]. It follows a sequential covering approach to learn a list of classification rules from the given data set. Then rules are included to the list of discovered rules. Then the training cases covered correctly by these rules, are subtracted from the training set. It covers all training cases. Now classification rule is of type:

IF < t1 AND t2 AND...> Then <C>. Where t1=term1, t2=term2, C=Class

### A. Pheromone Initialization

The initial amount of pheromone dropped at each path is inversely proportional to the number of values of all attributes, and is defined by

$$\tau_{Tij}(t = 0) = 1 / \sum_{i=1}^a b_i \quad (1)$$

Where 'a' is the total number of attributes, 'b<sub>i</sub>' is the number of values in the area of attribute i.

### B. Rule creation

Each rule in Ant miner includes a condition part as the ancestor and an expected class. The condition part is a combination of attribute-operator-value tuples. Here we

assume rule condition such as term<sub>ij</sub> B<sub>i</sub> = W<sub>ij</sub>, where B<sub>i</sub> is the *i*th attribute and W<sub>ij</sub> is the *j*th value in the area of B<sub>i</sub>. The likelihood that this clause is included to the current incomplete rule, is given by

$$P_{ij}(t) = \frac{\tau_{Tij}(t) \cdot \eta_{ij}}{\sum_{i=1}^a x_i \sum_{j=1}^{b_i} (\tau_{Tij}(t) \cdot \eta_{ij})} \quad (2)$$

Where  $\eta_{ij}$  is a problem dependent heuristic value for term<sub>ij</sub>.  $\tau_{Tij}$  is the amount of pheromone at present available (at time *t*) on the link between attribute *i*.

### C. Heuristic Value

In Ant Miner, the heuristic value is an information theoretic measure for the excellence of the term to be added to the rule which is measured in terms of the entropy for choosing this term to the others, and is given by the following equations:

$$\eta_{ij} = \frac{\log_2 k - \text{Info } T_{ij}}{\sum_{i=1}^a x_i \sum_{j=1}^{b_i} (\log_2 k - \text{Info } T_{ij})} \quad (3)$$

$$\text{Info } T_{ij} = \sum_{w=1}^k \left[ \frac{\text{Freq } T_{ij}^w}{|T_{ij}|} \right] * \log_2 \left[ \frac{\text{Freq } T_{ij}^w}{|T_{ij}|} \right] \quad (4)$$

Where *k* is the number of classes,  $|T_{ij}|$  is the total number of cases in partition  $T_{ij}$  (partition containing cases where attribute *A<sub>i</sub>* has value  $V_{ij}$ ),  $\text{Freq } T_{ij}^w$  is the number of cases in partition  $T_{ij}$  with class *w*. The high the value of  $\text{info } T_{ij}$ , the less expected that the ant will prefer term<sub>ij</sub>.

### D. Rule Pruning

The rule pruning procedure iteratively removes the term whose elimination will lead the maximum increase in the quality of the rule. The quality of rule is measured using the following equation:

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

Where, *TP* is the numeral of cases covered by the rule and having the similar class that is expected by the rule, *FP* is the numeral of cases covered by the rule and having a class that was not expected by the rule, *FN* is the numeral of cases that are not covered by the rule, whilst having the class that is expected by the rule, *TN* is the numeral of cases that are not covered and which have a different class from the class that is expected by the rule.

### E. Pheromone changing rule

After Rule construction finishes, pheromone are revised as follows:

$$\tau_{Tij}(t+1) = \tau_{Tij}(t) + T \tau_{Tij}(t) \cdot Q ; \forall i, j \in R \quad (6)$$

For replicating the phenomenon of evaporation, the amount of pheromone related with each term<sub>ij</sub> which does not take place in the assembled rule must be reduced. To decrease the pheromone of an unused term, divide the value of each  $\tau_{Tij}$  by the summation of all  $\tau_{Tij}$ .

### Ant Miner Issues and Challenges:

Ant-Miner generates simpler (smaller) rule lists than CN2. It give the impression of being beneficial as it is considerable to reduce the number of discovered rules and rule terms (conditions) in order to increase lucidity of the discovered facts, where it will be exposed to a person as a support for clever decision making.

Guidelines for potential research: First, extend Ant-Miner to cope with continuous attributes, so no requirement of discretization in a preprocessing step. Second, discover the presentation of other kinds of heuristic function and pheromone changing approach.

## 3. CANT MINER ALGORITHM

Ant miner needs the discretization method as a preprocessing and it is appropriate only for the nominal attributes. The drawback of this approach is that less information will be available to the classifier. Generally real-world classification problems are described by nominal (discrete values) and continuous attributes. Fernando, Freitas, and Johnson proposed an extension to Ant-Miner, named cAnt Miner, which was able to handle the continuous values as well [11].

### Purpose:

Ant miner that can handle continuous attributes to extract classification rules from data. Entropy based discretization used during rule construction to create Discrete intervals “on-fly”.

### Method:

Dynamic Discretization of continuous value is done by *Entropy based discretization* method. If nominal attribute, where every term<sub>ij</sub> has the form ( $a_i = v_{ij}$ ), the entropy for the attribute-value pair is computed as in equation (7) – used in the original Ant-Miner:

$$\text{entropy}(a_i = v_{ij}) \equiv \sum_{c=1}^k -p(c | a_i = v_{ij}) \cdot \log_2(p(c | a_i = v_{ij})) \quad (7)$$

where,

$p(c | a_i = v_{ij})$  is the experiential probability of examining class *c* conditional on having observed  $a_i = v_{ij}$ , *k* is the number of classes.

For computing the entropy of nodes representing continuous attributes (term<sub>i</sub>) as these nodes do not represent an attribute-value pair, we need to choose a threshold value *v* to dynamically partition the continuous attribute *a<sub>i</sub>* into two intervals:  $a_i < v$  and  $a_i \geq v$ .

The best threshold value is the value *v* that minimizes the entropy of the partition, given by equation (8):

$$ep_v(a_i) \equiv \frac{|S_{a_i < v}|}{|S|} \cdot \text{entropy}(a_i < v) + \frac{|S_{a_i \geq v}|}{|S|} \cdot \text{entropy}(a_i \geq v) \quad (8)$$

where

$|S_{a_i < v}|$  is the total number of cases in the partition

$a_i < v$  (partition of training cases where the attribute *a<sub>i</sub>* has a value less than *v*),

$|S_{a_i \geq v}|$  is the total number of cases in the partition

$a_i \geq v$  (partition of training cases where the attribute *a<sub>i</sub>* has a value greater or equal to *v*)

$|S|$  is the total number of training cases.

After selecting threshold  $v_{best}$ , the entropy of the term<sub>i</sub> matching to the minimum entropy value of the two partitions and it is defined as:

$$\text{entropy}(term_i) \equiv \min(\text{entropy}(a_i < v_{best}), \text{entropy}(a_i \geq v_{best})) \quad (9)$$

Guidelines for further research: First, examine the performance of special discretisation methods in the rule construction process. Second, evaluate other kinds of pheromone updating methods as dropping pheromone on the edges tends to a customized pruning process.

## 4. DENSITY BASED ANT\_MINER 2

### Purpose:

In this paper [8] a simply calculable density estimation equation was proposed. It resulted from the observation that the ACO algorithm does not require precise information in its heuristic value, as the pheromone should balance for small possible errors in the heuristic values. And instead of complex heuristic function the same work can be done with the simple

one with same accuracy and less computational cost. Here simple heuristic does the job as a complex one.

**Method:**

Heuristic Function:

$$N_{ij} = \frac{\text{majority\_class}_{T_{ij}}}{|T_{ij}|} \quad (10)$$

Where, Majority\_ T<sub>ij</sub> is the majority class in partition T<sub>ij</sub>. The heuristic has less computational cost and was equal to Ant-Miner in terms of its accuracy. In the future, more experiments can be conducted using a range of data sets. Adjustment of system parameters can be done to achieve better results.

**5. PARALLEL ANT\_MINER2**

A flexible parallel ant colony algorithm used for finding classification rule in the huge databases is called as Parallel ant\_Miner2 [12]. In this, rule pruning is done during rule construction [10]. Here, the probability is given by:

$$P_{ij}^p = \frac{\tau_{T_{ij}}^{\alpha}(t) \cdot \eta_{ij}^{\beta} \cdot \lambda_{ip}^{\gamma}}{\sum_{i=1}^a x_i \sum_{j=1}^{b_i} (\tau_{T_{ij}}^{\alpha}(t) \cdot \eta_{ij}^{\beta}(t) \cdot \lambda_{ip}^{\gamma})} \quad (11)$$

Where, λ<sub>ip</sub> is the heuristic factor for the ants in processor p to select values of attribute A<sub>i</sub>. There are two methods proposed to control the ants to renew the pheromone of terms. First method, update the pheromone level for term<sub>ij</sub> of processor p by contact among ants of same group, given by:

$$T_{ijp}(t+1) = (1-\rho) \cdot T_{ijp}(t) + [\rho \cdot \sum T_{ij}(t)] / |X| \quad (12)$$

Where |X| is the number of ants on processor p.  $\sum T_{ijp}(t)$ , is the total of all pheromone formed by all ants on processor p for term<sub>ij</sub>. Second method, Update the pheromone level for term<sub>ij</sub> of processor p by contact with the best of other groups, given by:

$$T_{ijp}(t+1) = (1-\rho) \cdot T_{ijp}(t) + [\rho \cdot \sum_{p=1}^G T_{ij\_best\_p}(t)] / |G| \quad (13)$$

Where |G| is the number of all groups.  $\sum_{p=1}^G T_{ij\_best\_p}(t)$  is sum of all pheromones produced by the all best ant of all processors.

The algorithm finds classification rules quicker with considerably improved accuracy and less redundancy than other leading methods.

The communication techniques used to update pheromones in this algorithm assist ants not to collect inappropriate and redundant terms of the rule [12]. The parallel executions of ants decrease the speed of convergence and as a result make it possible to mine extra new high quality rules by discovering all search space. Other experimental result shows that the parallel Ant\_Miner is more accurate than the other versions of Ant-Miner [10].

**6. ANT MINER 3**

In Ant\_Miner3 [10], a novel equation is proposed for updating pheromone.

**Purpose:**

Choice of term is done using amount of pheromone and heuristic function η which calculates the analytical influence of a term. The pheromone of each term is altered after an ant builds a rule, while η is always the same, so that the next ant tends to opt terms used in the earlier rule, whose pheromone is

increased, and is not likely choose unused terms, whose pheromone is decreased.

As a result, the ants meet to a single constructed rule too quickly.

**Method:**

a) Reduce the ants from revealing inequality toward finding. In order to improve the role of investigation, we apply the transition rule in choosing term<sub>ij</sub>

```

If q1 ≤ φ
  Loop
    If q2 ≤ ∑ Pij
      j ∈ Ji
      Then choose termij
    Endloop
  Else
    Choose termij with max Pij
  
```

where:

q1 and q2 are random numbers

φ is a parameter in [0, 1]

J<sub>i</sub> is the number of i-th attribute values

P<sub>ij</sub> is possibility calculated using equation (2)

(b) The pheromone of unused term is updated by normalization.

$$T_{ij}(t) = (1-\rho) \cdot T_{ij}(t-1) + (1-1/(1-Q)) \cdot T_{ij}(t-1) \quad (14)$$

Where ρ is the pheromone vanishing rates which express how fast the old path vanishes, large value of ρ indicates a fast disappearance and vice versa, q is the quality of the constructed rule, q controls the influence of the history of the current pheromone trail.

**7. TACO\_MINER**

This [13] work proposes an improved algorithm, called Threshold ACO Miner (TACO-Miner). Many classification algorithms are being used to extract links in data such as decision tree, neural networks, statistical and rough sets etc. But they need major skill to work efficiently and do not provide lucid rules. The main goal of this is to provide comprehensible classification rules which have higher predictive accuracy and simpler rule list. Experimentation on data sets from UCI repository is made to compare the performance of TACO-Miner with Ant-Miner, ACO-Miner. The results show that TACO Miner performs better with respect to predictive accuracy and simplicity of the rule list mined with less computational complexity and costs.

**Purpose:**

To overcome Ant miner drawbacks:

1. If rule quality measure Q is very small, evolutionary process will become stagnant.
2. The H (W<sub>j</sub>A<sub>i</sub> = V<sub>ij</sub>) of term<sub>ij</sub> is always the same when computing heuristic function η<sub>ij</sub>, despite of the contents of the rule in which the term occurs. This is not likely in real applications.
3. During the rule construction the weak terms have needlessly been added and the same is pruned later during the pruning phase. This raises the computational complexity and cost of rule construction very much.

**Method:**

```

REPEAT
  Choose the first term
  IF (Information Gain >= 0.6) THEN
    add the term in the empty rule set
  ELSE
    Reject the term from inclusion
  END IF
UNTIL No more term is left
  
```

From study 0.6 is the best value for the selection criteria.

Future Research directions: More experiments to understand the effect of system parameters on the performance of TACO-Miner could be carried out to find out appropriate parameter grouping appropriate for special classification problems. The predictive accuracy and simplicity of the rule list may still be improved by exploring other kinds of heuristic function and pheromone updating rule.

## 8. CANT MINER<sub>pb</sub>

In this [14] a new approach is proposed, which directs the search performed by the algorithm using the quality of a candidate list of rules, instead of a sole rule. The main motivation is to avoid the problem of rule interaction derived from the order in which the rules are discovered i.e., the outcome of a rule affects the rules that can be discovered subsequently since the search space is modified due to the removal of examples covered by previous rules. In the new sequential covering strategy proposed, the pheromone matrix used by the ACO algorithm is extended to include a tour identification that indirectly encodes the sequence in which the rules should be created, allowing a more effective search for the best list of rules.

## 9. SUMMARY

**Table 1. Analysis of Different Ant Miners**

Algorithm	Features
<b>AntMiner</b>	To Extract Classification Rules from Data. To generate simple rule list using ACO procedure in order to increase simplicity of discovered facts to support for intelligent decision making. This Algorithm takes decision while focuses on pheromone update rule.
<b>cAntMiner</b>	Extension of AntMiner deals with Continuous attributes. Entropy based discretization used during rule construction.
<b>AntMiner2</b>	The heuristic value is calculated using the entropy function in ant miner. And instead of complex heuristic function the same work can be done with the simple one with same accuracy and less computational cost.
<b>AntMiner3</b>	Selection of term is made from quantity of pheromone and heuristic function $\eta$ which measures the predictive power of a term. The pheromone of each term is changed after an ant constructs a rule, while $\eta$ 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.
<b>TACO Miner</b>	<b>Ant miner drawbacks:</b> If $Q$ is very small, evolutionary process will become stagnant. The $H(W_j A_i = V_{ij})$ of term $ij$ is always the same when computing heuristic function $\eta_{ij}$ , despite of contents of the rule. During the rule construction the unfit terms have unnecessarily been added and

	the same is pruned later during the pruning phase. This increases the computational complexity and cost of rule construction considerably.
<b>cAntMiner<sub>pb</sub></b>	The main aim is to avoid the problem of rule interaction. The Best List of Rules is discovered by each ant instead of List of Best Rules.

## 10. APPLICATIONS

A predominantly successful research direction in ant algorithms is devoted to their use to discrete optimization problems. The early applications of ACO were in the area of NP-hard combinatorial optimization problems. Ant colony optimization has been used effectively to a large number of complex combinatorial problems such as the traveling salesman problem, the quadratic assignment problem, scheduling problems and routing in telecommunication networks. Above all triumphant example of ACO algorithm in this area is Ant Net. Another instance of interesting research direction is swarm robotics, where the aim is to apply swarm intelligence techniques to manage large groups of cooperating sovereign robots. It can be utilized for solving stochastic, dynamic, multi objective, mixed variable optimization problems.

## 11. CONCLUSION

In this paper, we have briefly summarized all the Ant Miner algorithms used for extracting classification rules from data. Several strategies have been studied for controlling the influence of pheromone and all these methodologies have been demonstrated. This can be a complete source of information to form a basis or starting point for further research to develop innovative algorithms for finding classification rules in which core is ACO procedure. Based on many studies in the literature, some algorithms are more efficient and admired than others. However, to develop the understanding of parameters and effects of each parameter of every system needs a very detailed experimentation. The sole purpose of this paper is to help the researchers to select the one according to their need. Future research will focus on using these algorithms together, such that the strengths and efficiency of these techniques can be increased.

## 12. REFERENCES

- [1] Dorigo Dorigo M.,& Caro,G.D, "Ant Algorithm for Optimatisation",Artificial Life,5(3),137 ,1999.
- [2] Bonabeau, E., Dorigo,M.,& Thera ulaz, G."Swarm Intelligence: From Natural to Artificial System", New York: Oxford University Press,1999.
- [3] 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.
- [4] J.Kennedy and R.Mendes, "Population structure and Particle Swarm Performance", Proceeding of IEEE Congress on Evolutionary Computation, Honolulu, Hawaii USA, 2002.
- [5] R.S.Parnelli, H.S.Lopes and A.A.Freitas, "Data Mining with an Ant Colony Optimization Algorithm", IEEE Trans. On Evolutionary Computation, special issue on Ant colony Algorithm, 6(4), 321-332,2002.
- [6] M. Dorigo and T. Stuetzle., "Ant Colony Optimization", MIT Press, 2004.

- [7] Omkar.S.N., Kumar Manoj, Mley Dipti, “Urban Satellite Image Classification using Biologically Inspired Technique”, in proceedings of IEEE, 2007.
- [8] Bo,L., Abbas, H.A, Kay B, “Classification Rule Discovery With an Ant Colony optimization”, International Conference on Intelligent Agent Technology, IEEE, pp 83-88,2003.
- [9] J.Kennedy,R.C.Eberhart,Y.Shi, “Swarm Intelligence”, san Franciso: Morgan Kaufmann/Academic Press 2001.
- [10] Omid Roozmand and kamran Zamanifar;”Parallel Ant Miner 2 rules”, Springer -Verlag Berlin Heide Iberg ,pp 681-692,2008.
- [11] Fernando E.B. Otero, Alex A. Freitas, and Colin G. Johnson, “cAnt-Miner: An Ant Colony Classification Algorithm to Cope with Continuous Attributes”, Springer-Verlag Berlin, Heidelberg 2008.
- [12] Chen, H., Chen, L, li, T: “Parallel Ant Colony Algorithm for mining Classification Rules”, IEEE Conference on Granular Computing, pp85-90, 2006.
- [13] K .Thangavel.,P. Jaganathan, “Rule Mining with a new Ant Colony optimization Algorithm”, IEEE International Conference on Granular Computing ,pp 135 140,2007.
- [14] F. Otero, A. Freitas, and C. Johnson. “A new sequential covering strategy for inducing classification rules with ant colony algorithms”, To appear in IEEE Transactions on Evolutionary Computation, 2012.