A Study on Ant Colony Optimization with Association Rule

Dr. T. Karthikeyan[1]
Associate Professor, PG Department of Research,
PSG College of Arts & Science, Coimbatore.
t.karthikeyan.gasc@gmail.com

Mr. J. Mohana Sundaram[2]
Assistant Professor, Department of Computer Science
PSG College of Arts & Science, Coimbatore.
majornavy@gmail.com

ABSTRACT—Ant miner is a data mining algorithm based on Ant Colony Optimization (ACO). Ant miner algorithms are mainly for discovery rule for optimization. Ant miner[+] algorithm uses MAX-MIN ant system for discover rules in the database. Soil classification deals with the systematic categorization of soils based on distinguished characteristics as well as criteria. The proposed model delivers to Ant miner and Ant miner[+] algorithm were applied to both training and soil dataset to Association rule and found that Ant miner[+] performs better than Ant miner.

Keywords: Ant Colony Optimization, Ant miner, Ant miner+, Association.

I - INTRODUCTION

Data mining is a promising and relatively new technology and it is defined as a process of discovering hidden valuable and useful knowledge or information by analyzing large amount of data storing in databases or data warehouse by means of different techniques such as machine language, Artificial Intelligence (AI) and statistics. Machine Learning, a branch of Artificial Intelligence and the types of learning are supervised learning, unsupervised learning, semi-supervised learning and Reinforcement learning. One of the supervised learning is classification. Application of data mining includes retail industry, telecommunication industry, biological data analysis, intrusion detection and aerospace to take advantages over their competitors. Large-scale organizations apply various data mining techniques on their data, to extract useful information and patterns. Knowledge discovery in database is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in the data [7]. Some of the data mining tasks are clustering, classification, regression and dependence modeling. An application of data mining includes financial data analysis, retail industry, telecommunication industry, biological data analysis, scientific development and intrusion detection.

Swarm intelligence (SI) is an artificial intelligence technique and it is collective behavior of trustworthy, decentralized, self-organized system [3]. One of the swarm intelligence techniques is Ant Colony Optimization (ACO). Ant Colony Optimization is an meta heuristic algorithm inspired in the cooperative foraging behavior of ants to find and exploit the food source that is nearest to nest. ACO is based on supportive search paradigm that can be applicable to the solution of combinatorial optimization problem.

Ants communicate with each other by means of an indirect form of communication mediated by pheromone [3] [9]. AOC can also be used for the classification task of data mining. The first ACO algorithm for discovering classification rules was Anti-miner and it was proposed by Parpinelli, Lopes and Freitas [11]. Ant-based search is more lithe and robust than traditional approaches. Ant miner uses a heuristic value based on entropy measure.

Classification is a supervised learning. Given a set of predefined categorical classes, determine to which of these classes a specific data item belongs [2] [6]. The task of supervised classification is learning to predict class membership of test case given labeled training case is a familiar machine

learning problem. Classification is unsupervised learning, where training cases are also unlabeled. This type of classification, related r\to clustering, is often very useful in exploratory data analysis, where one has few preconceptions about what structures new data may hold. Some of the classification algorithms are neural networks, k-nearest neighbor classifiers, Support vector machines, Rule based classifier and Fuzzy classification. Examples of classification application include image and pattern recognition, medical diagnosis, loan approval, detecting faults in industry applications and classifying financial market trends.

Swarm intelligence is a division of evolutionary computation, which is the application of methods motivated by the natural world to hard problems in artificial intelligence. Swarm intelligence was introduced by Gerardo Beni and Jing Wang in 1989[3] and is the collective behavior of decentralized, self-organize systems, natural or artificial. SI is mainly based on reliable, robustness and simplicity.

The four bases of self-organization are positive feedback, negative feedback (for counter balance and stabilization), amplification of fluctuations (randomness, errors, random walks) and multiple interactions[3]. Examples are ant colonies, bird flocking, animal herding, bacterial growth, termites and fish schooling. Applications of SI ar Ant-based routing, Crowd simulation. There are two popular swarm-inspired methods in the computational intelligence area Ant Colony Optimization and Particle Swarm Intelligence(PSO). ACO was enthused by the behavior of ants and has lots of applications in discrete optimization problems. PSO is a population based stochastic optimization inspired by the behavior of flocks of birds and schools of fish[3]. Section II presents Ant Colony Optimization. Section III deals with Ant Minor. Section IV discuss about Ant Minor + algorithm Experimental Result are presented in the Section V. Finally Section VI gives the conclusion.

II ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO), which was introduced in the early 1990s as a novel technique to solve hard combinatorial optimization problems[9]. ACO is an optimization algorithm inspired in the collective foraging behavior of ants to find and exploit the food source that is nearest to the nest. ACO is based on cooperative search paradigm that is applicable to the solution of combinatorial optimization problem. Stigmergy, a type of indirect communication mediated by modification of the environment. Communications among individuals or between individuals and the surroundings is based on the use of chemicals formed by the ants called pheromone[3]. The different types of Ant Colony models are Ant system for Traveling Salesman Problem (TSP), Ant Miner, Ant quality, Ant-Cycle, Max-Min ACS, Ant System (AS), ranking model and Ant density. The [11] design the design of an ACO algorithm implies the specification of the following aspects:

- An appropriate manifestation of the problem, which allows the ants of incrementally build solutions through the use of a probabilistic transition rule, based on the amount of pheromone in the trail and on a local problem dependent heuristic

- A method to enforce the construction of valid solutions, i.e., solutions that are legal permissible in the real world situation corresponding to the problem definition:

- A problem dependent heuristic function(η) that measures the significance of items that can be added to the current partial solution;

- A rule for pheromone updating, which specifies how to fine-tune the pheromone trail (τ).

- A probabilistic transition rule based on the value of the heuristic function (η) and on the contents of the pheromone trail (τ) that is used to iteratively construct a solution.

III ANT MINER +

Ant colony based data miner called Ant Miner mainly for discover rules in database. Each artificial path constructed by an ant represents a candidate classification rule of the form [12][14]:

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

Each term is a triple <attribute, operator, value> where value is one of the values belonging to the domain of attribute [11].

The rule antecedent (IF part) contains a set of conditions, usually connected by a logical conjunction operator (AND). Each rule condition as a term, so that the rule antecedent is a

logical conjunction of terms in the form. The rule consequent (THEN part) specifies the class predicted for cases whose predictor attributes satisfy all the conditions specified in the rule antecedent [11].

Algorithm 1 represents an overview of Ant miner algorithm. In Ant-Miner, an artificial ant follows three procedures they rule construction, rule pruning and pheromone updating to provoke a rule from a contemporary training dataset [12]. The artificial ant starts with an empty rule (no attribute terms in rule antecedent), and iteratively adds term to its current partial rule based on the local problem-dependent heuristic function involving information gain and positive feedback involving artificial pheromone level equation(2) & (3) are given below.

When an ant completes its rule and the amount of pheromone in each trial is updated, another ants start to construct its rule, using the new amounts of pheromone to guide its search [14]. The process is continual for at most a predefined number of ants. Then all cases correctly covered by the discovered rule are removed from the training set and iteration is started. Ant miner algorithm is called again and again to find a rule in the training set is less than Max_uncovered_cases the search for rule stops. The discovered rules are stored in an ordered rule list.

Training set = all training cases;
WHILE (No. of uncovered cases in the Training set
> max Uncovered cases)
i=0;
REPEAT
i=i+1;
Ant$_i$ incrementally constructs a classification rule;
Prune the just constructed rule;
Update the pheromone of the trail followed by Anti;
UNTIL (i ≥ No_of_Ants)
Select the best rule among all constructed rules;
Remove the cases correctly covered by the selected rule
from the training set;
END WHILE

Algorithm 1: Overview of Ant Miner

A. Pheromone Initialization

Initially, for all attributes i and their possible values j, a preliminary amount of pheromone is deposited [4]. Pheromone deposited at every path is inversely proportional to the total number of values of all attributes, and is given by the following equation:

$$\tau_{i,j}(t=0) = \frac{1}{\sum\limits_{i=1}^{a} b_i} \qquad (1)$$

Where a is the total number of attributes, b_i is the number of values in the domain of attribute i.

B. Rule Construction

Each rule in Ant-Miner contains a condition part as the antecedent and a predicated class [4][5]. The condition part is a combination of attribute-operator-value tuples [10]. The probability P_{ij} that this condition is added to the current partial rule that the ant is constructing is given by the following equation:

$$P_{ij}(t) = \frac{\tau_{ij}(t).\eta_{ij}}{\sum\sum \tau_{ij}(t).\eta_{ij}}, \forall \, i \in I, \, i=1 \text{ to } a, \, j=b_i \qquad (2)$$

Where η_{ij} is the value of problem-dependent heuristic function for term $_{ij}$ equation (4). The higher the value of η_{ij} the more appropriate for classification the term $_{ij}$ is and so the higher its probability of being chosen. τ_{ij} is the amount of pheromone currently available.

C. Heuristic Function:

The heuristic function (η) is based on the amount of information related with the attribute i and the amount of information is given by equation

$$\text{Info } T_{ij} = -\sum^{k} \left(\frac{\text{Freq } \overline{T_{ij}^w}}{|T_{ij}|} \right) *\log_2 \left(\frac{\text{Freq } \overline{T_{ij}^w}}{|T_{ij}|} \right) \qquad (3)$$

Where K is the number of classes, $|T_{ij}|$ is the total number of cases in partition T_{ij} (partition containing the cases where attribute A_i has value V_{ij}),
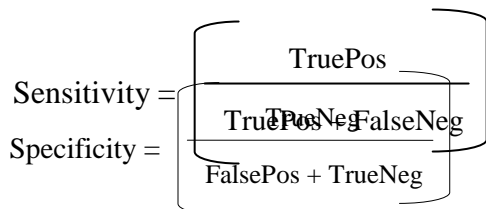
In Ant-Miner, the heuristic value is taken to be an information theoretic measure for the quality of the term to be added to the rule [4][5]. The quality her is measured in terms of the entropy for preferring this term to the others, and is given by the following equations:

$$\eta_{ij} = \frac{\log_2(k) - \text{Info } T_{ij}}{\underset{i=a,\, j=b_i}{\Sigma\,\Sigma}\log_2(k) - \text{Info } T_{ij,}} \quad (4)$$

D. Rule Pruning:

Immediately after the ant completes the construction of a rule, rule pruning is undertaken to increase the lucidity, accuracy of the rule and to avoid overfitting to noisy training data [1]. After the pruning step, the rule may be assigned a different predicted class based on the majority class in the cases covered by the rule antecedent. The rule pruning procedure iteratively removes the term whose removal will cause a maximum increase in the quality of the rule [1][4]. The quality of a rule is measured using the following equation.

Q = Sensitivity X Specificity   (5)

$$\text{Sensitivity} = \frac{\text{TruePos}}{\text{TruePos} + \text{FalseNeg}}$$

$$\text{Specificity} = \frac{\text{TrueNeg}}{\text{FalsePos} + \text{TrueNeg}}$$

Where TruePos is the number of cases covered (enclosed) by the rule and having the same class as that predicted by the rule, FalsePos is the number of cases covered by the rule and having a different class from that predicted by the rule, FalseNeg is the number of cases that are not covered by the rule, while having the class predicted by the rule, TrueNeg is the number of cases that are not covered by the rule which have a different class from the class predicted by the rule [4]. Sensitivity is the accuracy among positive instance and specificity is the accuracy among negative instances.

E. Pheromone Update Rule:

When a rules is constructed by an ant and it is pruned, pheromone updating for a term_{ij} performed based on the following equation.

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t)*Q\text{¥ term }_{ij} \in \text{ the rule} \quad (6)$$

To simulate the phenomenon of pheromone evaporation in real ant colony systems, the amount of pheromone associated with each term_{ij} which does not occur in the constructed rule must be decreased. The cutback of pheromone of an unused term is performed by dividing the value of each τ_{ij} by the summation of all τ_{ij}.

IV. ANT MINER + ALGORITHM

Ant miner+, which applies MAXMIN ant system (MMAS) and its different from a normal ant system in three ways:

- After each iteration only the beset ant is allowed to add pheromone to its trail. This allows for a better exploitation of the best solution found.

- To avoid stagnation of the search, the range of possible pheromone trails is limited to an interval maximum and minimum $[\tau_{min}, \tau_{max}]$.

Construction graph

WHILE (not early stopping)
  Initialize heuristics, pheromones and probabilities of edges
   WHILE (not converged)
    Create ants
    Let ants run from source to link
     Evaporate pheromone on edges
     Prune rule of best ant
     Update path of best ant
     Adjust pheromone levels if outside boundaries kill ants
     Update probabilities of edges
   END

Extract rule
 Flag data points covered by the extracted rule
END
Evaluate performance on test set.   (8)

<center>**Algorithm 2: Overview of Ant Miner+**</center>

The algorithm 2 represents the overview of Ant miner+ algorithm Max-Min ant system approach additionally requires the pheromone levels to lie within a minimum and maximum interval. Convergence occurs when all the edges of one path have a higher pheromone level then the pheromone level of all others edges [9]. Next the rule corresponding with the path is extracted and the training data covered by this rule is removed from the training set. The iterative process will be repeated until all ants visited the path.

Ant moving from one node to another is probabilistic based on the pheromone amount on the edge between the two nodes and the value calculated by a heuristic function equation (8) applied on the destination node. After a rule is constructed, it is evaluated against the training set and the pheromone on the rule edges I updated based on the quality of the rule equation (5). In each iteration the best rules generated are selected and added to the result rule set, the covered cases in the training set by the rules are removed and the pheromone in the construction graph is reset. The algorithm runs until all the ants finished their search.

A. Pheromone Initialization

Initially the maximum pheromone is deposited at each path.

$$\tau_{i,j}(t=0) = \tau_{max} \qquad (7)$$

B. Heuristic Function

The heuristic function (η) is based on the amount of information associated with the attribute i and j is

$$\eta_{ij=} \left| \frac{T_{ij} \& CLASS = class_{ant}}{T_{ij}} \right|$$

C. Pheromone Update Rule

When rules are constructed by an ant and it is pruned, pheromone updating for a $term_{ij}$ is performed based on the equation (9) and quality (Q) of the rule is based on the equation (5). In [5] quality of the rule is based on the sum of confidence and coverage but in the paper sensitivity and specificity are used for calculating the quality of the rule.

$$\zeta(V_{i,j}, V_{i+1})(t+1) = \rho \, \zeta(V_{i,j}, V_{i+1})(t+1) + Q/10 \quad (9)$$

V EXPERIMENTAL RESULTS

The performance evaluations are evaluated using 4 training set Iris, Glass, Wine, Soil and real soil dataset based on the accuracy and execution time. In the experiment training data sets are taken from the UCI repository. Irish dataset which contains 150 instances, 4 attribute and 3 classes (Iris Setosa, Iris Versicolour, Iris Virginica), wine dataset which contains 178 instances, 13 attribute and 3 classes, soil database contains 6435 instances, 36 attribute and 6 classes (red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble, very damp grey soil) and glass database contains 214 instances, 10 attribute and 7 classes.

Real data for soil has been collected and preprocessed using the data mining approach and the data are converted into CSV format for further processing. Real soil dataset contains 125 instances, 2 attributes and 4 classes.

Ant miner and Ant miner+ algorithm has been implemented using MATLAB R2010a. Parameters for ant is set as 100, 1000 and accuracy, execution time were calculated for the two algorithms for 5 iteration. Accuracy and execution time for two algorithms applied to different data set are shown in Table1.

Algorit	Data	100 Ants		1000 Ants	

hm	set	Accuracy	Execution Time	Accuracy	Execution Time
Ant Miner	Iris	57.4066	0.0407	57.6767	0.0617
	Soil	78.1603	0.4993	78.1706	0.462
	Wine	52.5358	0.0622	53.6767	0.0731
	Glass	50.094	0.08	50.094	0.0887
Ant Miner+	Iris	76.2057	0.0613	66.3077	0.6565
	Soil	79.7677	0.5	79.8468	0.5122
	Wine	92.0593	0.0742	92.0593	0.0773
	Glass	86.5157	1.0043	86.5134	0.2364

Table: 1 Accuracy and Execution time for Two algorithm
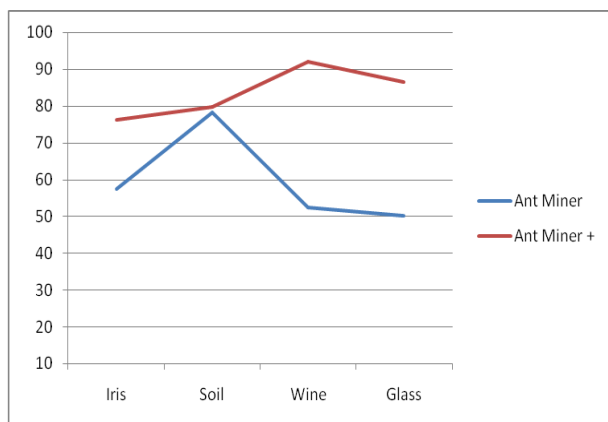

Figure: 1 Accuracy for two algorithms on 4 training Dataset
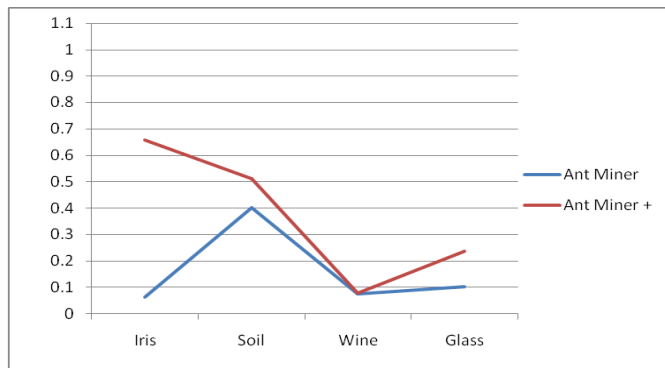

Figure: 2 Execution time for algorithms on 4 Training dataset

TheFigure 1 and 2 represents Accuracy graph and Execution Time graph for Ant Miner and Ant Miner + algorithm on Irish, Soil, Wine and Glass datasets.
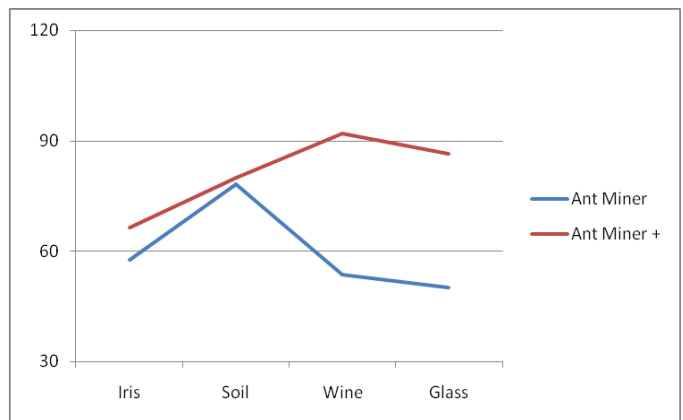

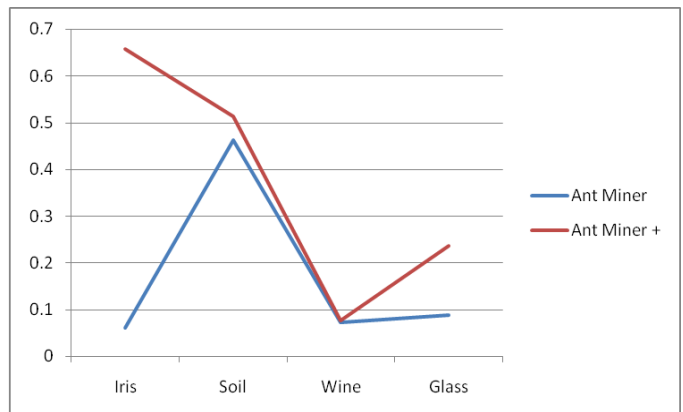Figure: 3 Accuracy Graph for 100 Ants on 4 training Dataset

Figure: 4 Accuracy time graph for 1000Ants on 4 training Dataset

The figure 3 and 4 show the Accuracy and Execution Time of the algorithms with 1000 ants. Both the algorithms were tested using various training datasets (Irish, Soil, Wine and Class). Increasing of ants slightly improves the accuracy and execution time.

VI CONCLUSION AND FUTURE RESEARCH

In this research Ant miner and Ant miner+ algorithms were experimented with different training datasets. The same have been applied for real dataset which were rendezvous in and

around the erode district and preprocessed for this proposed work.

This proposed research classifies the real soil data set that helps to know the best soil based on the PH and the Electrical conductivity for fertilization and to improve the quality of the soil.

The experimental value shows that Ant miner+ performs better than Ant miner based on Accuracy and Execution time. In future, the proposed work may be combined with Neural and Genetic Algorithms and can also be applied for the unsupervised learning.

REFERENCES

[1]   Adel Ardalan, Prof.Rahgozar,”***Ant Miner:Ant Colony-Based Association Rule Miner***”,spring 2006

[2]   B.Baesens,T.Van Gestel,S.Viaene,M.Stepanova,J.A.K.Suykens,and J.Vanthienen,”***Benchmarking state-of-the-art Classification algorithms for credit scoring***”,J.Oper.Res.Soc., vol.54.no.6,pp.627-635,2003

[3]   Bonabeau, E.,Dorigo,M., & Theraulaz, G.(1999),”***Swarm Intelligence: from Natural to Artificial Systems***”,New York;Oxford University Press.

[4]   Bo Liu,Hussein A. Abbass and Bob Mckay, “***Classification Rule Discovery with Ant Colony Optimization***”,IEEE Computational Intelligence Bulletin,February 2004,vol.3 No.1.

[5]   David Martens,Manu De Backer,Raf haesen,Student Member,IEEE, Jan Vanthienen,Monique Snoeck,and bart Baesens,”***Classification with Ant Colony Optimization***”,IEEE transactions on evolutionary computation,vol.11,no.5,October 2007.

[6]   D.J.Hand and S.D. ***Jacka,Discrimination and Classification*** New York:Wiley.1981.

[7]   Jiawei Han and Micheline Kamber, “Data Mining Concepts and Techniques”, Edition 2006 and ISBN: 978-1-55860-901-3

[8]   Loannis Michelakos, Nikolaos Mallios, Elpiniki Papageorgion and Michael Vassilakopoulos, “Ant Colony Optimization and Data Mining:Techniques and Trends”, 2010

[9]   Marco Dorigo and Thomas Stutzle, “Ant Colony Optimization”, Edition 2004 and ISBN-81-203-2684-9.

[10]   Michael Goebel and Le Gruenwald, “A Survey of Data Mining and knowledge Discovery Software Tools”, June 1999.

[11]   Parepinelli, R.S., Lopes, H.S., & Freitas, A., “Data Mining with an Ant Colony Optimization algorithm”, IEEE transactions on evolutionary computation, vol. 6, no. 4, August 2002

[12]   Parepinelli, R.S., Lopes, H.S., & Freitas, A., “Data Mining with an Ant Colony Optimization algorithm”.

[13]   Parepinelli, R.S., Lopes, H.S., & Freitas, A.(2002). “An Ant Colony Algorithm for Classification Rule Discovery”

[14]   Parepinelli, R.S., Lopes, H.S., & Freitas, A, “An Ant Colony Based System for Data Mining: Applications to Medical Data”.