Journal of Computational Intelligence in Bioinformatics.
ISSN: 0973-385X Volume 1 Number 2-3 (2008) pp. 129–140

http://www.ripublication.com/jcib.htm

Discovering Unordered Rule Sets for Mixed Variables Using Ant Colony Optimization

C. Nalini[1*] and P. Balasubramanie[2]

*[*1]Department of Information Technology,*
Kongu Engineering College, Erode, India
Email: nalinikec@gmail.com
[2]Dept. of Computer Science & Engineering,
Kongu Engineering College, Erode, India

Abstract

Ant colony optimization (ACO) algorithms have been applied successfully to combinatorial optimization problems. Ant colony optimization (ACO) can be applied to the data mining field to extract rule-based classifier. The Ant-Miner algorithm was first proposed by Parpinelli and his colleagues (2002), applies an Ant Colony Optimization (ACO) heuristic to the classification task of data mining to discover an ordered list of classification rules. Ant-Miner uses a discretization process to deal with continuous attributes in the data set. The transition rule and pheromone update strategy are the more problem independent features of Ant-Miner algorithm. In Ant –Miner algorithm a term had stochastically chosen and added to the current partial rule. This paper presents an enhanced Ant Miner, which includes two main contributions. Firstly, use Laplace-corrected confidence heuristic function to generate unordered rule sets for mixed variables and combine both stochastic selection and deterministic selection to include a term in the current partial rule to improve the predictive accuracy of a discovered rule. Secondly, it handles both continuous and discrete attributes without discretization using multimodal functions and produces a dynamic balance between diversification and intensification during search process. The result of the algorithm had compared against the original Ant-Miner algorithm in six public-domain datasets with respect to predictive accuracy and simplicity of the discovered rules. Empirical results show that the enhanced Ant-Miner can quickly discover better classification rules, which have roughly competitive predicative accuracy, and short rules.

Keywords: Data mining, Classification, Unordered rule set, Ant Colony Optimization, Laplace-corrected confidence, Predictive accuracy.

Introduction

Data mining is a multi-disciplinary field which aims to extract knowledge from databases [14]. The data mining task addressed in this paper is the classification task, where the goal is to predict the class of an example, given the values of a set of attributes for that example. In essence, the classification task consists of inducing a model from the data by observing relationships between predictor attributes and classes, which can be used later to classify new examples. The discovered knowledge is often represented in the form of IF (conditions) THEN (class). In the context of discovering classification rules in data mining, Ant Colony Optimization (ACO) [3] algorithms have been successfully applied to different problems [4]. Ant-Miner [7], [8], the first implementation of an ACO algorithm for the classification task of data mining, has been shown to be competitive with the well-known C4.5 [9] and CN2 [1] classification-rule discovery algorithms. Although real-world classification problems are often described by nominal (discrete values) and continuous (real-valued)attributes, Ant-Miner has the limitation of being able to cope only with nominal attributes in its rule construction process. In order to overcome this limitation, it discretize continuous attributes into nominal attributes in a preprocessing step. A potential disadvantage of this approach is that less information will be available to the classifier which can have a negative impact on the accuracy of the discovered knowledge. This paper proposes an extension to Ant-Miner, which incorporates an entropy-based discretization method in order to cope with continuous attributes during the rule construction process and it adds a new transition rule to create a balance between exploitation and exploration during search process to improve the accuracy of the algorithm and generate simple rules.

The remainder of the paper is organized as follows. Section 2 presents an outline of the Ant-Miner algorithm. Section 3 gives the related Work Section 4 explains the proposed algorithm. Section 5 discusses computational results and performance of the algorithm. Section 6 concludes the paper and suggests further areas of research.

A Brief Description of The Ant-Miner Algorithm

Training set = all training cases;
WHILE (No. of 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 Ant_i;
        UNTIL (i = No_of_Ants) or (Ant_i 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

Ant-Miner discovers an ordered list of classification rules based on a heuristic function involving information gain a popular heuristic function in data mining [9] and positive feedback involving artificial pheromone. For each iteration of the Repeat-Until loop, an ant attempts to discover a rule by selecting terms in a probabilistic manner, until all the attributes have been used to make the current rule, or adding any other available term would make the rule coverage less than min_cases_per_rule a user-specified threshold. The discovered rule is then pruned in an attempt to reduce over fitting to the training data and increase rule quality. Afterwards, the pheromone values for the terms in the current rule are increased, in order to increase the probability that other ants will select those terms, and then the pheromone values for all terms are normalized. The While loop iterates until the number of training examples remaining in the dataset becomes less than or equal to Max_uncovered_cases another user-specified threshold. The rule discovered in the Repeat-Until loop that has the highest quality is then added to the list of discovered rules, and the training examples correctly covered by that rule are removed from the training dataset. An example is correctly covered by a rule if the example satisfies the rule antecedent and has the class predicted by the rule.

Related Work on Ant-Miner Variations

Following the introduction of Ant-Miner, several variations were proposed [4].They involve different pruning and pheromone update procedures, new rule quality measures and heuristic functions, discovering fuzzy classification rules and discovering rules for multi-label classification problems.Chan & Freitas [2] have proposed a new rule pruning procedure for Ant-Miner. They have observed that the original Ant-Miner's pruning procedure processing time increases significantly with a large increase in the number of attributes, which affects the scalability of the method. To overcome this limitation, it was proposed a new prune procedure that led to the discovery of simpler(shorter) rules and improved the computational time in datasets with a large number of attributes.Martens et al. [5] have introduced a new classification algorithm, named AntMiner+, based on Ant-Miner. It differs from the original Ant-Miner implementation in several aspects. It employs different pheromone initialization and update procedures based on the MAX −MIN ant system (MMAS) [13]. Swaminathan [12] proposed an extension to Ant-Miner which enables interval conditions in the rules. While it still uses a discretization method to define discrete intervals for continuous attributes in a preprocessing step, the continuous values are not replaced in the dataset. For each discrete interval, a node is added to the construction graph and the pheromone value associated to the node is calculated using a mixed kernel probability density function (PDF).The above work generate ordered rule set and use entropy based heuristic functions.Nalini et al.[6] proposed an extension to Ant-miner which enables interval conditions in the rules and generate an unordered rule set based on Laplace-corrected confidence value, The balance between diversification and intensification is important to provide a good quality solutions. Despite the Ant-Miner variations proposed in the literature, to the best of our knowledge, extending Ant-Miner to produce a good quality solutions the

algorithm use two transition rules to create a dynamic balance between diversification and intensification during search process and use Rule Confidence Threshold value to measure the quality of a rule.

Proposed algorithm

RuleSet = [ ] // initialized as empty set
FOR EACH Class
TrainingSet = {all training cases}
PositiveSet = {training cases of current class}
NegativeSet = TrainingSet – PositiveSet
Calculate the mean and the standard deviations of each range
Initialize the pheromones; in case of continuous attributes pheromone is a mixed kernel PDF with each range representing a PDF with the corresponding mean and standard deviation
WHILE (|PositiveSet| > max_uncovered_cases)
t = 1;
j = 1;
REPEAT // iteration for constructing a rule
Ant_t starts with an empty rule and incrementally constructs a classification rule R_t by adding one term at a time to the current rule;
Prune rule R_t;
IF(LaplaceCorrectedConfidence(R_t)>ruleConfidenceThreshold
THEN increase pheromone of terms in rule R_t, in case of a continuous attribute add a PDF corresponding to the range
END IF
Update pheromones in all other terms by normalizing the pheromone values (simulating evaporation)
IF (R_t equals R_t -1) **THEN** j = j + 1;
ELSE j = 1;
END IF
t = t+1;
UNTIL (t = No_of_ants) OR (j = No_rules_converge)
Select the best rule among all constructed rules;
Add the best rule to Rule Set ;
TrainingSet = TrainingSet – {set of positive cases covered by R_{best}};
PositiveSet = PositiveSet – {set of positive cases covered by R_{best}};
END WHILE
Output Rule Set;
END FOR

In Unordered Rule Set Ant-Miner the consequent for the rule is known by the ant during rule construction at the initial stage and leads to faster convergence on good rules in comparison with the original Ant-Miner. The reason for this is that in

Unordered Rule Set Ant-Miner [13]each term's pheromone value directly represents that term's relevance for predicting a fixed target class value. So, an extra For-Each loop is added as the outer loop of the algorithm, iterating over the values in the class attribute domain. Each iteration of the For-Each loop discovers an unordered set of rules, all of which predict the current class value. The entire training set is reinstated in the beginning of each iteration, so that a maximal number of negative examples are available to the algorithm. Ants discover rules from the training data until the number of positive examples (belonging to the current class) remaining in the dataset that have not been covered by a discovered rule is less than or equal to the value determined by the *max_uncovered_cases* parameter.

Probability Transition rule

To produce a dynamic balance between diversification and intensification we used the following two transition rules are used. 1. A random proportional transition rule:

$$P_{ij} = \frac{\eta_{ij.}\ \tau_{ij}(t)}{\sum_{i=1}^{a} x_i \cdot \sum_{j=1}^{b_i} (\eta_{ij.}\ \tau ij(t))} \qquad \text{Eq.1}$$

where:

- Pij is the probability that termij is selected to be added to the current partial rule antecedent
- η_{ij} is the heuristic value associated with termij
- $\tau_{ij}(t)$ is the amount of pheromone associated with a termij at iteration t
- a is the total number of attributes
- bi is the number of domain values of the i-th attribute
- α and β are two adjustable parameters that control the relative weight of the heuristic and pheromone values respectively. Hereby giving them equal importance (i.e) α and β=1

2.The pseudo-random proportional transition rule :

$$\mathbf{S=\{}\begin{cases} \operatorname{argmax}_J\{[\eta_{ij.}]^{\alpha}.[\tau_{ij}(t)]^{\beta}\} & \text{if } q \le q_0 \\ \mathbf{\}} & \text{if } q > q_0 \end{cases} \qquad \text{Eq.2}$$

An ant choose a term to add to the rule antecedent depends on q, which is a random variable uniformly distributed over [0,1], and q0 an user defined parameter with range [0,1]. If q ≤q0 then the term with the highest proportion of heuristic and pheromone value is selected from the terms that may still be considered for inclusion. Otherwise, it selects the term using Eq.1. Therefore, q ≤q0 corresponds to an exploitation of the knowledge available about the problem, whereas q > q0 contributes towards more exploration. Limiting exploration by adjusting q0 allows ants to concentrate on best solutions instead of exploring constantly.

Heuristic function

The problem dependent heuristic function chosen is the Laplace-corrected confidence [3] for each term. It generates rules that are specific and helpful in reducing over fitting.

$$\eta_{ij} = \frac{|term_{ij}, k| + 1}{|term_{ij}| + no\_of\_classes} \quad \text{Eq.3}$$

where $|term_{ij}, k|$ is the number of training cases having $term_{ij}$ in the current positive class k; $|term_{ij}|$ is the number of training cases having $term_{ij}$; and No_of_classes is the number of values in the class attribute's domain.

Pheromone initialization

The initial pheromone levels for both nominal and continuous attributes must be set at the initialization step. In case of a nominal attribute, the pheromone level for each attribute value is initialized to 1/n, where n is the total number of attribute values for all of the attributes including the continuous attributes. For continuous attributes, first we split the values into ranges and then find mean and standard deviation of the corresponding range and use the Probability Density Function (PDF).

$$\tau_{ij} = \frac{\sum_{i=l}^{h} G(x)}{K_i} \quad \text{Eq.4}$$

where

- τ_{ij} = pheromone level of the jth range of the continuous attribute at the ith position
- k_j = normalization constant of the jth range
- *G(X)* = mixed kernel PDF for the continuous attribute, the area of the curve for G(x) between the ranges l and h
- l = the minimum value of the ith range
- *h* = the maximum value of the ith range

Achieving a uniform distribution over the continuous domain is not possible [7], and hence the attributes are initialized to different values. With the use of the normalization constant 'k_i,' the initial pheromone values of the continuous ranges can be made equal to that of other attribute values (1/n). The normalization constant 'k_i' is given by

$$k_i = \sum_{i=l}^{h} G(x) * n \quad \text{Eq.5}$$

where

- k_i =normalization constant for the ith range of a continuous attribute

- *n*= Total no of attribute values for all attributes including continuous attributes

The normalization constant is calculated at the beginning of the run when the pheromone levels are initialized.

Pheromone Updation

Each time an ant completes the construction of a rule, the amount of pheromone for all terms is updated. The pheromone of terms that occur in the constructed rule R is increased in proportion to the quality (Q) of the rule. The best rule of the iteration is stored and used to update the pheromone levels of the terms before the next iteration initiate The rule confidence threshold(Eq.6) determines if a rule is acceptable or not (i.e., whether or not the pheromone of its terms should be increased) is expressed by the following formula.

$$\text{Rule Confidence Threshold} = \frac{\text{MAX}(0.4, |k|)}{|\text{training set}|} \qquad \text{Eq.6}$$

where |k| is the number of training cases with the current (positive) class, and |training set| is the total number of cases in the current training set. The rule confidence threshold is therefore the maximum of the relative frequency of the predicted class and user defined threshold value (0.4).Here 0.4 is not an optimized value,by trial we fixed this value. Further research is needed to optimize the value. The use of the max operator guarantees that, when the predicted class has a low relative frequency, the confidence threshold is raised to 40%.Once a rule has been considered acceptable, increase the amount of pheromone to each of the terms appeared in that rule . For nominal attributes, the pheromone level is increased by using the formula

$$\tau_{ij}(t+1) = \tau_{ij}(t) + (\tau_{ij}(t) * Q) \qquad \text{Eq.7}$$

where $\tau_{ij}(t)$ is the current (at time index *t*) amount of pheromone associated with $term_{ij}$, and Q is a quality function calculated by using Eq.9, otherwise set to 0.

$$Q = \text{Specificity} * \text{precision} \qquad \text{Eq.8}$$

For continuous attributes, a new normal kernel added to the mixed kernel PDF. The mean and standard deviation of the added kernel are the mean and standard deviation of the range that selected by the ant during the iteration. The addition of a new kernel changes the shape of the mixed kernel PDF curve and thus changes the values of the pheromones for the neighboring ranges. The impact of this increase in pheromones on the pheromone value of a neighboring range decreases as the distance between the means of the added range and the neighboring range increases. This increase in pheromones of the neighboring ranges preserves the relationship represented by the continuous values.

Rule Pruning

Irrelevant terms were removed from rule during rule pruning process to improve the predictive accuracy of the rule, This involved speculatively removing each term in turn, evaluating the quality of the rule (Eq.8) without that term, and then analyzing the change in rule quality. This process was repeated until there was only one term left in

the rule antecedent or no increase in rule quality was observed during the speculative removal process.

Computational Results and Discussion

In order to evaluate the proposed algorithm, we have selected six datasets from the UCI Irvine machine learning repository [15] which had at least one continuous attribute. Table 1 shows a summary of the selected datasets.All experiments were conducted running a well-known 10-fold cross-validation procedure [1]. We have compared the performance of proposed against Ant-Miner, with respect to predictive accuracy and simplicity of the discovered rule lists. The algorithm has four user-defined parameters as defined in Table 2. In Unordered Rule Set Ant-Miner, Max_uncovered_cases refers to the maximum number of uncovered positive cases in the training set, while in the original Ant-Miner it refers to the maximum number of cases (either positive or negative ones) in the training set. For this reason, this parameter may need to be set lower in unordered rule set Ant-Miner than in the original Ant-Miner.

Table 1: Dataset Characteristics

Data Set	Total No. of Examples	Nominal Attributes	Continuous Attributes	Classes
Ljubljana breast cancer	286	9	0	2
Wisconsin breast cancer	699	0	9	2
Tic-tac-toe	958	9	0	2
Dermatology	366	33	1	6
Hepatitis	155	13	6	2
Cleveland heart disease	303	8	5	5

Table 2: Parameter settings

Parameter	Ant-Miner	Unordered Rule Set Ant-Miner
No_of_ants	3000	3000
Min_cases_per_rule	5	5
Max_uncovered_cases	10	5
No_rules_converg	5	5

Table 3 and Figure 1 summarize the results comparing the predictive accuracy of Ant-Miner and Proposed algorithm. Each entry in the table shows the average value of the accuracy obtained via the cross-validation procedure followed by the standard deviation. An entry in the proposed algorithm column is shown in bold if, for the corresponding dataset, the accuracy achieved with proposed algorithm was significantly greater than the accuracy achieved with Ant-Miner for that dataset according to a two-tailed Student's t-test with significance-level = 5%. In two datasets, namely hepatitis and Cleveland heart disease, proposed algorithm was significantly more accurate than Ant-Miner.

Table 3: Predictive accuracy.

Data set	Ant-Miner	Proposed algorithm
Ljubljana breast cancer	75.28+/-2.24	79.90+/-1.93
Wisconsin breast cancer	96.04+/-0.93	97.26+/-1.05
Tic-tac-toe	73.04+/-2.53	75.01+/-1.85
Dermatology	94.29+/-1.20	95.64+/-1.44
Hepatitis	90.00+/-3.11	**97.68+/-2.84**
Cleveland heart disease	59.67+/-2.50	**81.8+/-2.59**

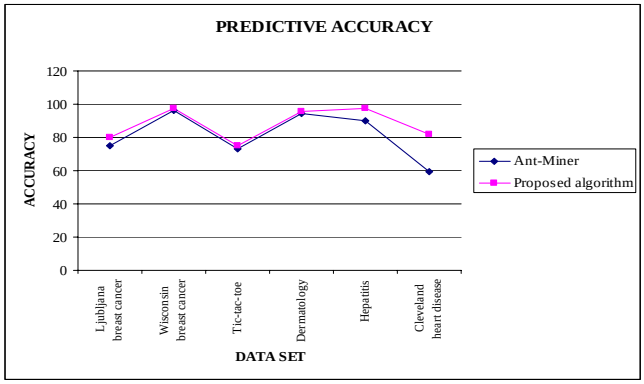

Figure 1: Predictive accuracy

Table 4 and 5, Figure 2 and 3 summarize the results concerning the simplicity of the discovered rule lists, measured by the total number of terms (conditions) in all discovered rules. Each entry in the table shows the average rule list size obtained via cross validation procedure followed by the standard deviation. An entry in the proposed algorithm column is shown in bold if, for the corresponding dataset, the rule list discovered by the proposed algorithm was significantly simpler than the rule list discovered by Ant-Miner for that dataset according to a two-tailed Student's t-test with significance-level= 5%. Concerning the simplicity of the discovered rule lists, the proposed algorithm discovered significantly simpler rule lists than Ant-Miner in four out of six data sets .In the proposed algorithm we gave equal importance to diversification and intensification so the algorithm generated more accurate and simpler rules than Ant-miner algorithm.

Table 4: Average no of rules

Data set	Ant-Miner	Proposed algorithm
Ljubljana breast cancer	7.1+/-0.31	**7.5+/-0.25**
Wisconsin breast cancer	6.2+/-0.25	**7.97+/-0.06**
Tic-tac-toe	8.5+/-0.62	7.16+/-0.25
Dermatology	7.3+/-0.15	**7.79+/-0.03**
Hepatitis	3.4+/-0.16	**5.67+/-0.14**
Cleveland heart disease	9.5+/-0.92	7.2+-0.85

Table 5: Average no of terms per rule

Data set	Ant-Miner	Proposed algorithm
Ljubljana breast cancer	1.28	1.23
Wisconsin breast cancer	1.97	1.21
Tic-tac-toe	1.18	1.26
Dermatology	3.16	2.83
Hepatitis	2.41	2.12
Cleveland heart disease	1.71	1.84

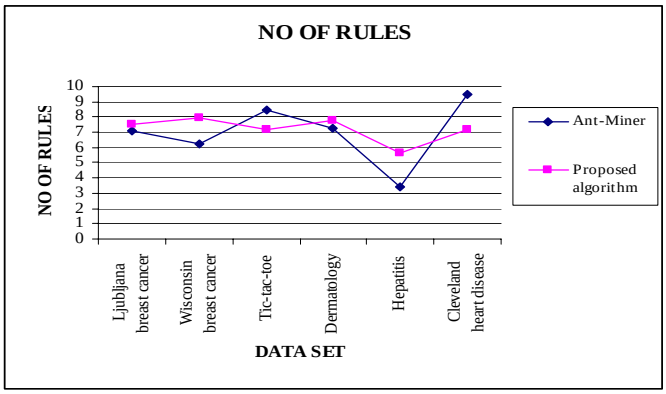

Figure 2: No of rules generated

Figure 3: No of terms per rule.

Conclusion and Future Work

This paper has presented an extension to Ant-Miner, which copes with continuous attributes during the rule construction process. The proposed algorithm had compared against Ant-Miner with respect to predictive accuracy and simplicity of the discovered rule lists in six public domain datasets. Regarding predictive accuracy, the proposed algorithm significantly outperformed than Ant-Miner in two datasets. Regarding simplicity of the discovered rule lists, the proposed algorithm found

significantly simpler rule lists than Ant-Miner in four out of six data sets. The algorithm had suitable for medical data sets to classify the data and predict the symptoms since we performed test on five medical data sets one non medical data set. We concluded that our algorithm was suitable for medical and other data sets. As future research direction, it would be interesting to incorporate Particle Swarm Intelligence (PSI) techniques to find the appropriate ranges for continuous attributes in datasets.

References

[1] Clark, P., Niblett, T.: The CN2 rule induction algorithm. Machine Learning 3(4)(1989) 261–283

[2] Chan, A., Freitas, A.: A new classification-rule pruning procedure for an ant colonyalgorithm.In: Artificial Evolution (Proceedings of the EA-2005). Lecture Notes in Artificial Intelligence 3871 (2006) 25–36

[3] Dorigo, M., St¨utzle, T.: Ant Colony Optimization. MIT Press (2004)

[4] Fayyad. U.M , Piatetsky-Shapiro.G, Smyth.P:From Data Mining to Knowledge Discovery: An Overview". In: U.M .Fayyad, G.Piatetsky-Shapiro, P.Smyth, editors,Advances in Knowledge Discovery & Data Mining, pages 1-34. MA, MIT Press, 1996.

[5] Martens, D., Backer, M.D., Haesen, R., Vanthienen, J., Snoeck, M., Baesens, B.:Classification with ant colony optimization. IEEE Transactions on Evolutionary Computation 11(5) (2007) 651–665

[6] Nalini.C,Balasubramanie.P: Discovering unordered rule sets for mixed variables using an ant-miner algorithm,Data Science journal,vol 7(2008) pp76-87.

[7] Parpinelli, R., Lopes, H., Freitas, A.: An ant colony algorithm for classification rule discovery. In Abbass, H., Sarker, R., Newton, C., eds.: Data Mining: a Heuristic Approach, Idea Group Publishing (2002) 191–208

[8] Parpinelli, R., Lopes, H., Freitas, A.: Data mining with an ant colony optimization algorithm. IEEE Transactions on Evolutionary Computation 6(4) (2002) 321–332

[9] Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)

[10] Smaldon.J,Freitas. A.A:, A New Version of the Ant-Miner Algorithm Discovering Unordered Rule Sets., GECCO'06, Seattle, Washington, USA,2006.

[11] Socha. K,ACO for Continuous and Mixed-Variable Optimization,Proceedings of the Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence,Brussels, Belgium,2004.

[12] Swaminathan, S.: Rule induction using ant colony optimization for mixed variable attributes. Master's thesis, Texas Tech University (2006)

[13] St¨utzle, T., Hoos, H.: MAX −MIN ant system. Future Generation Computer Systems 16(8) (2000) 889–914

[14] Witten, H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. 2nd edn Morgan Kaufmann (2005)
[15] www.ics.uci.edu/~mlearn/MLRepository.html