Ant Colony Optimization for Rule Induction with Simulated Annealing for Terms Selection

Rizauddin Saian
Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA
02600 Arau, Perlis, Malaysia.
rizauddin@perlis.uitm.edu.my

Ku Ruhana Ku-Mahamud
School of Computing
College of Arts and Sciences
Universiti Utara Malaysia, 06010 Sintok, Kedah
ruhana@uum.edu.my

***Abstract*—This paper proposes a sequential covering based algorithm that uses an ant colony optimization algorithm to directly extract classification rules from the data set. The proposed algorithm uses a Simulated Annealing algorithm to optimize terms selection, while growing a rule. The proposed algorithm minimizes the problem of a low quality discovered rule by an ant in a colony, where the rule discovered by an ant is not the best quality rule, by optimizing the terms selection in rule construction. Seventeen data sets which consist of discrete and continuous data from a UCI repository are used to evaluate the performance of the proposed algorithm. Promising results are obtained when compared to the Ant-Miner algorithm and PART algorithm in terms of average predictive accuracy of the discovered classification rules.**

Keywords—Rule induction; ant colony optimization; simulated annealing; classification; sequential covering algorithm; ant-miner; PART algorithm

I. INTRODUCTION

A sequential covering algorithm can extract IF-THEN classification rules directly from training data [1]. The rules are learned sequentially, one at a time. After each rule discovery, the covered examples by the rule are removed from the training set. The iterations stop when the number of examples in the training set is less than a predefined threshold value.

There are two types of Rule-Growing strategy to grow a rule: specific-to-general and general-to-specific. In the specific-to-general strategy a rule is selected at random from the list of examples in the training set. Terms from the selected rule are removed one by one while the rule's quality is improved.

On the other hand, in the general-to-specific strategy, each rule is initialized with a pre-defined class, and empty antecedent part. A function called the Learn-One-Rule function is used to extract a classification rule from the training set. Possible terms are greedily chosen to be included into the antecedent part of the rule. This process continues as long as the rule quality continues to be improved. This paper proposes a sequential covering based algorithm with a general-to-specific rule growing strategy to extract IF – THEN classification rules. The proposed Learn-One-Rule function to extract rules uses the Ant Colony Optimization (ACO) algorithm. Terms addition to the partial rule are optimized using the simulated annealing algorithm.

The rest of the paper is organized as follows. Section II reviews some literature on classification using ACO. Section III discusses the proposed algorithm. Section IV presents the results; and concluding remarks are presented in Section V.

II. RULE INDUCTION WITH ANT COLONY OPTIMIZATION

The ACO algorithm [2], specifically the ant-miner algorithm [3] employs artificial ants that cooperate to find good solutions for discrete optimization problems [4]. ACO has been applied to a variety of different problems, and recently the problem of data mining. The first application of ACO to the classification task is the ant-miner. The ant-miner algorithm is found to be competitive with the C4.5 algorithm [5] for classification, concerning predictive accuracy. An ant-miner is an ACO and data mining classification concept inspired algorithm proposed by Parpinelli et al. in 2002 [3]. The objective of the ant-miner is to extract classification rules from data, also known as rule induction. Parpinelli et al. [3] tested the ant-miner to medical databases [6]. The data sets involved comprise the Ljubljana breast cancer data set, Wisconsin breast cancer data set, Tic-tac-toe data set, Hepatitis data set, Dermatology data set, and Cleveland heart disease data set. Comparison of the results is made to the well-known C4.5 algorithm [5] using a 10-fold cross-validation procedure. Good classification performance is obtained in the experiments. It is also found that the ant-miner is able to achieve both good predictive accuracy and a reduced number of rules at the same time. However, the computational cost is high (time consuming), especially when the search space (the number of predicting attributes) is too large.

Seven different approaches have been taken by various researchers to improve the original ant-miner algorithm [3]. The first improvement is to change the heuristic measure. The original ant-miner algorithm [3] uses the entropy measure between terms to calculate the heuristic. Some research [7–11] proposes the use of a simpler heuristic function that is based on the frequency of the terms related to the predefined class. This simpler heuristic function is more computationally efficient, since the calculation is only based on the frequency, unlike the original heuristic function which depends on the calculation of entropy. The use of pheromones helps to optimize the constructed rules even though a simpler heuristic function does not consider the relationship between terms when new terms are added.

In the original ant-miner algorithm [3], the class is determined after each rule's construction. The second approach is to predefine the class before terms selection [7], [9], [10]. By fixing the class, only terms that reflect the

978-0-7695-4682-7/12 $26.00 © 2012 IEEE
DOI 10.1109/UKSim.2012.115

specific class are selected. Therefore, a simpler heuristic function could be used and, thus, reduce the processing time.

The third approach is the usage of a Pseudorandom proportional transition rule [4] applied by Liu et al. [8] and Wang and Feng [11]. This probability controls the effect of exploration and exploitation, based on a predefined constant parameter value, q_0. If the probability number generated is less than q_0, the term to be added is determined using the same function as the random proportional transition rule, as in the original ant-miner algorithm [3], which depends on the maximum value from the product of heuristic value and pheromone level from all terms. Otherwise, the term is probabilistically chosen by the random proportional rule, based on the product of heuristic value and pheromone level only.

The forth approach is to use different fitness functions to calculate the quality of the extracted rules. For example, Martens et al. [9] propose a new fitness function based on the confidence and coverage of the rule, instead of specificity and sensitivity of the rule used by the original ant-miner algorithm [3].

The pheromone levels evaporation for the original ant-miner algorithm [3] are performed implicitly by normalizing pheromone levels for all terms, after the levels are increased for the selected terms. However, this procedure sometimes faces problems in updating pheromone levels with low quality rules, since the updating procedure depends on the rule's quality. If the quality is very small, the new pheromone level will almost be the same as the previous level. Hence, the fifth approach is to use a different pheromone updating procedure. Liu et al. [8], Wang and Feng [11] and Smaldon and Freitas [10] propose different equations for updating pheromone levels, that could cope with a rule's quality that is close to zero. Additionally, Liu et al. [8], Wang and Feng [11] and Martens et al. [9] also propose that the updating procedure should use a predefined explicit evaporation rate.

The pruning procedure rule is known as a computational expensive procedure in rule induction, as well as the ant-miner. The sixth approach is the removal of the pruning procedure. In 2006, Martens et al. [9] proposed that pruning procedure removal is possible. Martens et al. [9] claimed that the removal of the pruning procedure rule improves the speed of the algorithm.

The seventh approach is to hybrid ACO with other optimization algorithms. For example, Saian and Ku-Mahamud [12] propose a hybrid algorithm of ACO algorithm and simulated annealing algorithm to optimize the extracted rule by an ant. A Simulated Annealing (SA) algorithm is a generic probabilistic metaheuristic for the global optimization problem of locating a good approximation to the global optimum of a given function in a large search space. An SA algorithm finds the best solution by choosing a random solution that depends both on the difference between the corresponding function values and also a global parameter temperature that is gradually decreased during the processes. This algorithm has the ability to reduce the problem of solutions stuck at local optimum. Hence, by using an SA algorithm for selecting terms for rules for each ant, a higher quality classification rule could be discovered.

III. The Proposed Algorithm

The problem scenario studied is as follows. Given a set of *m* attributes, $A = \{A_1, A_2, \ldots, A_m\}$ where each attribute A_i being a set of $V_i = \{v_1, v_2, \ldots, v_{f_i}\}$ of f_i values for attribute A_i, a set of *k* classes $C = \{c_1, c_2, \ldots, c_k\}$, and a training set $E = \{e_1, e_2, \ldots, e_n\}$, where each e_i is an example of data, in the training set of *n* examples. The objective is to learn a set of IF – THEN rules, with each rule in the form of (1).

$$\text{IF} < \textit{conditions} > \text{THEN} < \textit{class} > \qquad (1)$$

The $\text{IF} < \textit{conditions} >$ part is called the antecedent of the rule, and contains the logical combination of terms: $\textit{term}_1 \wedge \textit{term}_2 \wedge \ldots$, where each $\textit{term}_i$ is a triple with a combination of an attribute, an operator and a value in the domain of the attribute, $a_i = v_j$. For each attribute in a rule, only one value is possible. In other words, a rule cannot consist of two or more terms with the same attributes. The $\text{THEN} < \textit{class} >$ part is called the consequent, which predicts the class for the given new example.

The proposed algorithm is based on a sequential covering algorithm, with a general-to-specific rule growing strategy. Unlike the original ant-miner algorithm [3], class is predefined before a rule is extracted from the training data sets. The list of k classes is ordered in the class prevalence (the fraction of training examples that belong to a particular class). Rules are extracted, in order, from the examples in the training data set, for all $k-1$ classes.

The ACO algorithm, acting as the Learn-One-Rule function, is used to extract one best rule for the current list of examples from the training data set. Each rule is grown by adding terms one by one for the predefined class using the SA algorithm, while improving the rule's quality. The newly extracted rule is appended to the end of a list of discovered rules. Examples from the training data set covered by the newly extracted rule are removed. These processes continue until no further rule can be extracted for the current class.

Next, the proposed algorithm proceeds to extract rule(s) for the next class. These processes continue until rule(s) for all classes except for the last class have been extracted.

Finally, a default rule is added to the set of extracted rules. A default rule is a rule without the antecedent. The default rule is the last class in the ordered classes list. If there are only two classes, the proposed algorithm extracts rule for the class with the highest frequency of examples and the other class becomes the default rule. On the other hand, if more than one class exists, the proposed algorithm extracts the rules from all the classes, while the last class becomes the default rule. A summary of the sequential covering algorithm that used is given in Fig. 1.

Let E be the training examples and $Terms$ be the set of attributes-values pairs, $term_{ij}$.

Let C_0 be an ordered set of classes $\{c_1, c_2, \ldots, c_k\}$.

Initialize empty set of discovered rules $R = \{\}$.

for each class $c_i \in C_0 \setminus \{c_k\}$:

  while there is possible rule to extract:

    $r \leftarrow ACO(E, Terms, c_i)$.

    Remove examples from E covered by r.

    $R \rightarrow R \vee r$

$R \rightarrow R \vee r_{default}$

Figure 1. Sequential Covering Algorithm

In the ACO algorithm, as summarized in Fig. 2, each ant will create one rule by adding terms one by one, while the rule's quality is improved. The term selection uses the SA algorithm to optimize the selected term. The SA algorithm depends on a variable called the temperature. The temperature starts very high and gradually decreases, by a factor of a predefined threshold value, in a slow process. Terms addition will stop when the temperature reduces to a predefined lower limit temperature which indicates no further possible terms to add. Fig. 3 depicts the summary of the terms selection process.

Initialize ant colonies.

Initialize empty set of rules $R_{colonies} = \{\}$.

for all ant colonies:

  Extract one rule, r.

  Update pheromones.

  $R_{colonies} \rightarrow R_{colonies} \vee r$

Select best rule from $R_{colonies}$.

Figure 2. ACO Algorithm

A term is selected using the probability (P_{ij}) as follows:

$$P_{ij} = \frac{\tau_{ij} \times \eta_{ij}}{\sum_{i=1}^{a} \sum_{j=1}^{f_i} \tau_{ij} \times \eta_{ij}} \quad (2)$$

where

- η_{ij} is the heuristic value for term with attribute i and value j, and
- τ_{ij} is the pheromone level for term with attribute i and value j.

Let $Terms$ be the set of attributes-values pairs, $term_{ij}$.

Let c_i be the current class.

Initialize rule, $r : \{\} \rightarrow c_i$

while there is possible terms to add:

  $t \leftarrow SA(E, Terms, c_i)$

  $r \rightarrow r \vee t$

Figure 3. Terms Selection Process

The probability depends on the heuristic value and current pheromone level for each term. Over time, the pheromone levels for the higher quality terms will increase, and hence increase the chances of selecting those terms. The heuristic value for terms remains the same.

A heuristic function, along with the pheromone, is used to decide the selection of terms in a rule construction activity. The heuristic function is based on the frequency of the examples, defined by

$$\eta_{ij} = \frac{freq_{ij}^{w}}{\sum_{i=1}^{a} \sum_{j=1}^{b_i} freq_{ij}^{w}} \quad (3)$$

where $freq_{ij}^{w}$ is the number of examples for term with attribute i and value j for class w.

The rule is evaluated based on its quality. The new term is selected as the best term if the new partial rule's quality is better. A term with a lower quality rule also has a chance to be selected using the probability

$$p = \exp\left(\frac{-q_1 - q_2}{T}\right) \quad (4)$$

where

- q_1 and q_2 are the quality for the previous and current partial rules respectively, and
- T is the current temperature.

The temperature starts very high. Therefore, p is almost one, and all rules have almost the same probability to be chosen. As the temperature decreases slowly, the difference between the previous and current quality becomes more significant. Hence, the term with a slightly lower quality rule will be preferred over much lower ones.

The pheromone level for all terms are initialized equally using

$$\tau_{ij} = \frac{1}{\sum_{i=1}^{a} b_i} \tag{5}$$

where

- a is the total number of attributes, and
- b_i is the total number of values for attribute i.

The pheromone levels for terms used in the discovered rule, by each ant colony are be increased. The pheromone update at time t are carried out using

$$\tau_{ij}(t) = \left(1 + \tau_{ij}(t-1)\right) Q \tag{6}$$

where

- τ_{ij} is the pheromone level for term with attribute i and value j,
- $\tau_{ij}(t-1)$ is the pheromone level for previous term with attribute i and value j, and
- Q is the quality of the current constructed rule, defined by (7).

Rule pruning is performed after each ant discovers a rule. The pruning procedure iteratively removes one term at a time to improve the rule quality. The rule quality is calculated using fitness function (Q) given as

$$Q = \frac{1 + tp}{1 + k + tp + fp} \tag{7}$$

where

- tp is the number of examples covered by the rule and having the same class from the class predicted by the rule,
- fp is the number of examples covered by the rule and having a different class from the class predicted by the rule, and
- k is the total number of class.

IV. Experiment and Results

This study conducts experiments on seventeen data sets from a UCI repository [6]. The experiments are undeertaken using a ten-fold cross-validation technique [13]. Each data set is randomly shuffled and split into ten approximately equally-sized subsets. Each subset is used for testing while the rest are used for training. This process is repeated ten times, producing ten individual sets of performance statistics, the predictive accuracy. These performance statistics are averaged and the standard deviations for each of the performance statistics are calculated.

The proposed algorithm, like the original ant-miner algorithm [3], cannot cope with non-discrete attributes, only nominal attributes. Hence, non-discrete attributes are discretized. The discretization method used in this study is based on the method proposed by Fayyad and Irani [14]. Five hundred colonies each with one ant, are used in the experiments.

Fig. 4 shows the predictive accuracy of the proposed algorithm as compared to the original ant-miner algorithm [3] as well as the PART algorithm [15, 16], an industrial standard classification algorithm. The proposed algorithm discovers rules with a better accuracy rate than the original ant-miner algorithm [3] by 82.4%, with a huge improvement on three data sets: Vehicle: Tic-tac-toe and segment data sets, with an average percent difference of 15.4 %. Moreover, the average percent difference for the three data sets (Balance Scale, Ljubjana and Iris) where the original ant-miner algorithm [3] wins, is very small for an average of 1.2 %.

Compared to the PART algorithm, the proposed algorithm performs better on seven out of seventeen data sets. However, the average percent difference is very small at about 1.93 %. Incidentally, on the data sets where the proposed algorithm wins, the average percent difference is at 1.24 %.

As a conclusion, the proposed algorithm performs much better as compared to the original ant-miner algorithm [3], although, it is competitive with the PART algorithm.

V. Conclusions

This study proposed an algorithm based on a sequential covering algorithm. An ACO algorithm was used as the Learn-One-Rule function to extract rules from data sets, and an SA algorithm was used for terms selection while building the rule. The proposed algorithm used a simpler heuristic that does not take the relationship between terms into account, while selecting terms for growing a rule.

The performance of the proposed algorithm was compared to the original ant-miner algorithm [3] as well as the PART algorithm concerning the predictive accuracy, and found to be competitive with the PART algorithm, but much better than the original ant-miner algorithm [3]. Therefore, the SA algorithm is able to reduce the local optimum problem, where more related terms could be selected to grow a rule.

In future, it is suggested that the proposed hybrid algorithm should be enhanced in order to cope with non-discrete attributes on-the-fly. As an addition, it would be interesting to evaluate the performance of the proposed hybrid algorithm for selecting terms and fitness function for rule evaluation. Since, there is no perfect classification algorithm for all types of data sets, it is also practical to evaluate the performance of the proposed algorithm on other types of data sets, such as sparse and huge data sets.

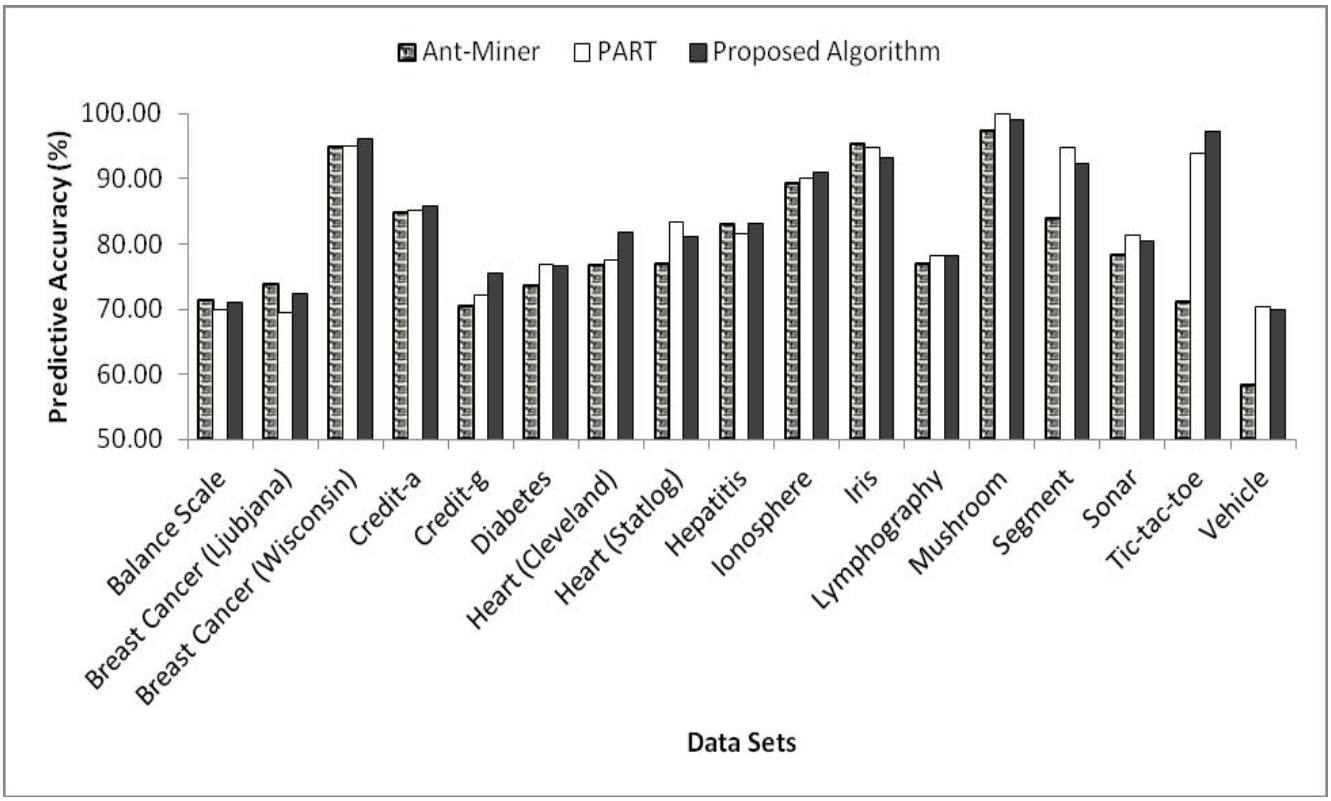

Figure 4. Average Predictive Accuracy

ACKNOWLEDGMENT

The authors wish to thank the Ministry of Higher Education Malaysia for funding this study under the Fundamental Research Grant Scheme, S/O code 11873 and RIMC, Universiti Utara Malaysia, Kedah, Malaysia for the administration of this study.

REFERENCES

[1] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*. Pearson Addison Wesley Boston, 2006.

[2] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 26, no. 1, pp. 29–41, 1996.

[3] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, "An Ant Colony Algorithm for Classification Rule Discovery," in *Data Mining: A Heuristic Approach*, H. A. Abbas, R. A. Sarker, and C. S. Newton, Eds. London: Idea Group Publishing, 2002, pp. 191-208.

[4] M. Dorigo and T. Stützle, *Ant colony optimization*. the MIT Press, 2004.

[5] J. R. Quinlan, *C4.5: programs for machine learning*. Morgan Kaufmann, 1993.

[6] A. Asuncion and D. J. Newman, *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences, 2007.

[7] M. Galea and Q. Shen, "Simultaneous ant colony optimization algorithms for learning linguistic fuzzy rules," *Swarm intelligence in data mining*, pp. 75–99, 2006.

[8] B. Liu, H. A. Abbass, and B. McKay, "Classification Rule Discovery with Ant Colony Optimization," *The IEEE Computational Intelligence Bulletin*, vol. 3, no. 1, pp. 31–35, Feb. 2004.

[9] D. Martens, M. De Backer, R. Haesen, B. Baesens, and T. Holvoet, "Ants constructing rule-based classifiers," *Swarm Intelligence in Data Mining*, pp. 21–43, 2006.

[10] J. Smaldon and A. A. Freitas, "A new version of the ant-miner algorithm discovering unordered rule sets," in *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, New York, NY, USA, 2006, pp. 43–50.

[11] Z. Wang and B. Feng, "Classification Rule Mining with an Improved Ant Colony Algorithm," in *AI 2004: Advances in Artificial Intelligence*, vol. 3339, G. Webb and X. Yu, Eds. Springer Berlin / Heidelberg, 2005, pp. 177-203.

[12] R. Saian and K. R. Ku-Mahamud, "Hybrid Ant Colony Optimization and Simulated Annealing for Rule Induction," in *2011 UKSim 5th European Symposium on Computer Modeling and Simulation (EMS2011)*, Madrid, Spain, 2011, pp. 70-75.

[13] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *International joint Conference on artificial intelligence*, 1995, vol. 14, pp. 1137–1145.

[14] U. Fayyad and K. Irani, “Multi-interval discretization of continuos attributes as preprocessing for classification learning,” in *Proceedings of the 13th International Join Conference on Artificial Intelligence, Morgan Kaufmann Publishers*, 1993, pp. 1022–1027.

[15] E. Frank and I. H. Witten, “Generating Accurate Rule Sets Without Global Optimization.,” in *Proc 15th International Conference on Machine Learning*, 1998, pp. 144–151.

[16] I. H. Witten, E. Frank, and H. Mark A., *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. Morgan Kaufmann Pub, 2011.