Ant Colony Optimization For Discovering Classification Rules

S. Sivakumari[1] R. Praveena Priyadarsini[2] P. Amudha[3]

Abstract

The focus of this paper is to investigate the classification performance of Ant Colony Optimization (ACO). We evaluate ACO algorithm in three public-domain, real-world datasets used to benchmark the performance of classification algorithms. We compare ACO algorithm to an industry standard PART and JRIP algorithms. The results show that accuracy of ACO algorithm is competitive with PART and JRIP on *Breast cancer* dataset and produces rule set of small size on *adult* and *mushroom* datasets.

Keywords : Ant Colony Optimization, PART, JRIP, Breast Cancer dataset.

1. Introduction

Data mining is the process of extracting models and patterns from large databases. Classification rule discovery is an important data mining task which associates an object/case to a class (among predefined set of classes) based on the object/case's attributes [1].

Applying Ant colony optimization (ACO) for discovering classification rules provides more flexible and robust search than traditional approaches [2,3]. ACO is one of the efficient meta_heuristics for combinatorial optimization problems [4].

Development of Ant colony algorithms for data mining is a promising research area. This paper discusses the ant colony based data miner (Ant-Miner) algorithm to extract classification rules from data. Parpinelli et.al, are the first to propose ACO algorithm for discovering classification rules with Ant-Miner [1]. The objective of ACO algorithm is to assign each instance to one class, from the set of predefined classes, based on the value of predictor attributes of the instance. The discovered knowledge is represented in the form, IF <conditions> THEN <class>. Entropy is the heuristic value employed in Ant-Miner. Modified version of Ant-Miner (Ant-Miner2) uses the heuristic value based on density estimation. Its improved version (Ant-Miner3) uses a different pheromone updating strategy and state transition rule.

We compare the performance of Ant-Miner3, PART (C4.5 rules) and JRIP (RIPPER) algorithms in terms of accuracy and number of rules generated. For experimentation, we have used three benchmark datasets *breast cancer* (Ljubljana), *mushroom* and *adult*.

The rest of the paper is organized as follows. Section 2 presents the main characteristics of ACO algorithm. Section 3 discusses the Ant-Miner algorithm, PART and JRIP algorithms for discovering classification rules. Section 4 presents the dataset description and preprocessing steps. Section 5 reports computational

[1]Professor & Head, [2] Senior Lecturer, [3]Lecturer, Department of CSE, Faculty of Engineering, Avinashilingam University for Women, Coimbatore.
Email : hod_cse_au@yahoo.co.in, priyadarshini_ rkumar@yahoo.co.in , amaranch_03@yahoo.co.in

results evaluating the performance of the algorithms across three unbalanced datasets. Finally, section 6 provides conclusions and directions for future research.

2. Ant Colony Optimization (ACO)

ACO is a system based on agents that simulate the natural behavior of ants, including mechanisms of co-operation and adaptation. ACO is one of the most recent techniques for approximate optimization which is inspired by the ant's foraging behavior [5]. While searching for food, ants deposit pheromone on their path. Paths with higher pheromone levels will more likely be chosen and thus reinforced and pheromone intensity of paths that are not chosen is decreased by evaporation. This positive feedback process, stigmergy, provides the ant colony shortest path finding capabilities [6]. Artificial ants have a probabilistic selection for paths with larger quantity of pheromone and use an indirect communication with other ants depending on the quantity of pheromone deposited on each path. Shortest path will have larger rate of growth in the amount of pheromone.

3. Methodology

A. Ant-Miner

In an ACO algorithm, each ant incrementally constructs a solution for the target problem [7]. For the problem of extraction of classification rules from data, Ant-Miner algorithm provides good solution. Classification rule has the form, IF <condition> THEN <class>. For generating classification rules, ants find several solution paths from the partial condition nodes to a class node. Each solution path corresponds to a classification rule that can include certain number of instances in the training set. Based on this concept, Ant-Miner algorithm is designed and is presented in Figure 1

```
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 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 rule;
    Remove the cases correctly by the selected rule from the training set;
END WHILE.
```

Figure 1: Overview of Ant-Miner Algorithm [8]

Ant-Miner algorithm is the result of applying ACO for discovering rules. This learning algorithm searches for a rule list in an incremental fashion. In each iteration, the algorithm discovers one rule at a time and adds it to the end of the list of discovered rules. The instances which are covered by the rule are removed from the training set. This process will be done iteratively and finally a set of rules would be discovered from the remaining training instances until the number of uncovered instances in the training set is less than a predefined threshold called max_uncovered_cases. The discovered rules are used to classify new instances which are unseen during training.

Ant-Miner consists of three major steps: rule construction, rule pruning and pheromone updating.

The initial amount of pheromone deposited at each path is inversely proportional to the number of values of all attributes. For state transition rule construction, Ant-Miner computes the value of heuristic function which may be either entropy measure or density-based. After the ant completes the construction of a rule, rule pruning is undertaken to increase the comprehensibility and accuracy of the rule. After rule pruning the amount of pheromone in all segments of all paths must be updated. For selecting the best rule between the discovered ones and for pheromone updating, Ant-Miner computes the quality of the rule according to equation (1)

$$Q = \frac{TP}{TP+FN} \quad \frac{TN}{FP+TN} \qquad (1)$$

where *TP (true positive)* is the number of instances covered by the rule that have the class predicted by the rule.

FP (false positive) is the number of instances covered by the rule that have a class different from the class predicted by the rule.

FN (false negative) is the number of instances that are not covered by the rule but that have the class predicted by the rule.

TN (true negative) is the number of instances that are not covered by the rule and that do not have the class predicted by the rule.

During iteration, Ant-Miner uses some parameters which are as follows:

1. *Number of ants (No_of_ants)*

2. *Minimum number of cases per rule (Min_cases_per_rule)*

3. *Maximum number of uncovered cases in the training set (Max_uncovered_cases)*

4. *Number of rules used to test convergence of the ants (No_rules_converg)*

The basic Ant-Miner uses a heuristic value based on entropy measure for the accuracy computation.

In the Ant-Miner2 version [9], computation is based on density estimation heuristic. Ant-Miner3 uses new pheromone updating method and new transition rule to increase the accuracy of classification by ACO.

B. PART and JRIP Algorithms

Ant-Miner differs from C4.5 decision trees [10] with respect to the heuristic function, is that in decision trees the information gain or entropy is computed for an attribute as a whole and in Ant-Miner it is computed for an attribute value pair only. Also, in decision tree algorithms, entropy measure is the only heuristic function used during tree construction whereas in Ant-Miner, entropy measure is used together with pheromone updating. Hence, Ant-Miner is more robust and less prone to get trapped into local optima in the search space.

PART is WEKA's improved implementation of C4.5 rules [11]. It is a rule induction algorithm that uses separate_and_conquer methodology with decision trees for building rules [12]. To build a single rule a pruned decision tree is build for the current set of instance, the leaf with the largest coverage is made into a rule and the tree is discarded. Using a pruned tree to obtain a rule instead of building it incrementally by adding conjunctions one at a time avoids the over-pruning problem.

JRIP is WEKA's implementation of RIPPER. It is a widely used rule induction algorithm [13] that extracts classification rules directly from data. This algorithm is particularly suitable for building models from unbalanced datasets. In two-class problems, RIPPER chooses the majority class as its default class and learns the rules for detecting the minority class. In RIPPER, support count of the rule is taken into account as an evaluation metric (Foil's information gain). The support count of a rule corresponds to the number of positive examples covered by the rule. This measure is employed to choose the best conjunct to be added into the rule antecedent from the last conjunct added to the rule. RIPPER performs additional optimization step to determine the best rules.

4. Data Set Description And Preprocessing

In order to improve the quality of data and mining results, the step of data preprocessing is essential. The present investigation analyses three unbalanced datasets available in UCI machine learning repository: *Adult*, *Mushroom* and Ljubljana *Breast Cancer* [14]. The dataset description is given in Table 1.

Table 1: Dataset Description

Dataset	No. of instances		Attributes
	Original	After resampling	
Adult	48842	2261	15
Mushroom	8124	2524	23
Breast Cancer	286	286	10

Adult dataset consists of 48,842 instances with 15 attributes (including class attribute) of various types which includes nominal, categorical and numerical. For the present study, we have selected 2,261 instances with the same class distribution ratio of the original dataset without missing values. The task is to predict if the income of a person is greater than 50K.

Mushroom dataset consists of 8,124 instances with 23 attributes (including class attribute) of nominal type. We have removed the instances with missing data and used only 2,254 instances for analysis. The task is to predict whether the particular *mushroom* is poisonous or edible.

Ljubljana *breast cancer* dataset consists of 201 instances of no-recurrence class and 85 instances of recurrence class with 10 attributes (including class attribute) which are linear and nominal. There is 1 missing data in *breast-quad* attribute which has been replaced by the data *left-up*. There are 8 missing data in the attribute *node-cap* and are replaced by the data *yes*. The task is to predict the possibility of recurrence and no-recurrence of *breast cancer* in patients.

The numerical attributes in the *adult* and *breast cancer* datasets are converted to nominal for experimentation. Such a conversion is not required for experimentation on *mushroom* dataset since all the available attributes are of nominal type.

5. Computational Results And Discussion

We evaluate comparative performance of Ant-Miner3, RIPPER and PART using 10 fold cross validation. In 10-fold cross validation, each dataset is divided into ten groups and each algorithm is run ten times using a different partition as test set each time, with the other nine as training set. Each subgroup is predicted via the classification rule constructed from the remaining nine subgroups. In the experiments ACO algorithm uses the following parameters:

No. of ants=100,

Max_uncovered_cases=10,

No. of rules covered=10,

Min_cases_per_rule=5.

Table 2 shows accuracy rates for the rule sets and the number of rules produced by Ant-Miner3, RIPPER and PART for ten runs on the three datasets. The predicted accuracies of the ten runs are averaged as the predictive accuracy of the discovered rule set.

Table 2 : Accuracy (%) And No. of Rules Generated

Dataset	Accuracy			No. of rules		
	Ant-Miner 3	RIPPER	PART	Ant-Miner3	RIPPER	PART
Adult	79.09% ± 0.59%	81.86 % ± 2.17 %	83.36 % ± 2.06	8±0	8.91 ± 1.58	43.82 ± 6.66
Mushroom	93.89 % ± 0.53%	100.00% ± 0.00%	99.98% ± 0.15%	6.7 ± 0.26	8.60 ± 0.59	13.27 ± 2.16
Breast cancer	75.1% ± 3.75%	70.50% ± 6.65%	69.98% ± 7.41	6.7 ± 0.3	3.03 ± 1.04	15.85 ± 3.98

Maximum accuracy is provided by all algorithms for *mushroom* dataset. ACO outperforms other two algorithms on Ljubljana *Breast cancer* dataset whereas JRIP [RIPPER] outperforms on *mushroom* dataset and PART [C4.5 rules] outperforms on a*dult* dataset. When the number of rules generated is considered, ACO generates fewer numbers of rules on *adult* and *mushroom* dataset, while JRIP gives minimum number of rules on *breast cancer* dataset.

6. Conclusion And Future Directions

This paper has presented a comparative study of ACO algorithm with JRIP [RIPPER] and PART [C4.5 rules]. Our study reveals that ACO algorithm is competitive with PART and JRIP in terms of accuracy on Ljubljana *Breast cancer* dataset and generates simpler rule sets on *adult* and *mushroom* datasets. JRIP is 100% accurate on *mushroom* dataset and comparatively good on other two datasets in producing rule sets. It produces small size of rule set on *breast cancer* dataset. PART produces rule sets that are as accurate as JRIP on all datasets. But the size of the rule set is larger which is undesirable. Possible extension of this work will be on applying particle swarm optimization (PSO) with ACO to improve the classification accuracy.

Acknowledgment

The authors would like to thank Dr. S.C. Sharma, Principal, R. V. College of Engineering, Bangalore, for his valuable guidance to carryout this work.

The authors would also like to thank M. Zwitter and M. Soklic, University Medical centre, Institute of oncology, Ljubljana, Yugoslavia for providing the *breast cancer* data set for running our experiments.

Reference

[1] S. Rafael Parpinelli, S. Heitor Lopes and A.Alex Freitas, " *An Ant Colony based system for data mining: applications to medical data* ", Proc. Int . Conf. Knowledge Discovery and Data, Boston MA, PP.55-62, 2000.

[2] S. Rafael Parpinelli, S. Heitor Lopes and A.Alex Freitas, "*An Ant Colony Algorithm for*

Classification Rule Discovery", In H.A.Abbass, R.A.Sarker, C.S.Newton (Eds), Data Mining: Heuristic Approach, PP. 192-208, London: Idea Group Publishing, 2002.

[3] B. Liu, H.A.Abbass and B. McKay, "*Classification Rule Discovery with Ant Colony Optimization"*, IEEE Computational Intelligence Bulletin, Vol.3 No.1, PP.31-35, 2004.

[4]. Xingliang zhu and JianZhang Li, " *An Ant Colony System-Based Optimization Scheme of Data Mining"*, Proceedings of the sixth International Conference on Intelligence systems Design and Applications, IEEE Computer Society, Vol.1, PP.400-403, 2006.

[5] Marco Dorigo, Mauro Birattari and Thomas stutzle, "*Ant Colony Optimization - Artificial Ants as a Computational Intelligence Technique"*, Technical Report No.TR/IRIDIA/2006-023, Belgium, 2006.

[6]. David Martens, et.al, "*Classification with Ant Colony Optimization"*, IEEE transactions on Evolutionary Computation, Vol.11, No.5, PP. 651-665, 2007.

[7]. Rafael.S.Parpinelli, Heiter S. Lopes and A.Alex Freitas, "*Data Mining with an Ant Colony optimization Algorithm"*, IEEE Transactions. on Evolutionary computation, Vol.6, No.4, PP.321-332, August 2002.

[8]. Wei-Jin Jiang, Yu-Hui Xu and Yu-sheng Xu, "*A Novel Data Mining Algorithm Based on Ant Colony System"*, Proceedings of the fourth International Conference on Machine Learning and Cybernetics, Guangzhou, PP.1919-1923, August, 2005.

[9]. B. Liu, H. A. Abbass and B. Mckay, *"Density-based Heuristic for rule discovery with Ant-Miner"*, The 6th Australia-Japan joint workshop on Intelligence and Evolutionary system, Canberra, Australia, PP.180-184, 2002.

[10]. J.R. Quinlan, "*C4.5: Programs for Machine Learning"*, San Macto, CA: Morgan Kaufmann, 1993.

[11]. I.H. Witten and E. Frank, "*Data Mining: Practical Machine Learning Tools and Techniques"*, Morgan Kaufmann, San Francisco, California, USA, 2nd edition, 2005.

[12]. P.N. Tan, M. Steinbach and V. Kumar, "*An Introduction to Data mining"*, Pearson Education Inc, 2006.

[13]. Frank.E, Witten .I.H, "*Generating Accurate Rule sets without Global Optimization"*, Machine Learning: Proceedings of the fifteenth International Conference, PP.144-151, 1998.

[14]. UCI Repository of Machine Learning Databases, University of California, Irvine, Department of Information and Computer Science. http://www.ics.uci.edu/mlearn/MLRepository.html

Author's Biography

S. Sivakumari, graduated from Madurai Kamaraj University, in Electronics and Communication Engineering during the year 1988. She obtained her Master degree in Applied Electronics from PSG College of Technology, Bharathiyar University, Coimbatore in the year 1995 and currently pursuing PhD degree in the area of data mining. At present she is a professor and Head of the Department of Computer Science and Engineering, Faculty of Engineering, Avinashilingam University for Women, Coimbatore, India. She has published many research papers in the International/National conferences. Her research areas include Data mining and Softcomputing. She has more than 19 years of experience in teaching and research. She is a member of ISTE and Computer Society of India.

R. Praveena Priyadarshini has graduated her B.E in Computer Science and Engineering in the year 1994 from Madras University, and received M.E degree in the year 2007 from Vinayaka University. She also belongs to the same department. She has seven years of teaching experience in the field of Computer science. Her research interests include Data mining and Software Engineering. She has published several papers in the National/International conferences. She is a member of ISTE and Computer Society of India.

P. Amudha has graduated her B.E in Computer Science and Engineering in the year 1993 from Bharathiar University, and received M.Tech in Information Technology in the year 2003 from Punjabi University. She also belongs to the same department. She has seven years of teaching experience in the field of Computer science. Her research interest include Data mining, Data structures, Compiler Design. She has published many papers in the National/ International conferences. She is a member of ISTE and Computer Society of India.