Journal of Theoretical and Applied Information Technology
15th November 2018. Vol.96. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ANT COLONY OPTIMIZATION ALGORITHM FOR RULE-BASED CLASSIFICATION: ISSUES AND POTENTIAL SOLUTIONS

HAYDER NASER KHRAIBET AL-BEHADILI[1], KU RUHANA KU-MAHAMUD[2], RAFID SAGBAN[3]

[1]Computer Science Department, Shatt Alarab University, Basra, Iraq

[2]School of Computing, Universiti Utara Malaysia, Kedah, Malaysia

[3]Department of Software, Faculty of Information Technology, University of Babylon, Babylon, Iraq

E-mail: [1]haider872004@gmail.com, [2]ruhana@uum.edu.my, [3]rsagban@uobabylon.edu.iq

ABSTRACT

Classification rule discovery using ant colony optimization (ACO) imitates the foraging behavior of real ant colonies. It is considered as one of the successful swarm intelligence metaheuristics for data classification. ACO has gained importance because of its stochastic feature and iterative adaptation procedure based on positive feedback, both of which allow for the exploration of a large area of the search space. Nevertheless, ACO also has several drawbacks that may reduce the classification accuracy and the computational time of the algorithm. This paper presents a review of related work of ACO rule classification which emphasizes the types of ACO algorithms and issues. Potential solutions that may be considered to improve the performance of ACO algorithms in the classification domain were also presented. Furthermore, this review can be used as a source of reference to other researchers in developing new ACO algorithms for rule classification.

Keywords: *Rule Discovery, Ant-Miner, Rule Pruning, Parameter Control, Metaheuristics*

1. INTRODUCTION

Data mining uses artificial intelligence, machine learning, database technology and, statistics power to mine massive amount of data to discover knowledge [1]. The data mining tasks can be classified into classification, clustering, regression, and association [2]-[4].

Classification has been one of the most studied topics in data mining. It is a supervised learning method for generating a classification model from given data and using the model to classify the class labels of unseen instances [5]. Classification problems exist in a variety of domains, including medical science, management science, engineering, and computer science. Common examples include medical diagnosis, customer segmentation, digit recognition, credit scoring, and bankruptcy classification [6]–[9]. Through the years, many techniques have been proposed to solve classification problems, which can be presented in different forms to represent knowledge, by employing machine learning, statistical, and artificial intelligence models and methods, such as linear models, decision trees, rule-based models, nonlinear models, and lazy evaluation methods. Other popular techniques and algorithms used for classification include support vector machines, Iterative Dichotomiser 3, Ant Colony Optimization (ACO), neural network, and K-nearest neighbor as shown in Figure 1 [2]. Some of these techniques generate classification models with high accuracy, but the structures of the models are highly complex and difficult to understand, as in the case of neural networks and support vector machines. Decision trees generate the model by selecting one attribute at a time by using a greedy heuristic method to decide the relevance of the attributes. As such, decision trees ignore attribute interactions and may consequently generate a suboptimal classification model. Meanwhile, the K-nearest neighbor algorithm presents high computational cost due to the large number of neighbors compared with unlabeled instances. Therefore, this algorithm is sensitive to noise or irrelevant data, which seriously affect classification performance.

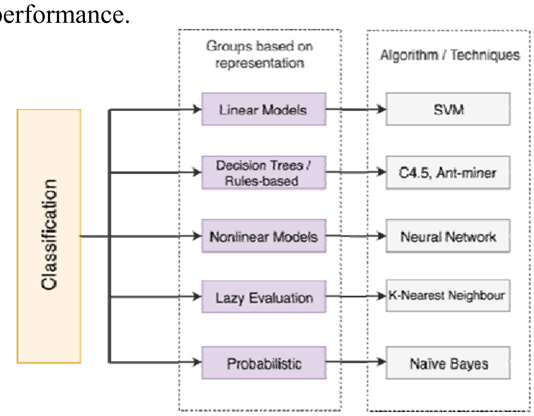

Figure 1. Taxonomy of Data Mining Classification Techniques

To overcome the aforementioned drawbacks, ACO-based classification methods have been proposed as alternatives. ACO is inspired by the natural phenomenon of stigmergy, which characterizes the behavior of ants, which applied to solve different combinatorial optimization problems [10], [11]. ACO uses a combination of two basic features. First, it uses a stochastic process, which helps explore a large area of the search spaces. Second, it uses an iterative adaptation procedure based on positive [12], [13]. These two features allow for the mitigation of the aforementioned drawbacks by generating an understandable and high-performance classification model. ACO algorithms have been extensively applied in the domain of classification rule discovery. ACO-based classification can also find globally optimal classification rules and handle attribute interactions better than existing rule induction algorithms, which typically behave as greedy in the rule construction process. However, ACO-based classification has some limitations which include local convergence, human-intensive parameter setting, computationally expensive rule pruning, low-quality dataset and attributes, lack of real-life applications, and open-source implementation. To best of our knowledge, there is no review paper present the main issues and the challenges that ACO-based classification needs to overcome.

The paper structure has been organized as follows. In Section 2, we describe the types of ACO algorithms for rule-based classification. Section 3 lists the main issues in designing and developing ACO algorithms for rule-based classification together with potential solutions to the issues. Finally, Section 4 concludes this review.

2. TYPES OF ACO ALGORITHMS FOR RULE-BASED CLASSIFICATION

In ACO-based rule discovery algorithms, useful information is expressed as rule antecedent (IF) and rule consequent (THEN). Therefore, knowledge is presented in the form of IF <conditions> THEN < class>. The rule antecedent (IF) consist of a set of conditions which is connected by a logical conjunction operator (AND), that is, IF term1 AND term2 AND. . . . Each term is a triple <attribute, operator, value> (i.e., < Blood pressure = high >).

The traditional ACO classifiers are divided into three categories, depending on how the ant searches for food and the pheromones are deposited. The early developed ACO classifiers are the family of Ant System (AS) algorithms, followed by various Ant Colony System (ACS) algorithms, which are extensions of AS algorithms with improved performance. These are then followed by the classifier according to a Max–Min Ant System (MMAS) algorithm [14]–[16]. All these algorithms are inspired by the foraging behavior real ant colonies. They use memory and stochastic behavior to generate a predicted rule list, which is understandable. Table 1 breaks down the literature of ACO (from 2002 to 2018) in rule discovery into three types (i.e., AS, ACS, and MMAS). We observe that the vast majority of ACO-based classifications have been developed based on the same overall design of AS. Experiments have shown that the best result obtained by AS and MMAS variants are competitive or better based on classification accuracy when compared to other techniques such as C4.5, RIPPER, CN2 and KNN [14], [16].

Table1. ACO variants for rule-based classification

NO	ACO Variants	Reference
1.	AS	[14] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39]
2.	ACS	[15] [40] [41] [42] [43] [44] [45]
3.	MMAS	[16] [46]

In 2002, Parpinelli, Lopes, and Freitas proposed the Ant-Miner, which is an AS algorithm. This algorithm has three main algorithmic components: rule construction, rule pruning, and pheromone updating. The Ant-Miner starts by initializing the pheromone, then each ant randomly

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

chooses a term from the dataset, with the term being <attribute = value>. From its start term, the ant iteratively moves from one term to an unvisited term and consequently constructs the classification rule. The rule construction ends after each ant has visited all terms according to the heuristic information and amount of pheromones. Rule pruning reduces the size of the constructed rules to improve understandability. Terms are pruned one at a time to improve quality. The procedure is iterated until no further improvement occurs, or at least one term is left in the rule. Subsequently, the pheromones will be updated according to two basic steps. First, the deposited amount of pheromones in all terms appears in the rule based on rule quality. Second, the pheromones that evaporate from every term do not appear in the rule. Afterward, the ants start to converge to the high-quality rule. Later, the process is completed following two stopping criteria. For the first criterion, the number of ants should be equal to or greater than the number of constructed rules. The second criterion tests the convergence, at which the ants start to converge by constructing a rule the same as the rule that has been constructed before. After that, the best rule from all the constructed rules will be added to the predicted rule list, and all instances (records) covered by the rule will be removed from the dataset.

A number of modifications were developed to improve the AS variant in Ant-Miner algorithm. Liu et al. [17] has proposed another heuristic function which consider a less accurate and simpler approach. The new heuristic function makes the algorithm computationally less expensive. Wu and Sun [18] proposed an improvement to Ant-Miner by using a population of ants to discover the rule in each iteration. In addition, they have considered the algorithm computationally time and introduced a new heuristic function to solve it. Chan and Freitas [19] introduces computationally less-expensive rule pruning procedure to use with the datasets that consists large attributes' number. Chan and Freitas [20] proposed a multi-label data classification, where the discovered rule contains more than one class label in the consequent (then) part. The algorithm generates a number of rules rather than one rule in each single iteration. In addition, the number of pheromone matrices is equal to the number of class attributes and pheromone will be updated for each class attribute which appear in the discovered rule. In another work introduced by Smaldon and Freitas [21] the rule generation process is performed after the class consequent is selected. The ant constructs rule for each class consequent and proposed unordered rule set. The idea is that each ant will know the consequent of the rule during generation process and does not change. In [22] Jin et al. proposed a modification on Ant-Miner rule mining algorithm based-on multi-population parallel strategy and cost-based discretization method. The parameters of this algorithm were offline adjusted step by step. Junbing et al. [23] had introduced multiple ant colonies instead of one colony to be used in the original Ant-Miner. Those colonies work in parallel and generate one rule for each single colony. The pheromone update procedure was modified by allowing each colony to deposit distinct type of pheromone. Other works proposed by [24] and [25] were on threshold value that is added in the rule construction step. The threshold value is responsible for accepting or rejecting the terms from inclusion in the current rule. Each term had heuristic value greater than pre-specify a threshold value will be accepted to be inclusion in the current rule, otherwise the term will be rejected. Otero et al. [26] proposed a modification to Ant-Miner, named cAnt-Miner which has the ability to cope with continuous attributes on the fly. This algorithm used the entropy-based discretization method. An extension to cAnt-Miner, named cAnt-Miner2 was introduced by the same research group [27]. This extension is proposed to give the algorithm flexibility to represent the continuous attributes and deposit the pheromone on edges of the construction graph rather than the vertices which are used in cAnt-Miner. Another extension of Ant-Miner was provided in [28] which uses the selection of class consequent prior to rule construction and give each class different pheromones type. The researchers then explore the effect of different quality measurement functions on the rules discovered in terms of rules size and predictive strength. Shahzad and Baig in [29] present a new method to select terms in the rule construction process by considering the relationship between previously selected term and the candidate term, as well as the rule consequent. Another heuristic information function proposed by Liang et al. [30] will consider both the instances coverage and the correlation between attributes. Baig and Shahzad [31] proposed a new heuristic function for Ant-Miner algorithm based on the correlation between the attributes. The work proposed by [32] hybridized the Ant-Miner with the concept of the fuzzy logic to generate a list of classification rule to diagnosis the hepatitis disease. The framework of this research contains two steps. The first step uses the Ant-Miner algorithm to generate a list of rules while the second step carries out the fuzzy logic concept to select the best rules. Liang et al.[33] proposed a multi-level rule choosing mechanism to select more accurate rule set. Other coupled between Ant-Miner and Simulated Annealing algorithm was proposed by Rizauddin and Ku-Mahamud [34] to optimize the terms selection in the rule construction process. The works proposed by Salama and Abdelba [35] introduces multiple pheromone types for each class consequent and also allows each ant to keep track on its own personal past history. In addition, it provides an online parameter adaptation by letting each ant to has its own values of α and β parameters and then discovered the rule. Prabha and Balraj in [36] used the Ant-miner algorithm in protein function prediction as a hierarchical multi-class classification problems. In [37] an improvement to Ant-Miner algorithm was proposed by letting each ant to dynamically select the quality function before constructing the rule.

ACS-based classification improves on AS-based classification by increasing the significance of exploration the search space toward the high-quality rule. This goal is achieved through an adaptive state transition strategy. In the adaptive state transition strategy, called pseudo-random proportional rule, with probability q0,such that $0 \leq q0 < 1$, the ants move to the next term with the maximum product between the heuristic information and pheromone amount [15], [40]. An extension to ACS-based classification was proposed by [41] which used Laplace correction as a new heuristic function. Another ACS variant reported in [42] was a modification on Ant-Miner for intrusion detection. The modification involves three aspects, the transition rule, the pheromone update procedure and the fitness function. Jiang et al. [43] have proposed ACS for rule-based classification with three major modifications on the transition rule, the heuristic function and the pheromone update procedure. In [44], an enhanced version of ACS focuses on the strategy punishing operator and mutation operator. The punishing operator is used in the pheromone update procedure to reduce the number of terms per rule while the mutation operator is used to enhance the quality of the best found rule. This is followed by a comparison between the mutated rule and the original best found rule before the best one among them is selected. Zhang and Sun [45] proposed a new pheromone update method based on the evaporation factor called ρ which deterministically updated during the search process. This mechanism aims to control the value of ρ factor. Firstly, the value of ρ will be set between $0 < \rho \leq 0.75$. Secondly, as the search process expanding, the value of ρ will be incrementally increased but this increasing amount will then slow down until finally, the maximum value of ρ is equal to 0.75. This mechanism avoids the ant to convergence on specific rule. This study has also proposed a new heuristic function which is considered a simpler approach but the result is less accurate.

The first algorithm based on the high-performance MMAS for rule-based classification is Antminer+ [16], [46]. Antminer+ is different from AS-based classification algorithms in many aspects. First, the pheromone is initialized with the upper trail limit, and upper and lower bounds are introduced to the amount of pheromones deposited. The pheromones are updated using a strong elitist strategy, which can be an iteration of the best ant or the global-best ant. Next, Antminer+ has a different heuristic function, a new pheromone updating strategy, and new self-adaptive mechanism to weight the α and β parameters. Finally, Antminer+ defines the environment as a directed acyclic graph, whereas all other algorithms use a fully connected graph. This mechanism allows for the selection of the nodes in one variable instead of the selection among all nodes as done in all previous algorithms. Figure 2 depicts the distribution of the literature on ACO variants in rule discovery from 2002 until 2018.

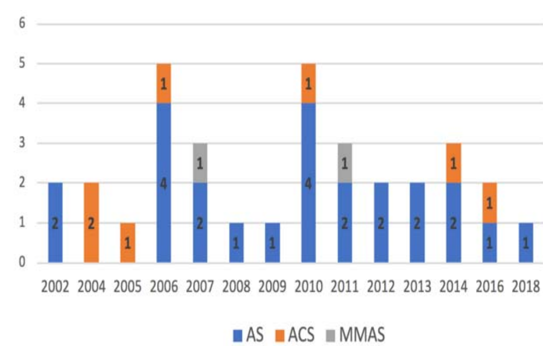

Figure 2. Distribution of ACO variants in classification rule discovery

Studies on rule discovery based on AS have been consistent throughout the years compared to studies based on MMAS and ACS. The three ACO types have been developed to enhance the exploration/exploitation balance and produce improved classification rules. However, the original ACS algorithm was developed based on the exploitation of information that was collected by previous ants. Such a strong elitist strategy was proposed to enhance exploitation over the exploration of the search space. By contrast, the subsequent ACS-based classification algorithms improved exploration by using the adaptive state transition strategy but did not improve exploitation. As a result, when the search space is too large, the imbalance between exploration/exploitation will affect the entire searching process and produce a nonoptimal solution. By contrast, Antminer+ uses a simpler heuristic function, which has the advantage of reduced computational cost. However, this heuristic function is less accurate than that of the Ant-Miner, on the rationale that the use of pheromones should compensate for the less accurate estimate of the simpler heuristic function. Therefore, the time complexity of the heuristic function of original Ant-Miner does not seem to pose a significant problem in the context of the entire algorithm as the heuristic function values for all terms can be computed in linear time with respect to the number of cases and attributes. The next section will discuss in detail the issues of ACO-based classification rule discovery. Potential solutions to the issues are also provided.

3. ISSUES AND POTENTIAL SOLUTIONS

The success of ACO-based classification highly depends on achieving the right balance between exploration and exploitation. This balance is necessary to avoid premature convergence to a suboptimal solution in any underlying search space for a given classification problem. The ACO algorithm needs to explore thoroughly the portions in the search space that seem promising so that it can exploit the best solutions in these areas. In addition, ACO faces an overfitting problem, when the classification rule perfectly fits that from which it was generated. Rule pruning is used to avoid overfitting, but among the components of ACO algorithms, it has the highest computational cost. Furthermore, despite the importance of setting the ACO parameters to increase the classification accuracy, the ACO parameters are kept constant during the learning process, and no comprehensive study has been conducted to highlight the ideal techniques for ACO parameters. In addition, ACO-based classification techniques require a preprocessing step to improve the quality of the dataset and discretize all variables as ACO handles discrete variables. Furthermore, the training time of existing ACO-based classification algorithms is considerably longer when compared with other rule-based classification algorithms, such as RIPPER and C4.5.

In the next subsections, useful insights into how the main challenges and issues of ACO rule-based classification can be overcome are highlighted.

3.1 Local Convergence

Data classification is considered an NP-hard problem due to the large learning space of different data sources, such as medical agencies, online forum platforms, and mobile apps [25], [47], [48]. For example, in the diagnosis application domain, which is considered one of the principal application areas of expert systems, data classification is used to detect diseases at an early stage [49]–[51]. The algorithms for handling such problems are divided into two types: exact and approximation. The first type is ensured to find the optimal solution. However, in a large and complex search space, exact algorithms cannot find the optimal solution and may even produce worse solutions. By contrast, approximation algorithms balance between runtime and solution quality to find an efficient solution. Approximation algorithms can be classified into heuristic and metaheuristic. The shortcomings of heuristic algorithms include stopping at poor-quality local optima and being highly dependent on initial solutions. These disadvantages of heuristic algorithms motivated the development of new general approaches called metaheuristic algorithms, which aim to avoid the complexity problems. Metaheuristic algorithms are high-level strategies or general heuristic designs for exploring the search space and finding a high-quality solution to an optimization problem.

In terms of data classification, ACO as a metaheuristic algorithm has been proven to be an effective technique for classification tasks and knowledge discovery. Experiments have shown that ACO achieves performance similar to or even better than those of other classification techniques in certain application domains [31], [42], [52]–[58]. The performance results for the classification problems in these domains show that ACO is one of the best and most successful metaheuristics in data classification.

ACO is a constructive algorithm that builds its solutions stepwise from scratch. An ACO algorithm searches for the solution with predefined termination criteria. If one of these criteria is reached, then the best solution found is returned. In some cases, achieving the termination criteria early leads to poor solutions. However, continuing the search beyond the termination criteria will not produce significant improvements in the solutions.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

That is, the algorithm prematurely converges, and no further exploration is possible. To prevent premature convergence, the ACO algorithm should work with each single solution and iteratively employ the neighborhood search through the addition of a local search procedure. Researchers need to focus on hybrid approaches to improve the quality of solutions because ACO cannot completely alleviate the problem of local convergence. Studies on hybridization with local search but without any improvement in classification result were also reported [16], [34]. A deep investigation must be carried on where local search can be combined to improve the classification algorithm. Two types of hybridization can be adopted. The first type is the best improvement, which is very time consuming, especially with large-scale problems. The second type is the first improvement of each neighborhood search.

3.2 The Computation of Rule Pruning

Rule pruning is an important ACO-based classification component [38]. This procedure is aimed to improve both the quality of the generated rule and its comprehensibility. Similar to other rule discovery algorithms, ACO can potentially discover the classification rules with a long rule antecedent (consisting of many terms), which reduces the comprehensibility of the rule. In this case, ACO generates rules whose number of terms is approximate to the same number of attributes in the dataset. Consequently, the rule will lose its understandability and will be extremely long and difficult to interpret. Thus, short and comprehensible rules should be generated.

The majority of ACO-based classification algorithms [14], [15], [17], [18], [22], [26]–[28], [32]–[35], [37], [40]–[44], [47], [59]–[63] are derived from the technique suggested in [64]. This method deletes one condition from the rule antecedent part while enhancing the predictive accuracy of the rule. This procedure recursively iterates until one condition is left in the rule or no condition removal can further improve the quality of the rule undergoing pruning. Afterward, it checks the rule consequent (THEN) whether the instances covered by the original rules changed compared with the pruned rule. However, the computational cost of pruning is the highest among those of the other components of these algorithms. Thus, the pruning technique is very sensitive to the number of attributes in the dataset. Generally, a large number of attributes lead to a large number of conditions in the rule prior to pruning and consequently lead to a large number of removal iterations during rule pruning. Other algorithms proposed simplify the pruning procedure by pruning the best rule discovered by all ants instead of pruning each candidate rule generated by each ant [16], [29], [31], [65]. Simplifying the pruning procedure will reduce the computational cost; however, it prevents high-quality rules from being found because the pruning procedure will not explore all rules generated by each ant, or at minimum the elite rules set.

Chan and Freitas (2006) proposed a hybrid rule-pruning procedure, which coupled the original rule pruning with a different pruning step based on information gain. The hybridization according to pre-defined parameter, which represents the number of terms need to be included in each rule. The proposed procedure operates on two principles. First, if the rule consists of a number of terms less than the predefined number of terms, then the original rule pruning proceeds. Second, if the constructed rule consists of terms whose number is more than the predefined number of terms, then the procedure will reduce the number of terms on the basis of the pre-calculated term's information gain through the roulette wheel selection technique and, later, the new rule proceeds directly to the traditional rule pruning procedure [19].

Other studies [24], [25] have introduced a threshold criterion for accepting or rejecting a term to be added in the rule according to its strength. However, the predefined number of terms and the threshold are considered very critical and highly dependent on the dataset, and the user determiner could worsen the quality of the discovered rule.

For further research is to check the performance of the pruning technique that prunes only an elitist list of rules instead of pruning each rule generated by each single ant. Another research direction is to check pre- scheduling rule to change the value of threshold or parameter during the learning stage. Automatic and adaptive on-the-fly parameter control can also be used to change the threshold or predefined number of terms, rather than having them statically determined by the user.

3.3 Parameter Control

Parameter setting is considered an optimization problem as the optimal parameter values are searched for in large search spaces [66]–[68]. Parameter setting can be used to balance exploration and exploitation in the search process. The parameters of ACO rule-based classification are the following: the first parameter is ant number (No_of_ants), which corresponds to the number of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

generated rules. A higher value of this parameter will generate more rules but result in slower learning time of the algorithm. The second parameter is the minimum number of instances covered by each single rule (Min_cases_per_rule), which aims to avoid constructing a rule that perfectly fits that from which it was generated and has a certain degree of generality. The third parameter is the maximum number of uncovered instances (Max_uncovered_cases), which aims to generate a list of rules that covered the maximum number of records in the dataset. The other parameters include the number of rule convergences (No_rules_converg) to test the rule convergence and parameters α and β. The latter two respectively aim to determine the influences of the pheromones on the heuristic information.

Furthermore, the performance of ACO algorithms for rule discovery is highly dependent on the settings of their parameters. An experiment showed that different values of the input parameters in Ant-Miner can vary the classification result by up to 20% [69]. Indeed, setting the ACO parameters is considered a nontrivial task. Thus, no uniform parameter settings exist across datasets and different application domains. In addition, the main challenges of controlling the parameters of ACO for classification rule discovery are attributed to the interactions between these parameters, the mutual effect of these parameters on each other, and their influences on the ACO behavior.

The parameter setting tasks can be either offline tuning or online control. Traditionally, the majority of ACO-based classification algorithms use offline tuning [14], [15], [17]–[32], [34]–[46]. Offline tuning uses the trial-and-error method to set the values of the parameters. This process is error prone, human intensive, and time consuming. Furthermore, offline tuning by trial and error depends, to a great extent, on the experience of the algorithm developer. This dependence worsens with an increase in the types of classification domains and the number of possible parameter values. The alternative method to set the parameters is online control, which consists of the modification of the parameter values during the run of the algorithm as a function of certain statistics on the algorithm behavior.

Many parameter control strategies have been studied in the area of ACO for optimization [70], [71]. Parameter control strategies vary the parameters values at computation time on the basis of the search progress or prescheduled modes. The types of parameter control strategies consist of deterministic parameter control, adaptive parameter control, and self-adaptive parameter control.

The deterministic parameter control varies the parameter values by using a deterministic rule. This strategy does not use any feedback from the search. A predetermined schedule is used to vary the parameter values when a set number of solutions have been obtained since the last time the rule was activated. Salama and Abdelbar (2010) proposed a deterministic parameter control mechanism called "ants with personality" or µAnt-Miner algorithm. It is used to control the values of α and β, which are drawn from a random strategy using a Gaussian distribution with a mean of 2 and a standard deviation (σ) in the range [0, 1]. High values of the standard deviation introduce a high range of diversity in selecting a term [35].

The adaptive parameter control changes the parameter values by using the feedback from the search. Different measures can be used for this strategy according to absolute or relative evidence to determine the parameter variation strategy. The selection of the values of the parameters may involve reward assignment, and the action of the ACO may decide whether the new value prevails in the search space or not.

The self-adaptive parameter control adds parameter values in the ACO search space. Better parameter values will lead to better rules. Ants during the search converge on the best parameter values. Martens et al. (2007) proposed a self-adaptive strategy for the Antminer+ algorithm by adding two additional vertex groups in the construction graph of the search space and letting the ants select appropriate values for the parameters α and β before constructing the rule. The boundaries of both parameters are limited within the range [1, 3] [16].

However, we find a clear weakness on parameter setting studies for ACO rule-based classification. Further research is required in-depth to understand the influence of individual parameters. Then, decisions about which parameters to adapt and how to adapt them are mostly arbitrary. The parameter control is a relatively large field, and many strategies have been proposed in the context of ACO for optimization, but their adoption in the ACO for classification is considered an open research opportunity with potential significant impact.

3.4 Poor Dataset Quality

Intelligent ACO algorithms for classification can be seriously affected by the quality of the dataset. Thus, applying ACO to low-quality datasets hinders the discovery of a high-quality set of rules. Therefore, the preprocessing of the data is imperative to improve its quality. The preprocessing tasks are feature selection, instance reduction, discretization, and data cleaning.

Feature selection, also known as attribute selection and attribute reduction, is the process of selecting a subset of available attributes from the dataset to learn a classification model. Redundant or irrelevant features increase the rule construction time. The goal is to remove those attributes without considerable loss of information. Thus, this process can simplify the classification rules, reduce the learning time, and reduce the overfitting problem by improving the generalization of the rules. Further discussion on the comparison of attribute selection methods that were used with Ant-mining can be found in [39].

Instance reduction, which is also known as dataset reduction, is the process of selecting a manageable volume of instances from the dataset. It aims to reduce the computational resources that are necessary for performing the learning process. Instance reduction can also be used for removing noisy instances from the dataset. In addition, techniques used for instance reduction have to balance the classification accuracy and the reduction rate as well as preserve the distribution of classes in the dataset.

ACO does not have the ability to cope with continuous attributes and requires a preprocessing step to discretize the continuous attributes before constructing its classification model. Thus, an open research direction is the improvement of the ACO algorithm to enable it to preprocess data during learning rather than requiring a preprocessing step. Otero et al. (2008) introduced a promising algorithm that can handle such attributes in the rule generation process [26], [27]. In addition, further research could be conducted to investigate the encapsulation of different discretization techniques in the rule generation process. Finally, the Ant-mining algorithms have been proved to be competitive with popular classification but in all experiments the datasets that have been used did not contain a large number of attributes. In contrast, the classification by using Ant-mining algorithms tends to be very time consuming and produce rules have a large number of terms where the dataset being classified consists of a large number of attributes. Thus, a preprocessing framework should be developed for removing features or inaccurate instances and render the data ready for ACO to generate high-quality classification rules.

3.5 Need for Real-Life Applications

UCI benchmark datasets have been used for testing the performance of ACO-based algorithms for classification rule discovery for different numbers of instances, attributes, class labels, and attribute types (i.e., categorical and continuous). However, the danger of overfitting is imminent by using UCI benchmark datasets, while marginal improvements on popular, analyzed benchmarks have been implemented to provide a clear indication of algorithmic superiority [72]. Therefore, real-world applications are required in well-known application domains, such as text mining, credit scoring, medical diagnosis, and DNA sequence classification. Evaluating ACO-based classification algorithms with other swarm intelligence algorithms in real-life applications could provide extensive knowledge in their behavior and performance.

3.6 Available Software and Open-Source Implementation

The majority of the ACO-based classification rule discovery software packages are not publicly available. Only a few open-source implementation packages and libraries are available for using ACO in the classification task. Consequently, the evaluation of different benchmark datasets becomes difficult. The list below contains the only available open-source software packages:

Ant-Miner: https://sourceforge.net/projects/guiantminer
Ant-Miner, cAnt-Miner, and cAnt-Miner2 algorithms: https://sourceforge.net/projects/myra
Antminer+: http://www.antminerplus.com

Software packages that house different variants of ACO algorithms for rule discovery under one common framework are required. These software packages must be reasonably high-performing. Another requirement is to provide software packages for ACO-based classification similar to that for the ACO for Traveling Salesman Problem (TSP), which consists of different variants of the ACO algorithm: AS, MMAS, ACS, Elitist Ant System, Best–Worst Ant System, and Rank-Based Ant System.

4. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

ACO-based classification rule discovery algorithms are widely used in classification tasks to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

establish a classification model that is easy to understand. In addition, those algorithms show competitive performance results as compared to the traditional classification techniques and outperforms in some application domains. Nevertheless, these algorithms also have several issues and opportunities. Those issues include the local optimization problem, parameter setting, computationally expensive rule pruning, specific data requirements (i.e, less-quality dataset and attributes) and, finally, the requirement of open-source implementation and real-life research application. Consequently, those issues may reduce the classification accuracy and increase the computational time of the algorithm. This study has presented various enhancement possibilities and provided promising research directions for future studies by considering the existing issues and challenges. First, the right balance between exploration and exploitation should be determined. Second, the ACO parameters should be set during the learning process. Third, ACO requires a preprocessing step to improve the quality of the dataset and discretize all continuous attributes. Furthermore, although the pruning procedure compulsory which used to avoid overfitting, its computational cost is the highest in the ACO-based classification algorithms. Consequently, the learning time of existing ACO-based classification algorithms is considerably longer compared with other rule-based classification algorithms, such as C4.5 and RIPPER. Real-life applications, as well as open-source implementation, similar to RapidaMiner or Weka, are necessary to facilitate experimental design. Finally, ACO algorithm variants similar to ACO for TSP have not been fully applied in a classification rule discovery context. Using them could produce robust classification results.

ACKNOWLEDGEMENTS

The authors thank the Ministry of Higher Education Malaysia for funding this study under Fundamental Research Grant Scheme (S/O code 13240).

REFERENCES:

[1] J. Fürnkranz, D. Gamberger, and N. Lavrač, "Foundations of Rule Learning," *Cogn. Technol.*, vol. 6, no. 2003, pp. 19–55, 2012.

[2] A. Gavrilovski, H. Jimenez, D. N. Mavris, A. H. Rao, S. Shin, I. Hwang, and K. Marais, "Challenges and Opportunities in Flight Data Mining: A Review of the State of the Art," in *AIAA Infotech @ Aerospace*, 2016, no. January, pp. 1–18.

[3] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "A survey of multiobjective evolutionary algorithms for data mining: Part I," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 4–19, 2014.

[4] A. M. Jabbar and K. Ku-Mahamud, "Ant-based sorting and ACO-based clustering approaches: A review," in *IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE2018)*, 2018, no. July.

[5] T. N. Phyu, "Survey of Classification Techniques in Data Mining," *Int. MultiConference Eng. Comput. Sci.*, vol. I, pp. 18–20, 2009.

[6] R. P. Datta and S. Saha, "Applying rule-based classification techniques to medical databases : an empirical study," *Int. J. Bus. Intell. Syst. Eng.*, vol. 1, no. 1, pp. 32–48, 2016.

[7] H. N. K. Al-behadili, "Classification Algorithms for Determining Handwritten Digit," *Iraq J. Electr. Electron. Eng.*, vol. 12, no. 1, pp. 96–102, 2016.

[8] M. J. Kim and I. Han, "The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms," *Expert Syst. Appl.*, vol. 25, no. 4, pp. 637–646, 2003.

[9] H. N. K. Al-behadili, "Intelligent Hypothermia Care System using Ant Colony Optimization for Rules Prediction," *J. babylon Univ.*, 2016.

[10] H. J. A. Nasir and K. R. Ku-Mahamud, "Grid Load Balancing Using Ant Colony Optimization," in *2010 Second International Conference on Computer and Network Technology*, 2010, pp. 207–211.

[11] H. J. A. Nasir, K. R. Ku-Mahamud, and A. M. Din, "Load balancing using enhanced ant algorithm in grid computing," in *Proceedings - 2nd International Conference on Computational Intelligence, Modelling and Simulation, CIMSim 2010*, 2010, no. September, pp. 160–165.

[12] R. Sagban, K. R. Ku-Mahamud, and M. S. Abu Bakar, "ACO ustic: A nature-inspired exploration indicator for ant colony optimization," *Cient. World J.*, 2015.

[13] M. Dorigo and T. Stutzle, *Ant Colony Optimization: Overview and Recent Advances*, vol. 146. 2010.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[14] R. Parpinelli, H. Lopes, and A. A. AFreitas, "Data Mining With an Ant Colony Optimization Algorithm," *IEEE Trans. Evol. Comput.*, vol. 47, no. 6 (4), pp. 321–332, 2002.

[15] B. Liu, H. A. Abbass, and B. Mckay, "Classification rule discovery with ant colony optimization algorithm," in *IEEE Computational Intelligence Bulletin*, 2004, vol. 7435 LNCS, no. 1, pp. 678–687.

[16] D. Martens, M. De Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens, "Classification With Ant Colony Optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 5, pp. 651–665, 2007.

[17] Liu, H. A. Abbass, and B. Mckay, "Density-Based Heuristic for Rule Discovery With Ant-Miner," *6th Australas. Jt. Work. Intell. Evol. Syst.*, pp. 1–6, 2002.

[18] H. Wu and K. Sun, "A simple heuristic for classification with ant-miner using a population," in *the 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2012*, 2012, vol. 1, pp. 239–244.

[19] A. Chan and A. Freitas, "A new classification-rule pruning procedure for an Ant Colony Algorithm," in *International Conference on Artificial Evolution (Evolution Artificielle).*, 2006, vol. 3871 LNCS, pp. 25–36.

[20] A. Chan and A. a. Freitas, "A new ant colony algorithm for multi-label classification with applications in bioinfomatics," in *the 8th annual conference on Genetic and evolutionary computation*, 2006, pp. 27–34.

[21] J. Smaldon and A. a Frietas, "A New Version of the Ant-Miner Algorithm Discovering Unordered Rule Sets," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation. ACM,* 2006, no. 6, pp. 43–50.

[22] P. Jin, Y. Zhu, K. Hu, and S. Li, "Classification rule mining based on ant colony optimization algorithm," *Intell. Control Autom.*, no. 1, pp. 654–663, 2006.

[23] J. He, D. Long, and C. Chen, "An improved ant-based classifier for intrusion detection," *Third Int. Conf. Nat. Comput. ICNC 2007*, vol. 4, no. 60273062, pp. 819–823, 2007.

[24] K. Thangavel and P. Jaganathan, "Rule Mining Algorithm with a New Ant Colony Optimization Algorithm," *Int. Conf. Comput. Intell. Multimed. Appl. ICCIMA 2007*, vol. 1, pp. 561–563, 2007.

[25] S. Tripathy, S. Hota, and P. Satapathy, "MTACO-Miner : Modified Threshold Ant Colony Optimization Miner for Classification Rule Mining," in *Emerging Reserch in Computing, Information, Communication and Application*, 2013, no. August 2013, pp. 1–6.

[26] F. Otero, A. A. Freitas, and C. G. Johnson, "cAnt-miner: An ant colony classification algorithm to cope with continuous attributes," in *International Conference on Ant Colony Optimization and Swarm Intelligence.*, 2008, pp. 48–59.

[27] F. Otero, A. Freitas, and C. Johnson, "Handling continuous attributes in ant colony classification algorithms," in *2009 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009 - Proceedings*, 2009, pp. 225–231.

[28] K. M. Salama and F. E. B. Otero, "Using a unified measure function for heuristics, discretization, and rule quality evaluation in Ant-Miner," *2013 IEEE Congr. Evol. Comput. CEC 2013*, pp. 900–907, 2013.

[29] W. Shahzad and A. R. Baig, "Compatibility As a Heuristic for Construction of Rules By Artificial Ants," *J. Circuits, Syst. Comput.*, vol. 19, no. 01, pp. 297–306, 2010.

[30] Z. Liang, J. Sun, Q. Lin, Z. Du, J. Chen, and Z. Ming, "A novel multiple rule sets data classification algorithm based on ant colony algorithm," *Appl. Soft Comput. J.*, vol. 38, pp. 1000–1011, 2016.

[31] A. R. Baig and W. Shahzad, "A correlation-based ant miner for classification rule discovery," *Neural Comput. Appl.*, vol. 21, no. 2, pp. 219–235, 2012.

[32] S. Madhusudhanan, M. Karnan, and K. Rajivgandhi, "Fuzzy based ant miner algorithm in datamining for hepatitis," *2010 Int. Conf. Signal Acquis. Process. ICSAP 2010*, pp. 229–232, 2010.

[33] S. C. Liang, Y. C. Lee, and P. C. Lee, "The application of ant colony optimization to the classification rule problem," *IEEE Int. Conf. Granul. Comput. GrC 2011*, pp. 390–392, 2011.

[34] R. Saian and K. R. Ku-Mahamud, "Hybrid Ant Colony Optimization and Simulated Annealing for Rule Induction," *2011 UKSim 5th Eur. Symp. Comput. Model. Simul.*, no. March 2016, pp. 70–75, 2011.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[35] K. M. Salama and A. M. Abdelbar, "Extensions to the Ant-Miner classification rule discovery algorithm," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6234 LNCS, pp. 167–178, 2010.

[36] G. M. Prabha and E. Balraj, "A HM Ant Miner Using Evolutionary Algorithm," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 3, no. 3, pp. 1687–1692, 2014.

[37] S. Rajpiplawala and D. K. Singh, "Enhanced cAntMiner PB Algorithm for Induction of Classification Rules using Ant Colony Approach," *IOSR J. Comput. Eng.*, vol. 16, no. 3, pp. 63–72, 2014.

[38] H. N. K. Al-behadili, K. R. Ku-mahamud, and R. Sagban, "Rule Pruning Techniques in the Ant-Miner Classification Algorithm and Its Variants : A Review," in *IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE2018)*, 2018, pp. 1–7.

[39] R. Saian and K. R. Ku-Mahamud, "Comparison of attribute selection methods for web texts categorization," *2nd Int. Conf. Comput. Netw. Technol. ICCNT 2010*, pp. 115–118, 2010.

[40] Z. Wang and B. Feng, "Classification Rule Mining with an Improved Ant Colony Algorithm," in *Australian conference on artificial intelligence*, 2004, pp. 357–367.

[41] B. Fakhar, "A Novel Method for Extracting Classification Rules based on Ant-Miner," *J. Math. Comput. Sci.*, vol. 8, pp. 377–386, 2014.

[42] D. Agravat, U. Vaishnav, and P. B. Swadas, "Modified Ant Miner for intrusion detection," in *ICMLC 2010 - The 2nd International Conference on Machine Learning and Computing*, 2010, pp. 228–232.

[43] Jiang, Y. Xu, and Y. Xu, "A novel data mining algorithm based on ant colony system," in *Fourth International Conference on Machine Learning and Cybernetics,* 2005, no. August, pp. 18–21.

[44] J. Ji, N. Zhang, C. Liu, and N. Zhong, "An Ant Colony Optimization Algorithm for Learning Classification Rules," *WI '06 Proc. 2006 IEEE/WIC/ACM Int. Conf. Web Intell.*, pp. 1034–1037, 2006.

[45] X. Zhang and W. Sun, "An Adaptive Ant Colony Algorithm for Classification Rule Mining," in *2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE2016)*, 2016, vol. 133, pp. 295–299.

[46] W. Verbeke, D. Martens, C. Mues, and B. Baesens, "Building comprehensible customer churn prediction models with advanced rule induction techniques," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2354–2364, 2011.

[47] S. Hota, P. Satapathy, and A. K. Jagadev, "Modified Ant Colony Optimization Algorithm (MAnt-Miner) for Classification Rule Mining," in *Intelligent Computing, Communication and Devices.*, 2015, vol. 1, pp. 267–275.

[48] H. Jiang, A. Yang, F. Yan, and H. Miao, "Research on Pattern Analysis and Data Classification Methodology for Data Mining and Knowledge Discovery," *Int. J. Hybrid Inf. Technol.*, vol. 9, no. 3, pp. 179–188, 2016.

[49] M. Kubat, R. C. Holte, and S. Matwin, "Machine Learning for the Detection of Oil spills in Satellite Radar Images," *Mach. Learn.*, vol. 30, no. 2–3, pp. 195–215, 1998.

[50] Liu, S. Liu, W. Cai, S. Pujol, R. Kikinis, and D. Feng, "Early diagnosis of Alzheimer's Disease with Deep Learning," *2014 IEEE 11th Int. Symp. Biomed. Imaging*, pp. 1015–1018, 2014.

[51] I. van der Waal, "Skin Cancer Diagnosed by Using Artificial Intelligence on Clinical Images," *Oral Dis.*, vol. 12, no. 10, pp. 3218–3221, 2017.

[52] K. M. Salama, A. M. Abdelbar, and A. A. Freitas, "Multiple pheromone types and other extensions to the Ant-Miner classification rule discovery algorithm," *Swarm Intell.*, vol. 5, no. 3–4, pp. 149–182, 2011.

[53] J. Uthayakumar, T. Vengattaraman, and P. Dhavachelvan, "Swarm intelligence based classification rule induction (CRI) framework for qualitative and quantitative approach: An application of bankruptcy prediction and credit risk analysis," *J. King Saud Univ. - Comput. Inf. Sci.*, 2017.

[54] C. Wu, "Ant colony optimization on building an online delayed diagnosis detection support system for emergency department," in *The 7th International Conference on Computational Intelligence in Economics and Finance (CIEF 2008)*, 2008, pp. 1–7.

[55] D. Martens, B. Baesens, and T. Fawcett, “Editorial survey: Swarm intelligence for data mining,” *Mach. Learn.*, vol. 82, no. 1, pp. 1–42, 2011.

[56] K. M. Salama, A. M. Abdelbar, F. E. B. Otero, and A. A. Freitas, “Utilizing multiple pheromones in an ant-based algorithm for continuous-attribute classification rule discovery,” *Appl. Soft Comput. J.*, vol. 13, no. 1, pp. 667–675, 2013.

[57] I. Chorbev, D. Mihajlov, and I. Jolevski, “Web based medical expert system with a self training heuristic rule induction algorithm,” in *Proceedings - 2009 1st International Conference on Advances in Databases, Knowledge, and Data Applications, DBKDA 2009*, 2009, pp. 143–148.

[58] A. Kumar, “Cascaded clustering and Ant-Miner based classification,” *Int. J. Adv. Educ. Res.*, vol. Volume 3, no. Issue 1, pp. 438–441, 2018.

[59] P. Jaganathan, K. Thangavel, A. Pethalakshmi, and M. Karnan, “Classification rule discovery with ant colony optimization and improved quick reduct algorithm,” *IAENG Int. J. Comput. Sci.*, no. February, pp. 286–291, 2007.

[60] I. Michelakos, E. Papageorgiou, and M. Vasilakopoulos, “A hybrid classification algorithm evaluated on medical data,” *Proc. Work. Enabling Technol. Infrastruct. Collab. Enterp. WETICE*, pp. 98–103, 2010.

[61] J. Smaldon, “Data mining with an ant colony optimization algorithm,” 2006.

[62] L. Yang, K. S. Li, W. S. Zhang, Y. Wang, and Z. X. Ke, “Application of a New mAnt-MinerPR Algorithm in Classification Rule Mining,” *J. Digit. Inf. Manag.*, vol. 13, no. 5, 2015.

[63] R. Robu, S. Holban, and N. Robu, “Ant Miner Extensions for Improved Data Classification,” *Int. J. Comput. Eng. Appl.*, vol. IX, no. Vii, pp. 135–154, 2015.

[64] J. R. Quinlan, “Generating production rules from decision trees,” *Proc. Tenth Int. Jt. Conf. Artif. Intell.*, vol. 30107, pp. 304–307, 1987.

[65] Z. Liang, J. Sun, Q. Lin, Z. Du, J. Chen, and Z. Ming, “A novel multiple rule sets data classification algorithm based on ant colony algorithm,” *Appl. Soft Comput.*, vol. 38, pp. 1000–1011, 2016.

[66] Z. W. Geem, J. H. Kim, and G. V Loganathan, “A new heuristic optimization algorithm: Harmony search,” *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[67] H. Schwefel, “On the Evolution of Evolutionary Algorithms,” *Genet. Program. 7th Eur. Conf. EuroGP 2004 (Coimbra, Port. April 5-7, 2004)*, vol. 3003, pp. 389–398, 2004.

[68] Z. Hao, R. C. Cai, and H. Huang, “An adaptive parameter control strategy for ACO,” *Proc. 2006 Int. Conf. Mach. Learn. Cybern.*, no. August, pp. 203–206, 2006.

[69] R. Robu, C. Vacar, N. Robu, and S. Holban, “A study on Ant Miner parameters,” in *IISA 2015 - 6th International Conference on Information, Intelligence, Systems and Applications*, 2016.

[70] T. Stutzle, M. Lopez-Ibnez, P. Pellegrini, M. Maur, M. M. de7 Oca, M. Birattari, and M. Dorigo, “Parameter Adaptation in Ant Colony Optimization,” Bruxelles,Belgium, 2010.

[71] T. Stutzle, M. Lopez-Ibnez, P. Pellegrini, M. Maur, M. M. De Oca, M. Birattari, and M. Dorigo, “Parameter Adaptation in Ant Colony Optimization,” in *Autonomous search*, 2012, pp. 191–215.

[72] W. Liao, “Enterprise Data Mining : A Review and Research Directions,” in *Recent Advances In Data Mining Of Enterprise Data: Algorithms And Applications*, 2008, pp. 1–109.