Comparative Analysis of Ant miner And Ant miner+ Algorithms based on breast cancer and tic-tac-toe data sets

Veena Dahiya, Chhavi Rana
UIET, M.D.U, Rohtak, Haryana(124001) India.
veenasaini541@gmail.com
chhavi1jan@yahoo.com

***Abstract*: This work describes a new algorithm for classification, named AntMiner+, based on an artificial ant system with inherent self-organizing capabilities. The usage of ant systems generates scalable data mining solutions that are easily distributed and robust to failure. AntMiner+ uses a *MAX-MIN* ant system which is an improved version of the originally proposed ant system, yielding better performing classifiers. The algorithm used to discover such rules is inspired in the behavior of a real ant colony, as well as some concepts of information theory and data mining. We compare the performance of Antminer+ with Antminer , a well known data mining algorithm for classification, in two datasets.**

***Keywords*: Ant miner+, Data mining, Classification.**

1. INTRODUCTION

In recent decades, innovative storage technologies and the success of the Internet have caused a true explosion of data. This data is typically distributed, continuously updated and contains valuable, yet hidden knowledge. Data mining is the overall process of extracting knowledge from this raw data. Although many techniques have been proposed and successfully implemented, few take into account the importance of the comprehensibility aspect of the generated models or the ability to deal with distributed data. Artificial ant systems are inspired on real ant colonies and are specifically designed to provide robust, scalable and distributed solutions. By performing local actions and indirect communication only, ants are able to achieve complex overall behavior. The approach described in this paper, named AntMiner+, takes advantage of the inherent benefits of ant systems and puts them in a data mining context.

Ant miner+ is a classification technique which is based on the principles of ant colony optimization. Ant miner+ by Martens et al. (2007) is an extension of Ant miner using MAX-MIN Ant System algorithm (Stutzel & Hoos , 2000). The method can handle ordinal attributes but it cannot handle numeric attributes. The goal is to inter comprehensible rule based classification modals from a data set.[1]

The remainder of this chapter is structured as follows. In section 2 we define social inspects and real ant system. Next, we examine the related work in section 3 and in section 4; we examine the overview of Ant miner+. The experiments and results are explained in section 5. Finally we conclude and provide some recommendations for future work in section 6.

2. SOCIAL INSECTS AND REAL ANT SYSTEMS

In a colony of social insects like ants, bees, wasps and termites work by themselves in their simple task, independently of others members of the colony. However the tasks performed by different aspects are related to each other in such a way that the colony, as a whole, is capable of solving problem through cooperation. Important, survival related problems such as selecting and picking up materials, finding and storing food, which require sophisticated planning, are solved by insect colonies without any kind of supervisor or centralized controller. This collective behavior which emerges from a group of social insects has been called "swarm intelligence". [2]

Ants are able to find shortest path between a food source and the nest without the aid of visual information, and also to adapt a changing environment. [3] It was found that the way ants communicate with each other to find the right way to follow is based on pheromone trails. While ants move, they drop a certain amount of pheromone on the floor, leaving

Behind a trail of this substance that can be followed by other ants. The more ants follow a given trail, the more attractive this trail becomes to be followed by other ants. This process can be described as a loop of positive feedback, in which the probability that an ant choose a path is proportional to the number of ants that have already passed by that path.

When an established path between a food source and the ants nest is distributed by the presence of an object, ants soon will try to go around the obstacle. The basic idea is illustrated in figure 1.

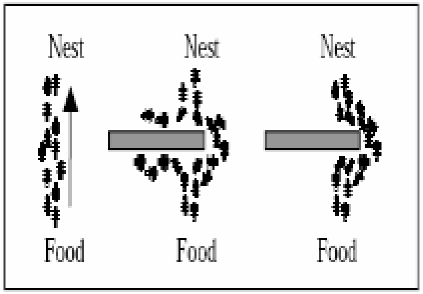

Fig.1: Ants find the shortest path around an obstacle

Firstly, each ant can choose to go around to the left or to the right of the object with a 50%-50% probability distribution. All ants move roughly at the same speed and deposit pheromone in the trail at roughly the same rate. However, to go around the longer path going an ant takes more time than going by the shorter path. This makes the pheromone to the accumulated faster in the shorter path than in the longer one. Besides, ants prefer to follow paths with more pheromone leading to a faster convergence to the shorter path.

3. RELATED WORK

1. David Martens, Manu De Backer, Raf Haesen, 2007[4] anaiyses & propoed of two aspects of Antminer. On the one hand, they provide an overview of previous ant-based approaches to the classification task and compare them with state-of-the-art classification techniques, such as C4.5, RIPPER, and support vector machines in a benchmark study. On the other hand, a new ant-based classification technique is proposed, named Ant Miner+. The key differences between the proposed Ant Miner+ and previous Ant Miner versions are the usage of the better performing MAX – MIN ant system, a clearly defined and augmented environment for the ants to walk through, with the inclusion of the class variable to handle multi class problems, and the ability to include interval rules in the rule list. Furthermore, the commonly encountered problem in ACO of setting system parameters is dealt with in an automated, dynamic manner. Our benchmarking experiments show an Ant Miner+ accuracy that is superior to that obtained by the other Ant Miner versions, and competitive or better than the results achieved by the compared classification techniques.

2. ABDUL RAUF BAIG, WASEEM SHAHZED, SALABAT KHAN AND FARIHA ALTAF ,2011 [5] proposed reports some improvements in a recently proposed ACO based classification algorithm, called C Ant Miner, whose main feature is a heuristic function based on the compatibility of pairs of attribute-values and class labels, and its application on medical datasets. We study the performance of the algorithm for twelve commonly used datasets and compare it with ten well known classification algorithms, three of which are ACO based. Experimental results show that the accuracy rate obtained by Cant Miner is better than that of the compared algorithms. We also discuss some other issues related to comprehensibility of the classifier building process.

3. Gopinath Chennupati, 2014 [6**]**propose**d** an acclaimed machine learning technique named, ensemble of classifiers is applied, where an ACO classifier is used as a base classifier to prepare the ensemble. The main trade-off is, the predictions in the new approach are determined by discovering a group of models as opposed to the single model classification. In essence, we prepare multiple models from the randomly replaced samples of training data from which, a unique model is prepared by aggregating the models to test the unseen data points. The main objective of this new approach is to increase the stability of the Ant-Miner results there by improving the performance of ACO classification. We found that the ensemble Ant-Miners significantly improved the stability by reducing the classification error on unseen data.

4 .Matej Piculin , Marko Robnik-Sikonja, 2014[7] propose an adaptation of Ant Miner+ for rule mining which intrinsically handles numeric attributes. We describe the new approach and compare it to the existing algorithms. The proposed method achieves comparable results with existing methods on UCI datasets, but has advantages on datasets with strong interactions between numeric attributes. We analyze the effect of parameters on the classification accuracy and propose sensible defaults. We describe application of the new method on a real world medical domain which achieves comparable results with the existing method.

4.OVERVIEW OF ANT MINER+

Ant miner+ is based on the previous Ant Miner versions; the main novelties implemented in Ant Miner are as follows:

• Environment:

—The environment is defined as a directed acyclic graph (DAG), so that the ants can choose their paths more effectively.

—To allow for interval rules, the construction graph additionally exploits the difference between nominal and ordinal variables.

—Inclusion of the weight parameters for the pheromone and heuristic value in the construction graph.

• Implementation of the better performing MIN – MAX Ant System.

• The usage of more accurate class-specific heuristic values, with the exclusion of the majority class to be used as the final default class.

• Application of an early stopping criterion.

4.1 Ant miner+ algorithm

Construct graph
While (not early stopping)
Initialize heuristics, pheromones and
Probabilities of edges
While (not converged)
Create ants
Let ants run from source to sink
Evaporate pheromone on edges
Prune rule of best ant
Update path of best ant
Adjust pheromone levels if outside boundaries
Kill ant
Update probabilities of edges
End
Extract rule
Flag data points covered by the extracted rule
End
Evaluate performance on test set

Pseudo code of Ant miner+ algorithm

The main workings of Ant Miner are described in pseudo code. First, a directed acyclic construction graph is created that acts as the ants' environment. All ants begin in the *START* vertex and walk through their environment to the *STOP* vertex, gradually constructing a rule. Only the ant that describes the best rule will update the pheromone of its path. Evaporation decreases the pheromone of all edges. A supplementary modification of the pheromone levels may be needed since the - approach additionally requires the pheromone levels to lie within a given interval. Since the probabilities are the same for all ants in the same iteration, they are calculated only once, at the beginning of the iteration. Convergence occurs when all the edges of one path have a pheromone level *T*max and all other edges have pheromone level *T*min, next the rule corresponding to the path with *T*max extracted and the training data covered by this rule is removed from the training set. This iterative process will be repeated until early stopping occurs or until none of the ants describe a rule that covers at least one training point. In the latter case, no rule can be extracted as all the paths have a quality of zero. This will typically be attributed to an abundance of noise in the remaining data, indicating that further rule induction is useless.

4.2 Construction graph

A first modification to allow for interval, such as *Age* ϵ [20, 60], to be include in the rule is the expansion of the set of n variables {V1, V2,…., Vn} in to a set of m variables by taking the ordinal variable twice. This result in the new set of variables $\{V_1^*, V_2^*, \ldots, V_m^*\}$. With m=|VNom|+2.|VOrd|.

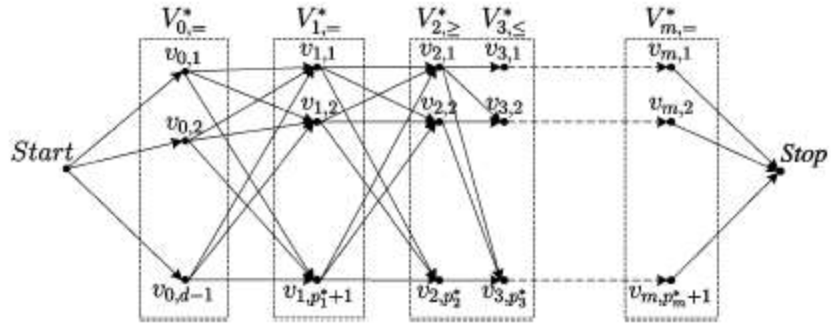

Fig.2: Construction graph of Antminer+

Vertices: firstly, the root and sink vertices Start and Stop are defined.
Second as just for the Ant miner construction graph G, we define a vertex group for each of the variables V_i^*, i.e. a vertex υi,j is created for all values *Valuei,j* of the variables V_i^* i=1, 2……m. The first group of vertices defined for an ordinal variable V_i^*, are υi,j(j=1,2,…p_i^*) with p_i^* the number of value for variables V_i^*,. The vertex group should be regarded as the lower bound for variable V_i^*, while the second vertex group υi+1,j(j=1,2,…, p_i^* +1) Should be seen as the upper bound for V_i^* +1, which is the ordinal case is equal to V_i^*.

Since ants have to pass a vertex for each variable before reaching the final STOP vertex, a dummy vertex υi,p_i^*+1 is defined for each nominal variable V_i^*. The value for this vertex is undetermined, implying that any ant choosing a dummy vertex does not make use of the variable V_i^*. In its rule. For the ordinal variables, such a dummy vertex is unnecessary since an ant choosing the first vertex and then the last vertex [thus describing $V_i^* \geq$ min(V_i^*) **and** $V_i^* \leq$ max(V_i^*)] also makes no use of the variable.

To allow the ants to choose the class for which to extract a rule, an extra vertex group is added that comes first in the construction graph. This is similar to considering the class variable as just one of the variables, and will be treated as such when calculating the heuristic values. Therefore, we introduce a new variable V_0^* that corresponds to the class variable. To extract a rule list that is complete, such that all future data points can be classified, one class is not included in the construction graph and will be the predicted class for the final **else** clause. Hence, the class variable corresponds to a vertex group with d-1 vertices. The class to exclude is taken to be the majority class since

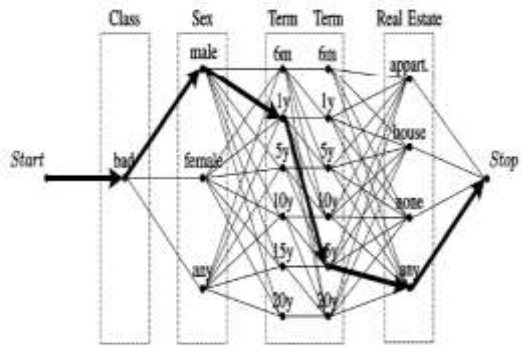

Fig. 3: Example of path described by an ant for the construction graph defined By Ant Miner+

the smaller classes are usually the classes of interest: the bad credit customers, the malignant breast masses Additionally, this prevents a path with all dummy vertices, which corresponds to a rule predicting the majority class in all cases, to be chosen by the ants. As our experiments will show, excluding one class from the class vertex group results in an improved accuracy.

Correspondence: The correspondence is Γ^{+} defined as

$$
\begin{aligned}
\Gamma^{+}(Start) &= v_{0,j} && j = 1,2,\ldots,d-1 \\
\Gamma^{+}(v_{i,j}) &= v_{i+1,k} && i = 0,1,\ldots,m-1 \\
& && k = 1,2,\ldots,p^{*}_{i+1}+1 \quad \text{if } V^{*}_{i+1} \text{ nom} \\
& && k = 1,2,\ldots,p^{*}_{i+1} \quad \text{if } V^{*}_{i+1} \text{ ord} \\
\Gamma^{+}(v_{m,j}) &= Stop && j = 1,2,\ldots,p^{*}_{m}+1 \quad \text{if } V^{*}_{m} \text{ nom} \\
& && j = 1,2,\ldots,p^{*}_{m} \quad \text{if } V^{*}_{m} \text{ ord.}
\end{aligned}
$$

To avoid that ants incorporate inconsistencies in their rule, such as age>=30 and age<=20 we further constrain the environment by removing the edge between any two vertices corresponding to the same ordinal variable, where the first vertex has a higher value than the second one. Therefore, in Figure. 2, among others, the edge between υ2,2 and υ3,1 is missing. This graph is a rooted DAG with a complexity of $O(m \cdot avg^{2})$. . Although is a value slightly higher than n(n<=m<=2-n) , the complexity is clearly lower than the complexity of the construction graph defined by Ant Miner. Notice that on average, the ants in Ant Miner+ need to choose among avg.n edges at each decision point, where for Ant Miner+ they choose among edges. By reducing the choices without constraining the rule format, Ant Miner ants can make their decisions more effectively. A credit scoring example of the construction graph described so far is shown in Figure. 3. Assume we have three variables: sex of the applicant, term of the loan, and information concerning real estate property of the applicant. The dummy vertices for the nominal variables sex and real estate have the corresponding “any” value.

An ant that follows the path indicated in bold from START to STOP describes the rule:

If Sex=male and Term>=1y and Term <=15y

Then customer=Bad

2.3 Edges probabilities

The probability for an ant in vertex υi-1, *k* to choose the edges to vertex υi,j is defined by the pheromone value (*T(υi*-1,*k*,*υi,j*) of the edge, the heuristic value(*ηυi,j*) of vertex υi,j , and normalized over all possible edge choices.

$$P_{ij}(t) = \frac{[\tau_{(v_{i-1,k},v_{i,j})}(t)]^{\alpha} \cdot [\eta_{v_{i,j}}(t)]^{\beta}}{\sum_{l=1}^{p_i}[\tau_{(v_{i-1,k},v_{i,l})}(t)]^{\alpha} \cdot [\eta_{v_{i,l}}(t)]^{\beta}}.$$

For the Ant Miner edge probabilities, we need to consider and normalize over all variables *Vi (i=1,…,n)*, while for Ant Miner+, we only need to consider and normalize over the next variable Vi^{*} . The α and β are weight parameters that indicate the relative importance of the pheromone and heuristic value.

4.4 Heuristic Value: The heuristic value gives for each vertex in the construction graph a notion of its quality and importance in the problem domain. For the classification task, we define the importance of a vertex υi,j, with value *Valuei,j* for variable Vi^{*} , by the fraction of training cases that are correctly covered by this term,

$$\eta_{v_{i,j}}(t) = \frac{|T_{ij} \ \& \ CLASS = class_{ant}|}{|T_{ij}|}$$

Since the heuristic value is dependent on the class chosen by the ant, each vertex has as many heuristic values as there are class values. This heuristic value for each possible class is more accurate than using just one heuristic for a vertex. Ant Miner+ ants already know the class for which to extract a rule, such a class-specific and accurate heuristic function can be used.

For dummy vertices, the same formula is used to calculate the heuristic value. As *Ti,j* stands for *Vi=any* for a dummy vertex, the heuristic value is simply the percentage of uncovered training points that have the class *classant* . For the class vertices, a similar value is chosen, that is: the ratio of uncovered training points that have class equal to the class of the vertex. So, if 60% of the uncovered training data is of class 1, the heuristic value for the vertex will be 0.6.

2.4 Pheromone Updating

Updating the pheromone trail of the environment of MIN- MAX Ant Systems is accomplished in two phases: evaporation and reinforcement. Evaporation is accomplished by diminishing the pheromone level of each trail by a factor ρ. Typical values for this evaporation factor ρ i.e. in the range [0.8, 0.99] .

$$\tau_{(v_{i,j},v_{i+1,k})}(t+1) = \rho \cdot \tau_{(v_{i,j},v_{i+1,k})}(t) + \frac{Q^{+}_{best}}{10}$$

Reinforcement of the pheromone trail is only applied to the best ant's path. The best ant can be chosen as either the iteration best ant, or the global best ant. Results described in the literature motivate our choice towards the iteration best ant. This means that, taking into account the evaporation factor as well, the update rule for the best ant's path is described, where the division by ten is a scaling factor that is needed such that both the pheromone and heuristic values lie within the range [0, 1].

Clearly, the reinforcement of the best ant's path should be proportional to the quality of the path Q^{+}_{best} , which we define as the sum of the *confidence* and the *coverage* of the corresponding rule. Confidence measures the fraction of remaining (not yet covered by any of the extracted rules) data points covered by the rule, that are correctly classified. The coverage gives an indication of the overall importance of the specific rule by measuring the number of correctly classified remaining data points over the total number of remaining data points. More formally, the pheromone amount to add to the path of the iteration best ant is given by the benefit of the path of the iteration best ant with *ruleant*, the rule antecedent (if part) being a conjunction of terms corresponding to the path chosen by the ant, $rule^{c}_{ant}$ the conjunction of *ruleant* with the class chosen by the ant, and *Cov* a binary variable expressing whether a data point is already covered by one of the extracted rules (*Cov=1*) or not (*Cov=0*). The number of remaining data points can, therefore, be expressed as |*Cov=0*|

$$Q^{+} = \underbrace{\frac{|rule^{c}_{ant}|}{|rule_{ant}|}}_{\text{confidence}} + \underbrace{\frac{|rule^{c}_{ant}|}{|Cov=0|}}_{\text{coverage}}.$$

2.5 Weight Parameters

An issue typically encountered when applying ACO algorithms is the need to instantiate several system parameters, such as the number of ants, the evaporation factor, and the weight parameters α and β. Ant Miner+ allows other values to be chosen and actually lets the ants themselves choose suitable values. This is done by introducing two new vertex groups in the construction graph: one for each weight parameter. We limit the values for the weight parameters to integers between 1 and 3, which typically provided values around α=2 and β=1 . Since we do not have a specific preference, the heuristic values for these vertices are all set to 1, so the search is only influenced by the pheromone levels. The final construction graph, with the inclusion of weight parameters α and β, and the class variable *VO* is provided in Figure: 4.

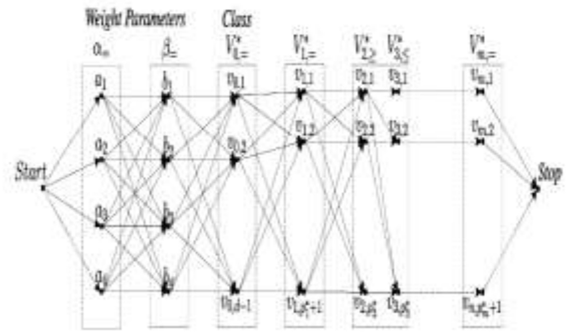

Fig.4: Multiclass construction graph of Ant Miner+ with the inclusion of weight parameters.

4.7 Pruning

Rule pruning is a technique used to improve the generalization behavior of a rule-based classifier by removing irrelevant terms from a rule [8]. When all the ants have created their rule, the rule of the best ant is selected for pruning. Suppose this rule has t terms, then all possible rules with one term less, thus t-1 rules are created and the corresponding confidence is calculated. The rule that experiences the highest confidence increase is selected for further pruning. This process is repeated until none of the rules with one term less have a confidence that is higher or the same. When a rule is pruned, a better rule is obtained, both in terms of accuracy (as only confidence is considered, and not, the coverage) and comprehensibility (the rule is less complex since it consists of one term less). The path corresponding to this pruned path will be updated next.

4.8 Early Stopping

Ant Miner+ stops constructing rules when either a predefined percentage of training points has been covered by the inferred rules or when early stopping occurs.

The ultimate goal of Ant Miner+ is to produce a model which performs well on new, unseen test instances. If this is the case, we say that the classifier generalizes well. To do so, we basically have to avoid the classifier from fitting the noise or idiosyncrasies in the training data. This can be realized by monitoring the error on a separate validation set during the training session. When the

error measure on the validation set starts to increase, training is stopped, thus effectively preventing the rule base from fitting the noise in the training data. This stopping technique is known as *early stopping*. Note that this causes loss of data that cannot be used for constructing the rules and hence, this method may not be appropriate for small data sets. Implementing the early stopping criterion involves the separation of the data set in three parts: a training set used by Ant Miner+ to infer rules from, a validation set to implement the early stopping rule, and a test set to calculate an unbiased performance estimate. As a rule of thumb, two third of the complete dataset is typically used for training and validation and the remaining third for testing. Of the data points set aside for training and validation, two third is training data and one third validation data.
As the figure shows, early stopping makes sure the rules generalize well. Going beyond the induced stop increases the training accuracy, yet lowers the validation (and also test) accuracy because over fitting occurs.

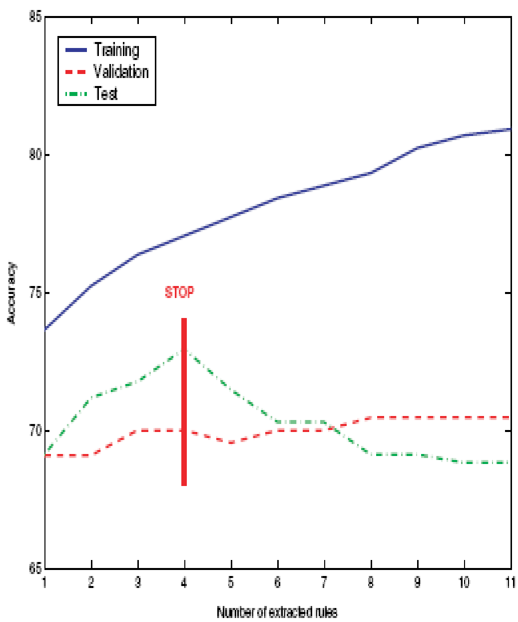

Fig. 5: Illustration of the early stopping rule

5. EXPERIMENTS AND RESULTS

5.1 Experimental Set-Up

Ant Miner+ has been applied to a wide range of publicly available data sets, corresponding to both binary and multi class classification problems [9]. Training, validation, and test set are determined in the manner discussed before. To eliminate any chance of having unusually good or bad training and test sets, 10 runs are conducted where the data is first randomized before the training, validation and test set are chosen.

In all the experiments reported in this paper these parameters were set as follows:

- No _of_ ants =1000;
- Min _cases _per_ rule =10;
- Max _ uncovered _cases =10;
- No _ rules _ converge =10.

We have made no serious attempt to optimize the setting of these parameters. Such an optimization should be tried in future research. It is interesting to notice that even the above non-optimized parameters settings has produced quite good results. In addition, the fact that Ant miner+ parameters were not optimized for the data sets used in over experiments makes the comparison with Ant miner fair, since we used the default, non-optimized parameters for Ant miner as well.

5.2 Datasets

Ant Miner+ has been applied to a wide range of data sets. Two datasets concern medical diagnosis of breast cancer and other datasets used concern the tic-tac-toe toy problem. Both the tic-tac-toe and the breast cancer datasets come from the publicly available UCI data repository.

Brest cancer: Two different datasets of breast cancer. One obtained from the University Of Wisconsin Hospitals, Madison from Dr. William H. Wolberg and the other from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia and is provided by M. Zwitter and M. Soklic . We use Ljubljana breast cancer dataset.

Tic-tac-toe: Also included in our experiments is the *tic-tac-toe* dataset, which encodes the complete set of possible board configurations at the end of tic-tac-toe games where X is assumed to have played first. The target concept is 'win for X'.

5.3 Result

Among the several criteria that could be used to evaluate the predictive accuracy of discovered rules, the cross-validation accuracy rate was used. Although this measure is computationally expensive, it gives a wide exploration of the characteristics of the cases in the dataset. For all datasets, a 10-fold cross validation ($k=10$) was used. In this procedure, all cases are used only once as testing and (k-1) times as training. The final accuracy rate is simply the average of the accuracy rate of the k iterations. All the k data partitions are randomly generated considering all available cases.

Table 1 summarizes the results obtained by the proposed Ant Miner algorithm in the two datasets. The table shows the accuracy rate, the number of rules found and the number of terms (the shown values are the average values of the cross-validation procedure followed by the corresponding standard deviation).

Table 1: Results with the Ant miner algorithm

Data sets	Predictive accuracy	Number of rules	Number of conditions
Ljubljana Breast Cancer	75.93% +/- 2.38%	6.1 +/- 0.1	7.9 +/- 0.31
Tic- Tac-Toe	69.74% +/- 1.76%	10.8 +/- 0.29	15.9 +/- 0.48

The obtained results were compared with other machine learning methods found in the literature, for the same datasets. Table 2 compress the accuracy rate of Ant miner with the accuracy rate of the well known Ant miner + using 10 fold cross validation procedure for both algorithms.

	Accuracy	Number of Rules	Number of Conditions
Ljubljana Breast Cancer Data Set			
Ant Miner	75.93% +/- 2.38%	6.1 +/- 0.1	7.9 +/0.13
Ant Miner+	77.47	3.4	3.2
Tic-Tac-Toe Data Set			
Ant Miner	69.74% +/- 1.76%	10.8 +/- 0.29	15.9 +/- 0.48
Ant Miner +	99.75	9.0	3.0

6. Conclusion and future work

AntMiner+ is a technique that successfully incorporates swarm intelligence in data mining. Using a *MAX-MIN* system, AntMiner+ builds comprehensible, accurate rule-based classifiers that perform competitively with state-of-the-art classification techniques. Although ants have a limited memory and perform actions based on local information only, the ants come to complex behavior due to self-organization and indirect communication. The intrinsic properties of ant systems allow us to easily distribute our approach to provide robust and scalable data mining solutions. Still, several challenges lie ahead. Most real-life datasets contain various continuous variables. One approach to deal with this is by categorizing these variables in a pre-processing step. Incorporating the variables in a dynamically changing construction graph is another option and focus of current research. An issue faced by ant systems, is the necessity to instantiate various parameters, such as the weight parameters α and β. These parameters are typically determined by trial and error or with more fancy techniques such as genetic algorithms or local search techniques. We are currently investigating the possibility of including these two variables in the construction graph, which would have the supplementary benefit of generating dynamically changing parameters as the environment changes. Once again, the ants will take over the work from the user.

References

[1] D. Martens et al.” Ants Constructing Rule-Based Classifiers” Studies in Computational Intelligence (SCI), vol. 34, pp.21–43 ,2006
[2] E. Bonabeau, M. Dorigo and G. Theraulaz, Swarm Intelligence “From Natural to Artifical Systems” New York, NY: Oxford University Press, 1999.
[3] M. Dorigo, A. Colorni, and V. Maniezzo .” The ant system: optimization by a colony of cooperating agents” IEEE Transactions on Systems, Man, and Cybernetics-Part B, vol. 26, no 1, pp. 1-13,1996.
[4] David Martens, Manu De Backer, Raf Haesen, Student Member, IEEE, Jan Vanthienen, Monique Snoeck, and Bart Baesens” Classification With Ant Colony Optimization”, vol. 11, no. 5, 2007
[5] Abdul Rauf Baig, Waseem Shahzad, Salabat Khan and Fariha Altaf” ACO BASED DISCOVERY OF COMPREHENSIBLE AND ACCURATE RULES FROM MEDICAL DATASETS” International Journal of Innovative Computing, Information and Control , vol. 7, no. 11, 2011
[6] Gopinath Chennupati” eAnt-Miner : An Ensemble Ant-Miner to Improve the ACO Classification” ,2014
[7] Matej Picˇulin , Marko Robnik-Šikonja” Handling numeric attributes with ant colony based classifier for medical decision making” Expert Systems with Applications 41, pp.7524–7535, 2014
[8] P.-N. Tan, M. Steinbach, and V. Kumar” Introduction to Data Mining. Reading", MA: Addison Wesley, 2005.
[9] S. Hettich and S. D. Bay, The UCI KDD Archive. Irvine, CA: Dept. Inf. Comput. Sci., Univ. California, 1996 [Online]. Available: http:// kdd.ics.uci.edu.

