

Ant-Inspired Algorithms for Decision Tree Induction

An Evaluation on Biomedical Signals

Miroslav Bursa¹(✉) and Lenka Lhotská²

¹ Czech Institute of Informatics, Robotics and Cybernetics,
Czech Technical University in Prague, Prague, Czech Republic
bursam@ciirc.cvut.cz

² Department of Cybernetics, Faculty of Electrical Engineering,
Czech Technical University in Prague, Prague, Czech Republic
lhotska@fel.cvut.cz

Abstract. In this paper we present an evaluation of ant-inspired method called *ACO-DTree* over biomedical data. The algorithm maintains and evolves a population of decision trees induced from data. The core of the algorithm is inspired by the Min-Max Ant System.

In order to increase the speed of the algorithm we have introduced a local optimization phase. The generalization ability has been improved using error based pruning of the solutions.

After parameter tuning, we have conducted experimental evaluation of the *ACO-DTree* method over the total of 32 different datasets versus 41 distinct classifiers. We conducted 10-fold crossvalidation and for each experiment obtained about 20 quantitative objective measures. The averaged and best-so-far values of the selected measures (precision, recall, f-measure, . . .) have been statistically evaluated using Friedman test with Holm and Hochberg post-hoc procedures (on the levels of $\alpha = 0.05$ and $\alpha = 0.10$). The *ACO-DTree* algorithm performed significantly better ($\alpha = 0.05$) in 29 test cases for the averaged f-measure and in 14 cases for the best-so-far f-measure.

The best results have been obtained for various subsets of the UCI database and for the dataset combining cardiocotography data and data of myocardial infarction.

1 Introduction

In many industrial, business, healthcare and scientific areas we can see still growing use of computers and computational resources. Together with the boom of high-speed networks and increasing storage capacity of database clusters, data warehouses and cloud technologies, a huge amount of various data can be stored. Such data are often mixed from different source, containing different data types, unusual coding schemes, and seldom come without any errors (or noise). *Data mining* is not only an important scientific area, but also an important tool in industry, business and healthcare as the people making decision are becoming

overwhelmed by the data. As an example we can mention the Human Genome Project, Hubble Space Telescope and the Human Brain mapping Project. A nice overview of swarm intelligence for data mining can be found in [21].

Automatic construction of simple rules in the form of decision trees has been attempted virtually in all disciplines in which data exploration methods have been developed. It has been traditionally developed in the fields of statistics, engineering (pattern recognition) and decision theory (decision table programming). Recently renewed interest has been generated by research in artificial intelligence (machine learning) and neurosciences (neural networks) [22].

Multiple metaheuristics have been used for inducing decision trees. However, to our best knowledge we did find only a few successful attempts to use ant colony optimization method for the induction of decision trees (see Sect. 2.2 for more information).

2 Background Information

2.1 Ant Colony Optimization

The high number of individuals and the decentralized approach to task coordination in many ant species shows that ant colonies show high degree of parallelism, self-organization and fault tolerance. In studying these paradigms, there is a high chance to discover inspiration concepts for many successful metaheuristics.

Simple Ant Colony Optimization. The first ant colony optimization algorithm was the ant system [12]. This system has been improved and the first optimization algorithm available was the simple ACO (SACO) [10]. The SACO in fact implements the double bridge experiment [9]. It has been used for the TSP problem, where each edge is assigned a cost value (length) L . For the SACO each edge is also assigned a pheromone value τ_{ij} that is updated during the run. The pheromone value is used for inducing the potential solutions.

It has been observed that initial experiments on the binary bridge have rapidly converged to a solution and the ants did not explore alternative paths. Therefore an pheromone evaporation at the end of each iteration has been introduced.

Ant System. Ant System (AS) [11] improves SACO by changing the transition probability between nodes to include heuristic information and by the inclusion of a tabu list (thus adding a memory to the algorithm). The AS in fact balances the exploration and exploitation tradeoff that is controlled by the parameters α and β . If $\alpha = 0$, no pheromone is used, when $\beta = 0$, the algorithm degrades to SACO.

A different formulation of the transition probability has been defined in [20] where the α parameter defines a relative importance of the pheromone and removes the β parameter.

Ant Colony System. The Ant Colony System [13] has been developed as an improvement to performance of the AS. It uses a different transition rule, different pheromone update rule, local pheromone update and a candidate list to promote specific nodes.

The pseudorandom-proportional action rule is defined with a random number $r \sim U(0, 1)$ and an user-specified parameter $r_0 \in [0, 1]$. The r_0 parameter balances exploration ($r > r_0$) and exploitation ($r \leq r_0$).

In case of ACS, only the globally best ant is allowed to reinforce pheromone concentrations on the links (cf. AS). The authors implemented two methods of selecting the best path: (1) *iteration-best* where the pheromone is updated according to the best solution found in current iteration or (2) *global-best* where the pheromone is updated according to the best solution found ever.

Max-Min Ant System. It has been discovered that the AS prematurely stagnates for complex problems. All the ants followed exactly the same path and there was a very little exploration and too rapid exploitation of the highest pheromone concentrations. Therefore the Max-Min Ant System (MMAS) has been developed [28]. The main difference between AS and MMAS is that there is a limitation on the pheromone intensities, $\tau_{ij} \in [\tau_{min}, \tau_{max}]$ where the τ_{min} and τ_{max} are problem dependent static parameters. A positive lower bound τ_{min} facilitates better exploration as all the links have positive probability of being selected into solution. Additionally, only the best ant may reinforce pheromone. The firsts version of MMAS focused on the iteration-best path, later versions used different strategies (using combination of both strategies, reinitialization in case of stagnation, etc.).

We have tried all the ant metaheuristics mentioned above and finally, ended with a modified version of MMAS algorithm, including adaptive change of pheromone deposit and evaporation rate and the elitism.

More variants can be found in relevant literature, i.e. in the series of [14].

2.2 Decision Trees

Decision tree induction algorithms are used in many areas, such as astronomy, character recognition, finance, marketing, medicine, natural language processing, software engineering, etc. ([1]). The *decision trees* are hierarchical, sequential classification structures that recursively partition the set of observations (data). Each node of the decision tree is either a test that partitions the data, or a decision (classification) about the object. In this paper the decision trees are constructed automatically from the data.

Automated construction of decision trees (or decision rules) has been performed in many areas where data exploration methods have been developed.

The Problem of Construction. The construction of optimal decision trees (in the sense of minimal number of tests performed) is a NP-Complete problem ([17]). Authors [8] proved that even finding the root node in an optimal strategy is a NP-hard problem.

The algorithms for decision trees induction using ant colonies can be found in recent publications only, such as [24].

A comprehensive overview of decision trees can be found in [27].

Use of Metaheuristics for Tree Induction. The CART algorithm [5] is fully deterministic and it is prone to be stuck in local optima. The problem of traditional decision tree constructing algorithms is that it usually uses optimal split for a node only, but does not consider a group of data. Therefore a heuristic approach is suitable and provides more general results.

In 1990, Koza [18] introduced genetic programming for induction of decision trees. In 1993 [16] introduced an algorithm called SADT that uses a simulated annealing process to find oblique splits at decision node. Later in 1997, [22] introduced an OC1 system. It does not use the brute-force approach of SADT, but uses a random search only to improve the existing solution. Therefore it may be regarded as a hybridization of the CART and SADT technique. The classical approaches are very sensitive to incomplete and noisy data. A situation that is very usual in medical environment.

Evolutionary techniques have also been successfully implemented, i.e. [19]. More information can be found in [26]. It usually uses a population of trees, selecting the best according to a certain fitness function.

The authors in [1] propose to distinguish between (1) evolutionary induction of decision trees and (2) evolutionary design/improvement of decision tree components. In (1) the individuals are decision trees, in (2) the individuals are components of decision tree classifiers. Another logical possibility is to use evolutionary algorithms to optimize the parameters of an algorithm. In our methodology we use the first approach, therefore the decision trees are sometimes referred to as *individuals*.

An adaptive algorithm for ant-inspired induction of decision trees (ACDT) [3] uses so called twoing (binary) decision criterion or other decision rules that are not able to cope with continuous attributes. The authors proved that the algorithm is comparable to the classical CART algorithm. The authors later presented a modification using inequality rule for coping with continuous attributes [4]. However, similar approach has already been used in our publication few years earlier [6, 7]. For comparison the authors have used only 11 datasets from UCI repository and compared with an Ant-Miner algorithm [25] and the modification using continuous attributes [23].

The second ant-colony inspired induction algorithm can be found in [24]. It uses a virtual start node and two types of edges: nominal or continuous. In case of continuous attributes a dynamic discretization is used (as in C4.5 or cAnt Miner rule induction algorithm (of the same authors)). The authors also use the same heuristics as in C4.5 algorithm (information gain ratio for generating threshold values and further dividing the training sets into subsets). The pheromone is used as an accurate feedback of the quality. Authors evaluate their algorithm on 22 UCI datasets and compare them to weka's J48 and SimpleCART algorithm.

3 The *ACO-DTree* Method Description

We have used population approach with elitism. The potential solutions (decision trees) represent individuals. The pseudocode of the algorithm can be described as follows. The trees are induced from an incidence matrix (representing a memory of the algorithm) containing pheromone values. In the initial phase, the pheromone matrix is initialized either with random or constant values and the first population of trees is induced. In the iterative phase the population is first optimized (decision values in tree nodes are optimized) and then the population is increased and the worst solutions are removed. According to the elitist ratio, the best solutions contribute back to the pheromone matrix. Pheromone evaporation and adaptive measures keep the population diversified while keeping the pheromone matrix from being saturated (premature convergence).

Algorithm 1. Pseudocode of the creation and training of the *ACO-DTree* classifier

```

1: initialize ACO pheromone matrix
2: create initial population of decision trees
3: for all NUM_ITERATIONS do
4:   update adaptive values that depend on the iteration number
5:   if PSO_TRAIN_ITERATIONS elapsed then optimize population using the
      PSO algorithm [Parallel]
6:   end if
7:   increase the population with new PSO trained individuals [Parallel]
8:   prune the population [Parallel]
9:   remove the worst individuals too keep the population count constant
10:  update pheromone (elitist mode)
11:  evaporatePheromone
12:  if stopping criteria reached then
13:    BREAK
14:  end if
15: end for
16: save resulting classifier and related data

```

Fitness Measures of the Decision Trees. For each algorithm (over a dataset) the following outputs are saved for each partial crossvalidation run and for the final result (i.e. for 10-fold crossvalidation, 10 + 1 results are available). Only the most important are listed, the total count of measures saved is about 70:

- Classified/misclassified/unclassified instances (absolute and relative value)
- FN/FP/TN/TP (absolute and relative value)
- F-Measure
- Precision/Recall
- Area under ROC/PRC (AUROC/AUPRC)
- Confusion matrix
- Kappa statistic
- Various information scores (Kononenko & Bratko)
- Various SF measures (entropy gain, etc.)

4 Experimental Part

4.1 Methodology Overview

As we are comparing 73 different groups of classifiers over 31 datasets, we need to divide them into some natural clusters. The classifier groups are shown in the Table 1 (the *ACO_DTree* method is not listed as it is compared with all other classifiers). The data groups are shown in Table 2. The reason for such division is to obtain a detailed information about the proposed method behavior instead of displaying only the *All Classifiers over All datasets* result (displayed in the rightmost bottom cell of the evaluation tables) as it is usual in other publications.

We have used the 10-fold crossvalidation. The data were first randomized (shuffled) and then divided into 10-folds (using stratification).

Evaluation. There are many measures available for evaluation. In this paper evaluate the performance using the *precision* and *recall* measure. We can use (1) the average value (over 10 crossvalidation runs) or (2) maximal (BSF, best-so-far) value.

The statistical significance of the results will be assessed as follows. First we will use Friedman nonparametrical test to verify the hypothesis that mean values of the measures are equal (H_0). If the null hypothesis is rejected, we continue with post-hoc tests – Nemenyi test for comparing the classifiers against the others. If a statistically significant difference between performance of two classifiers is found, we proceed with Holm and Hochberg post-hoc tests. If we cannot reject the H_0 at $\alpha = 0.05$ we try at $\alpha = 0.1$.

4.2 Classifiers

The classifier groups are shown in the Table 1. The *Distinct instances* column shows a number of distinct classifiers implemented as a separate class. The *Total instances* contains also variations containing different strategies (i.e. different kernel function, etc.). The grouping is according to the WEKA datamining software Java classes.

4.3 Datasets

We have divided the datasets into the following groups for detailed evaluation: See Table 2.

- **UCI** – Contains datasets from the UCI database. This is a selected part of UCI database that is used in many articles from the domain of artificial intelligence. It contains 23 datasets in total.
- **ECG** – Contains MIT and AHA ECG databases, in total 6 datasets. These datasets are prepended with the **MIT_** or **AHA_** prefix.
- **CTG + MI** – Contains data of myocardial infarction (2 datasets) and cardiocography data (two datasets).

Table 1. Overview of the classifiers grouping for evaluation. Note that the *ACO-DTree* method is included in each group and is not listed explicitly.

Classifier	Distinct instances	Total instances
NaïveBayes	4	6
Functions	8	15
Lazy	4	15
Misc	4	8
Rules	11	12
Trees	10	16
Total	41	72

- **MIT** – Contains selected signals of the MIT ECG database together with the whole MIT database (total of 4 datasets). Does not contain the AHA ECG database. These datasets are prepended with the MIT_ prefix.
- **All** – Includes all datasets available. In total it is 32 datasets.

Table 2. Overview of the dataset grouping for evaluation. The groups are not exclusive, overlapping occurs. Therefore the total sum is not equal to the sum of all the subgroups.

Classifier	No. datasets
ECG	6
CTG and MI	4
ECG_MIT	3
All	16
Total	32

5 Results

This section summarizes the statistical evaluation of the experiments. Only the final results are included in this part (obtained after parameter tuning).

We have evaluated the average (mean) and BSF measures from 10-fold cross-validation. In order to present the results in compact form, the following symbols are used in the result tables:

- The \emptyset symbol means that the Friedman test did not reject the H_0 hypothesis¹.
- The symbols $\overset{R}{05}$ and $\overset{R}{10}$ mean that the Friedman test rejected the H_0 hypothesis on the level of α equal to 0.05 or 0.1 respectively. To save space we do not use decimals.

¹ The H_0 hypothesis states that all the classifiers performed the same.

– The symbols $\checkmark_{0.05}$ and $\checkmark_{0.10}$ indicate a significantly better results for the *ACO-DTree* method when compared to the other classifiers (by the Holm or Hochberg post-hoc test) on the level of α equal to 0.05 or 0.1 respectively.

Each cell of the table contains left and right part: The left side of each cell contains the statistical result symbol for an average value of the measure. BSF value of the measure is located in the right part of the cell (Table 3).

This kind of presentation is quite unusual, however it provides much more information than traditionally used approach when only the bottom-right cell is actually presented – the result for all classifiers vs. all datasets.

Note that each cell stands for the run of multiple classifiers (included the *ACO-DTree* method) over multiple datasets in order for the statistical tests to be valid.

Table 3. Statistical results for the *precision* measure (avg. and max) for the classifiers with optimized parameters.

Data \ Classifier: precision	Bayes	Funct.	Lazy	Misc	Rules	Trees	All
ECG dataset	\checkmark_{10} \emptyset	R_{05} \emptyset	R_{05} R_{05} \emptyset \emptyset	\emptyset \emptyset	R_{05} \emptyset	R_{05} \emptyset	R_{05} \emptyset
CTG and MI	\checkmark_{05} \emptyset	\checkmark_{05} \emptyset	\checkmark_{05} R_{05} \emptyset \emptyset	\checkmark_{05} \emptyset	\checkmark_{05} R_{05} \emptyset \emptyset	R_{05} \emptyset	\checkmark_{05} R_{05} \emptyset \emptyset
ECG MIT	R_{05} \emptyset	R_{05} \emptyset	R_{05} R_{05} \emptyset \emptyset	R_{05} \emptyset	R_{05} \emptyset	R_{05} \emptyset	R_{05} \emptyset
All datasets	\checkmark_{05} R_{05} \emptyset	\checkmark_{05} \emptyset	\checkmark_{05} \checkmark_{05} \emptyset \emptyset	\checkmark_{05} \emptyset	\checkmark_{05} \checkmark_{05} \emptyset \emptyset	R_{05} \emptyset	\checkmark_{05} \checkmark_{05} \emptyset \emptyset

5.1 Summary

Based on the statistical evaluation that is presented in Sect. 4 we can conclude that the *ACO-DTree* methods performs significantly better on the level of $\alpha = 0.05$ when compared over 32 different datasets and 41 distinct instances of classifiers. This overall result could be further divided for different measures and for various groupings of the datasets and/or classifiers (Table 4).

We conducted 10-fold crossvalidation and for each experiment saved and statistically evaluated about 20 quantitative objective measures. The averaged and best-so-far values of the selected measures have been statistically evaluated using Friedman test with Holm and Hochberg post-hoc procedures (on the levels of $\alpha = 0.05$ and $\alpha = 0.10$). The Nemenyi statistical test has also been conducted.

The *ACO-DTree* algorithm performed significantly better ($\alpha = 0.05$) in 29 test cases for the averaged f-measure and in 14 cases for the averaged precision measure. In case of the accuracy measure it was 13 and 15 cases respectively. The best results have been obtained for various subsets of the UCI database and for the dataset combining cardiocography data and for the data of myocardial infarction.

Table 4. Statistical results for the *recall* measure (avg. and max) for the classifiers with optimized parameters.

Data\Classifier: recall	Bayes	Funct.	Lazy	Misc	Rules	Trees	All
ECG dataset	R_{05}	R_{05}	R_{05}	R_{05}	R_{05}	R_{05}	R_{05}
CTG and MI	\checkmark_{05}	\checkmark_{05}	\checkmark_{05}	\checkmark_{05}	$\checkmark_{05} R_{10}$	R_{05}	\checkmark_{05}
ECG MIT	R_{05}	R_{05}	R_{05}	R_{05}	R_{05}	R_{05}	R_{05}
All datasets	$\checkmark_{05} \checkmark_{05}$	\checkmark_{05}	\checkmark_{05}	\checkmark_{05}	\checkmark_{05}	R_{05}	$\checkmark_{05} R_{05}$

6 Conclusion and Discussion

We have proposed, implemented and experimentally evaluated an ant-colony inspired method for induction of decision trees. We have also identified the areas where the *ACO_DTree* algorithm performed significantly better when compared to other state-of-the art classifier implementations. In addition to the accuracy measure we have evaluated also other quantitative measures. The advantage of the proposed method lies in the robustness achieved by replicating the self-organization behavior of ants that has been incorporated into an algorithm for induction of decision trees.

The algorithm uses randomization of the axis-parallel tree to improve the final solution. We have introduced a local search phase that leads to faster convergence of the algorithm. The local phase can be either random (jittering) or using another nature-inspired approach (PSO).

The algorithm has shown competitive results when compared to other classifiers. For the percent of correctly classified results (an average value) it has been statistically better in 4 cases for the whole UCI database and once for the ECG MIT and (CTG and MI) dataset on significance level $\alpha = 0.05$ and in one case for the whole ECG dataset ($\alpha = 0.10$).

More than 12400 experiments over 73 classifiers and 32 datasets has been run in the final evaluation phase.

6.1 Discussion

We are aware that in the branch of classifiers and decision trees, it is a common solution to use only the classifier accuracy for evaluation. In case of biomedical signals it is important to measure inter- and intra- patient results, together with measures used in medicine, such as SE and SP. In this project we have obtained about 20 different quantitative measures that can be used for evaluation. In addition, apart from the mean average value we can evaluate the best/worst-so-far solution found and even the standard deviation. The question is how this measures and their features should contribute to the final evaluation.

6.2 Future Work

It would be interesting to study (and extend) the algorithm to use hypercurves for splitting as opposed to axis-parallel splitting. The problem is that the task of finding such trees is much more complicated and the readability and comprehensibility of such tree is reduced. It is desirable to use sophisticated splits only when the extra complexity of the split is compensated by the contribution to tree quality.

Acknowledgment. This research project has been supported by the project number NT11124-6/2010 “Cardiotocography evaluation by means of artificial intelligence” of the Ministry of Health Care.

The research is supported by the project No. 15-31398A Features of Electromechanical Dyssynchrony that Predict Effect of Cardiac Resynchronization Therapy of the Agency for Health Care Research of the Czech Republic.

And I would like to acknowledge the UCI repository [15] and the relevant donors and creators of the datasets. This breast cancer data (Parp_BRLj) was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. Thanks go to M. Zwitter and M. Soklic for providing the data. The blood transfusion data (UCLbloodt) has been provided by Prof. I-Cheng Yeh [29]. The Breast Cancer Wisconsin dataset (Parp_BRWis) was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg [2]. CTG data originate from Faculty Hospital in Brno, Czech Republic. The CTG database is freely available online [30].

References

1. Barros, R., Basgalupp, M., de Carvalho, A., Freitas, A.: A survey of evolutionary algorithms for decision-tree induction. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **42**(3), 291–312 (2012)
2. Bennett, K., Mangasarian, O.: Robust linear programming discrimination of two linearly inseparable sets. *Optim. Methods Softw.* **1**(1), 23–34 (1992)
3. Boryczka, U., Kozak, J.: Ant colony decision trees – a new method for constructing decision trees based on ant colony optimization. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) *ICCCI 2010, Part I. LNCS*, vol. 6421, pp. 373–382. Springer, Heidelberg (2010)
4. Boryczka, U., Kozak, J.: An adaptive discretization in the ACDT algorithm for continuous attributes. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) *ICCCI 2011, Part II. LNCS*, vol. 6923, pp. 475–484. Springer, Heidelberg (2011)
5. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: *Classification and Regression Trees*. Wadsworth International Group, Belmont (1984)
6. Bursa, M., Lhotska, L.: Automated classification tree evolution through hybrid metaheuristics. In: Corchado, E., Corchado, J.M., Abraham, A. (eds.) *Innovations in Hybrid Intelligent Systems. AISC*, vol. 44, pp. 191–198. Springer, Heidelberg (2007)
7. Bursa, M., Lhotska, L., Macas, M.: Hybridized swarm metaheuristics for evolutionary random forest generation. In: *Proceedings of the 7th International Conference on Hybrid Intelligent Systems 2007, IEEE CSP*, pp. 150–155 (2007)

8. Cox, L.A., Quiu, Y., Kuehner, W.: Heuristic least-cost computation of discrete classification functions with uncertain argument values. *Ann. Oper. Res.* **21**(1), 1–30 (1989)
9. Deneubourg, J.L., Aron, S., Goss, S., Pasteels, J.M.: The self-organizing exploratory pattern of the argentine ant. *J. Insect Behav.* **3**, 159–168 (1990)
10. Dorigo, M., Di Caro, G.D.: New ideas in optimization. In: *The Ant Colony Optimization Meta-Heuristic*, pp. 11–32. McGraw-Hill (1999)
11. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. B* **26**(1), 29–41 (1996)
12. Dorigo, M.: Optimization, learning and natural algorithms. Ph.D. thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy (1992)
13. Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1**(1), 53–66 (1997)
14. Engelbrecht, A.P.: *Computational Intelligence: An Introduction*, 2nd edn. Wiley, New York (2007)
15. Frank, A., Asuncion, A.: UCI machine learning repository (2010). <http://archive.ics.uci.edu/ml>
16. Heath, D., Kasif, S., Salzberg, S.: Learning oblique decision trees. In: *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence IJCAI* (1993)
17. Hyafil, L., Rivest, R.L.: Constructing optimal binary decision trees is np-complete. *Inf. Process. Lett.* **5**(1), 15–17 (1976)
18. Koza, J.R.: Concept formation and decision tree induction using the genetic programming paradigm. In: Schwefel, H.-P., Männer, R. (eds.) *Parallel Problem Solving from Nature. LNCS*, vol. 496, pp. 124–128. Springer, Heidelberg (1991)
19. Llor, X., Garrell, J.M.: Automatic classification and artificial life models. In: *Proceedings of Learning Workshop* (2000)
20. Maniezzo, V., Colorni, A.: The ant system applied to the quadratic assignment problem. *IEEE Trans. Knowl. Data Eng.* **11**(5), 769–778 (1999)
21. Martens, D., Baesens, B., Fawcett, T.: Editorial survey: swarm intelligence for data mining. *Mach. Learn.* **82**, 1–42 (2011)
22. Murthy, K.: *On Growing Better Decision Trees from Data*. Ph.D. thesis, Johns Hopkins University, Baltimore, MD (1997)
23. Otero, F., Freitas, A., Johnson, C.: cAnt-Miner: an ant colony classification algorithm to cope with continuous attributes. In: *CIDM*, pp. 225–231 (2009)
24. Otero, F.E.B., Freitas, A.A., Johnson, C.G.: Inducing decision trees with an ant colony optimization algorithm. *Appl. Soft Comput.* **12**(11), 3615–3626 (2012)
25. Parpinelli, R.S., Lopes, H.S., Freitas, A.A.: Data mining with an ant colony optimization algorithm. *IEEE Trans. Evol. Comput.* **6**(4), 321–332 (2002)
26. Podgorelec, V., Kokol, P., Stiglic, B., Rozman, I.: Decision trees: an overview and their use in medicine. *J. Med. Syst.* **26**(5), 445–463 (2002)
27. Rokach, L., Maimon, O.: Top-down induction of decision trees classifiers - a survey. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **1**(11), 1–12 (2002)
28. Stutzle, T., Hoos, H.: Max-min ant system. *Future Gener. Comput. Syst.* **16**(8), 889–914 (2000)

29. Yeh, I.C., Yang, K.J., Ting, T.M.: Knowledge discovery on RFM model using Bernoulli sequence. *Expert Syst. Appl.* **36**(3), 5866–5871 (2009). <http://dx.doi.org/10.1016/j.eswa.2008.07.018>
30. Chudacek, V., Spilka, J., Bursa, M., Janku, P., Hruban, L., Huptych, M., Lhotska, L.: Open access intrapartum ctg database. *BMC Pregnancy Childbirth* **14**, 16 (2014)