MACOC: a medoid-based ACO clustering algorithm

Héctor D. Menéndez †, Fernando E. B. Otero ‡, and David Camacho †

†Departamento de Ingeniería Informática, Universidad Autónoma de Madrid
Madrid, Spain
{hector.menendez,david.camacho}@uam.es
‡School of Computing, University of Kent, Chatham Maritime, United Kingdom
F.E.B.Otero@kent.ac.uk

Abstract. The application of ACO-based algorithms in data mining is growing over the last few years and several supervised and unsupervised learning algorithms have been developed using this bio-inspired approach. Most recent works concerning unsupervised learning have been focused on clustering, showing great potential of ACO-based techniques. This work presents an ACO-based clustering algorithm inspired by the ACO Clustering (ACOC) algorithm. The proposed approach restructures ACOC from a centroid-based technique to a medoid-based technique, where the properties of the search space are not necessarily known. Instead, it only relies on the information about the distances amongst data. The new algorithm, called MACOC, has been compared against well-known algorithms (K-means and Partition Around Medoids) and with ACOC. The experiments measure the accuracy of the algorithm for both synthetic datasets and real-world datasets extracted from the UCI Machine Learning Repository.

Keywords: Ant Colony Optimization, Clustering, Data Mining, Machine Learning, Medoid

1 Introduction

Ant Colony Optimization (ACO) algorithms have been widely used in several research fields. One of the most successful application field of ACO algorithms is data mining [14, 17, 3, 4, 18]. Data mining techniques are based on knowledge extraction or pattern identification inside an information source (dataset). They usually are focused on supervised or unsupervised learning techniques. Supervised techniques use the information of a class (target) attribute in order to identify predictive patterns in the data, while unsupervised techniques identify patterns that can group similar data points into categories. The advantage of applying ACO algorithm in these problems is that ACO algorithms perform a global search in the solution space, which in turn has the potential to find more accurate solutions, and they are less likely to get trap in a local minima.

The work presented in this paper is focused on the application of ACO in the unsupervised learning task of clustering, where the goal is to group (cluster)

similar data points in the same group and, at the same time, maximise the difference between different clusters. It has been inspired by a previous algorithm, called ACO-based Clustering algorithm (ACOC), proposed by Kao and Cheng [10]. ACOC is a centroid-based clustering algorithm, which tries to optimize the centroid (central point) position of each cluster. Following this idea, we focused the proposed algorithm on addressing the centroid-based approaches problems: they need to know the properties of the search space in order to determine the central point, and they are sensitive to noise effects. Inspired by other clustering algorithms [15, 16], we reformulated the original ACOC algorithm in a different way to create a medoid-based algorithm. Medoid-based clustering algorithms are usually more robust to noise effects, and do not need the properties of the search space to find a solution—they usually have the distance amongst the data instances, which can be obtained as a Gram matrix of a kernel or a distance measure, and they try to choose those data instances to define the best clusters. These selected instances are called medoids.

In order to check the performance of the proposed algorithm, we have compared it against the original ACOC algorithm using synthetic and real-world datasets and also included the well-known clustering algorithms PAM (Partition Around Medoids) [11] and K-means [13] in the comparisons.

The rest of the paper is structured as follows: Section 2 introduces the related work, Section 3 presents the new algorithm, Section 4 shows the experiments on synthetic and real-world datasets, and, finally, last section explains the conclusions and the future work.

2 Related Work

Ant Colony Optimization (ACO) has become a promising field for data mining problems. ACO algorithms combine the ants foraging behaviour to generate patterns that describe the data according to a supervised or unsupervised learning criteria—depending on the type of algorithm, classification or clustering, respectively.

On the one hand, classification [12] approaches are based on the class information of the data. These classes form part of the data instances and are considered by the algorithm in order to generate a general predictive model that describe (classify) the data. Different classification models have been designed in machine learning. The most common models are based on [12]: decision trees (C4.5), classification rules (RIPPER), artificial neural networks, random decision forest, support vector machines, naïve Bayes, k-nearest neighbours, amongst others. From the ACO point of view, there are several adaptations of these algorithms—e.g., Parpinelli et al. [19] present an ACO to create classification rules; Otero et al. introduce an ACO algorithm for decision tree induction [17]; Blum and Socha introduce a neural network ACO model [3] and Borrotti and Poli focused their work on the naïve Bayes model [4].

There are also approaches that combine ACO with classical classification algorithms in order to improve their results. Some of these techniques, for example,

optimize the parameter selection for the classifier (e.g., for SVMs [22]), other are focused on the feature selection process for the data preprocessing phase [6]. On the other hand, clustering [12] is based on a blind search within the data. Clustering techniques try to join similar data points into groups (clusters) according to a cost or objective function, which is usually minimized or maximized, making this clusters different from each other at the same time. There is a large number of clustering approaches, similar to classification, depending of the goal the algorithm should achieve. These techniques are usually divided in three types of clustering [8]: partitional (each instances belongs to a single cluster), overlapping (each instance belongs to one or more clusters) and hierarchical (partitional solutions are nested to generate a tree of clusters). Depending on the clustering algorithm, there are several models focus on the solutions. These models are also divided in parametric [13] or non-parametric model [20], where the former has an statistical estimator that is adapted to the data while the latter separates the data using different topologies or techniques.

The most classical algorithms are K-means [13] and EM [5]. Both K-means and EM are parametrical partitional clustering algorithms, which usually try to optimize an estimator parameters. From a similar perspective, medoid-based algorithms try to find the solution within the data [11]. Medoid-based algorithms, as was mentioned before, do not need the features of the search space in order to find a solution—they can deal with the information extracted from the data distances. This special property makes medoid-based algorithms a good choice for problems where the search space is not well defined, such as time series clustering. There are also bio-inspired algorithms that deal with the clustering problem, several of them focused on genetic algorithms. Hruschka et al. [8] presents a survey of clustering algorithms from different genetic approaches. From other bio-inspired perspectives, ACO algorithms have also produced promising results. Kao and Cheng [10] introduced a centroid-based ACO clustering algorithm; França et al. [7] introduce a bi-clustering algorithm; and Ashok and Messinger focused their work on graph-based clustering [1]; several other approaches are discussed in [9].

3 Medoid-based ACO Clustering Algorithm (MACOC)

This section presents the Medoid-based ACO Clustering Algorithm (MACOC). The MACOC algorithm is similar to Partition Around Medoids (PAM) algorithm, where the goal of the algorithm is to choose the best M medoids (data instances) based only on distance information. This kind of algorithms usually use a dissimilarity/similarity matrix that measures the distances between the data points. The medoid-based approach is a generalization of the centroid-based approach, but in the medoid case, the properties of the search space are not required—only the distances between the data points.

As an ACO algorithm, MACOC algorithm is based on ACOC algorithm [10]. They have a similar search graph, where the ants try to define the optimal cluster assignment for each of the instances (data points). This graph is based on instances and clusters (Fig. 1). It has an associated NxM matrix, where N

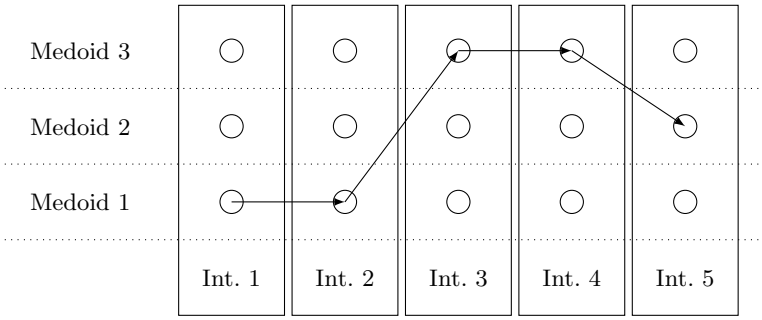

Fig. 1. Representation of the ant travelling around the search graph. Ant visit each instance in order to assign them to a medoid based on the heuristic information and pheromone levels.

is the number of instances and M is the number of clusters (medoids). While in the case of ACOC the construction graph is full-connected, MACOC uses a graph divided in levels, where each instance defines a level (Fig. 1). The nodes are visited following the sequence of instance, therefore the graph is not full-connected, reducing significantly the memory usage and the complexity of the solution space. It should be noted that this does not incorporate any bias in the search, since the order of the instances is not relevant.

The algorithm is based on several ants looking for the best path in the construction graph. Each ant (k) has the following features:

- set of chosen medoids M^k (which are randomly selected).
- weight matrix W^k (based on the distance between the instances and the ant's medoids). This is similar to ACOC W^k matrix.

The ant has two possible search strategies, exploration and exploitation, similar to the ACOC algorithm. MACOC chooses the strategy for the cluster assignation j according to the ACOC equation:

$$j = \begin{cases} argmax_{u \in N_i}\{[\tau(i,j)][\eta(i,j)]^{\beta}\} & \text{, if } q \leq q_0 \\ S & \text{, otherwise} \end{cases} , \tag{1}$$

where N_i is the set of nodes associated to instance i (see Fig. 1), j is the chosen cluster, $\tau(i,j)$ is the pheromone value between i and j, q_0 is the user-defined exploitation probability, q is a random number for strategy selection, $\eta(i,j)$ is the heuristic information between i and j, and S is the ACO-based search strategy. The heuristic information between an instance i and a candidate medoid j is defined by the formula:

$$\eta(i,j) = \frac{1}{d(i,j)} , \tag{2}$$

and the ACO-based exploration strategy S—in the same way than ACOC, is defined by:

$$S = P(i,j) = \frac{[\tau(i,j)] \cdot [\eta(i,j)]^{\beta}}{\sum_{l=1}^{m} [\tau(i,l)] \cdot [\eta(i,l)]^{\beta}} \ . \tag{3}$$

One of the main differences between ACOC and MACOC is that MACOC keeps more information about the ants movements in the pheromone matrix. In the case of ACOC, the pheromone matrix is a relationship between the instance and the centroid-label (i.e., the index of the cluster), which is not the centroid itself. In the case of MACOC, the pheromone matrix is a relationship between the instance and the medoid (another data instance), which means that if/when the medoid-label changes as a result of the random selection process, the previous pheromone value is still available. In other words, in the ACOC algorithm, if the centroid value that was previously used as the centroid c_1 is used as the centroid c_2, the previous pheromone values are lost, since they are associated with the label (position) c_1; in the MACOC algorithm, if the medoid instance that was previously used as medoid m_1 is used as the medoid m_2, the previous pheromone values are still used, since the pheromone is associated with the data instance and not with the medoid label.

The MACOC algorithm is structured in the same way than ACOC and can be described as follows:

1. Initialize the pheromone matrix (τ_0), which is global for all ants
2. Initialize ants: choose n random medoids for M^k (n is the number of clusters) and set the matrix W^k to 0. For each ant, until all instances have been visited:
   (a) Select the next data object i
   (b) Select a cluster j: i) Choose a strategy; ii) Calculate neighbouring nodes probability and iii) Visit the node
   (c) Update W^k
3. Choose the best solution:
   (a) Calculate the objective function for each ant:

$$J^k = \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij}^k \cdot d(x_i, m_j^k) \ , \tag{4}$$

   where $w_{ij}^k \in W^k$ and d is a distance function
   (b) Rank the ants solutions
   (c) Choose the best ant (iteration-best solution)
   (d) Compare it with the best-so-far solution and update this value with the maximum between them
4. Update the pheromone trails (global updating rule): only the r best ants are able to add pheromones:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{h=1}^{r} w_{ij}^h \cdot \Delta\tau_{ij}^h \ , \tag{5}$$

   where ρ is the pheromone evaporation rate, $(0 < \rho < 1)$, $w_{ij}^h \in W^h$, t the iteration number, r is the number of elitism ants and $\Delta\tau_{ij}^h = 1/J^k$ is the quality of the solution created by ant h

5. Check termination condition:
   (a) If the number of iterations is greater than the total iterations: re-centralise the instances assigning each data point to its closest medoid and finish
   (b) Otherwise, go to step 2

4 Experiments

This section presents the experiments which have been carried out to measure the quality of the proposed MACOC algorithm. The comparisons have been carried out against K-means, PAM and ACOC algorithms.

4.1 Datasets Description

For the synthetic experiments we have created the following datasets:

- *Synthetic Data 1*: This dataset is formed by 9 two-dimensional gaussian models and in this case, there are 3 gaussians which are closer than the rest.
- *Synthetic Data 2*: This second dataset is also formed by 9 two-dimensional gaussian models, however, in this case, there are noisy data in the background.

For the real-world experiments, we have chosen four datasets extracted from UCI Machine Learning Repository [2], which are commonly used as benchmark for classification and clustering:

- *Iris*: Contains 50 instances distributed over 3 classes, with 4 attributes each.
- *Wine*: Contains 178 instances distributed over 3 classes, with 13 attributes each.
- *Vertebral Column* (Ver. Col.): Contains 310 instances distributed over 3 classes, with 6 attributes each.
- *Breast Tissue* (Bre. Tis.): Contains 106 instances distributes over 6 classes, with 10 attributes each.

4.2 Experimental Setup and Evaluation Methods

We selected three algorithms to measure the MACOC quality, namely K-means, PAM and ACOC.

K-means [13] is an iterative algorithm based on centroids, which are randomly selected at the beginning. The goal of the algorithm is to find the best centroid positions. It is executed in two steps: in the first step, it assigns the data to the closest centroid (cluster); and in the second, it calculates the new position of the centroid as a centroid of the data which has been assigned to it.

PAM [11] is similar to K-means, but it used medoids instead of centroids. PAM can works with a dissimilarity/similarity matrix, which is used to calculate the cost of each medoid belonging to a cluster.

ACOC [10] is the algorithm which our algorithm is inspired. It work with centroids and ants. The main different, apart of the algorithm centroid nature, is that ACOC uses a pheromone matrix from the data instances to the centroid-labels, while our algorithm use a pheromone matrix between all the data to remember the previous medoid assignation. The parameters of ACOC and MACOC algorithms have been set in a similar way to the original work [10]: the number on ants is 10, the number of elitism is 1, the exploration probability is 0.0001, the initial pheromone values follow an uniform distribution $[0.7, 0.8]$, $\beta = 2.0$, $\rho = 0.1$, and the maximum number of iterations is 1000. The only difference is that the MACOC initial pheromone values have been set as $\frac{1}{n}$ (where n is the number of clusters).

All the experiments have been carried out 50 times—except for K-means, which was carried out 100 times since this algorithm tends to converge to local minima—using the Euclidean distance as the metric, defined by:

$$d(x_i, x_j) = ||x_i - x_j|| = \sqrt{\sum_q (x_i^q - x_j^q)^2} \ , \tag{6}$$

where x_i, x_j represent two data instances and q represents each attribute of the data instance. Additionally, all algorithms need the number of cluster as an initial parameter.

The evaluation of the experiments has been focused on two different ideas: the synthetic dataset has been evaluated according to the cluster discrimination and the performance of the algorithm to discriminate the original clusters in the noisy case; the real-world datasets have been evaluated using the accuracy.

4.3 Synthetic Experiments

The first synthetic dataset is generally easy for all the algorithms (Fig. 2 and 1). The discrimination of the clusters is clearer in this case, resulting in a clear separation of the clusters. The only algorithm that has several problems in identifying the clusters is K-means (see Fig. 2 and Table 1)—probably a result of an early convergence to local minimal solution. PAM provides a good solution of the cluster discrimination and also provides a stable solution (its standard deviation is 0). It means that the algorithm is able to find the medoids with no problems. In the case of ACOC, the solution discriminates the cluster kernels, however, the boundaries are not well-defined (see Fig. 2). MACOC obtains good results for both cluster identification and boundary definition, and also accuracy results (see Table 1).

The second dataset introduces noise and the noise significantly modifies the behaviour of the algorithms (Fig. 3 and Table 1). K-means is not able to identify the cluster kernels and it joins several clusters together, generating a cluster with noisy data. ACOC is also able to find the kernels and discriminate them, however, the boundaries are not well-defined and several instances overlap with other clusters. Finally, PAM and MACOC achieve similar results. In the case of PAM, there are some boundary problems in the central clusters while in the

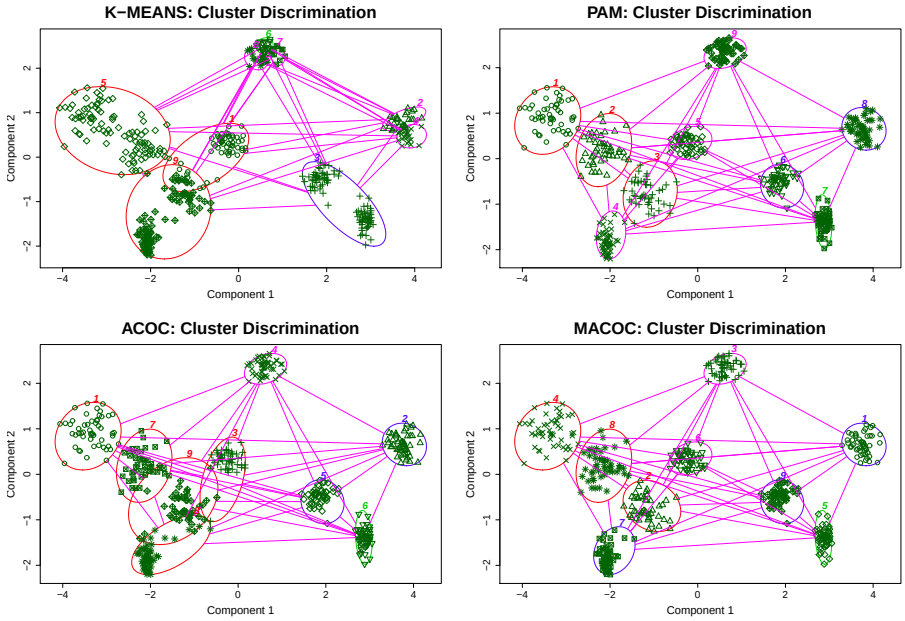

Fig. 2. Results for the synthetic 9 gaussian distribution.

case of MACOC the boundaries are clearer, except for one instance (see Fig. 3, at the right of the MACOC image).

These results suggest the following conclusions regarding the comparison with ACOC: while ACOC has boundary problems, which are increased when there is noisy information, MACOC obtains good results for cluster boundary definition and it is more robust to the presence of noise.

4.4 Real-World Experiments

Table 2 shows the results of the algorithms applied to real-world datasets extracted from UCI Machine Learning repository [2].

In the Iris case, K-means and PAM obtains similar results according to the median. K-means obtains the worst minimum accuracy results (58%) and it is the less robust algorithm (its standard deviation is 0.1313). PAM is the most robust algorithm in this case (0 standard deviation), while ACOC and MACOC obtain similar robustness results. The highest minimum value is achieved by both ACOC and PAM. The highest maximum, mean and median values are achieved by MACOC (95.33%, 90.67% and 90.65%, respectively). While MACOC shows better results than ACOC, these results can not be considered different because the null hypothesis can not be refused according to Wilcoxon Test [21].

The application of the algorithms on Wine dataset has shown that K-means also obtains the worst results according to the accuracy and robustness. PAM

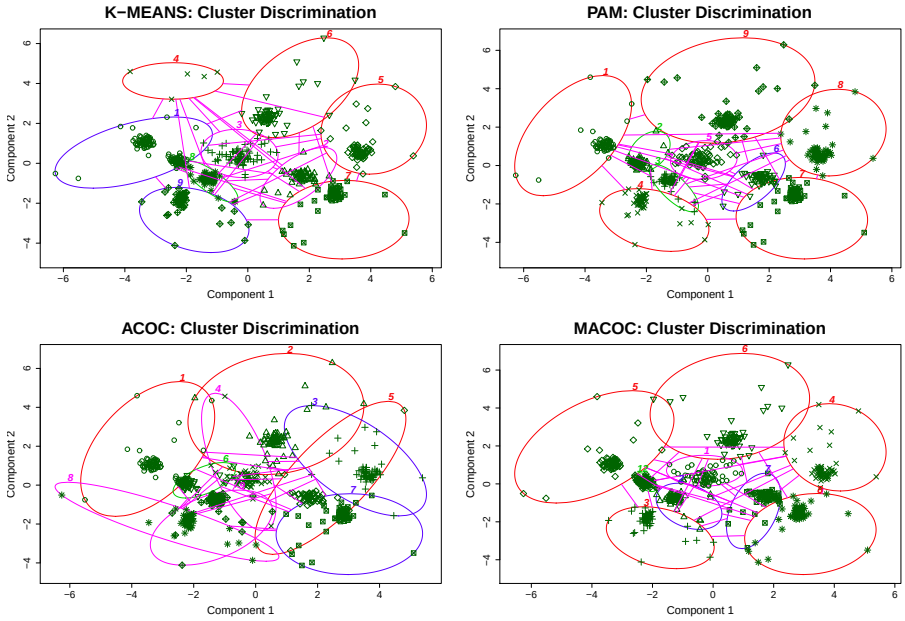

Fig. 3. Results for synthetic 9 gaussian distribution.

obtains the highest minimum value and it is again the most robust algorithm. According to the maximum, mean and median values, MACOC achieves the best results (72.47%, 71.47% and 71.91%, respectively). In this case the null hypothesis is refused with a good significance level, therefore, MACOC results are statistically significantly better than ACOC.

Vertebral column (Ver. Col.) dataset shows different results than the rest of the datasets. In this case, the best algorithm is K-means, which achieves the maximum value for all the metrics. MACOC is the second algorithm according to median, mean and max (52.58%, 53.26% and 65.48%). Again, the Wilcoxon test shows that the null hypothesis can be refused with a high significance level (3e-05), therefore, MACOC results are statistically significantly better than ACOC.

Finally, Breast Tissue dataset shows that MACOC achieves the best results according to mean, median and max (35.55%, 34.91% and 40.57%). In this case, ACOC and PAM achieves similar results, specially according to the median (33.96 %). The null hypothesis can be refused with a significance level of 0.05 (in this case, Wilcoxon test is 0.023), therefore, MACOC results are statistically significantly better than ACOC.

These results show that MACOC improves the performance over ACOC, given that that the solutions obtained by MACOC and ACOC are usually statistically different according to Wilcoxon test in favour of MACOC. Overall, this results are promising regarding the use of medoids instead of centroid, since this is the main difference between MACOC and ACOC. In a similar way that

Table 1. Results of the application of the algorithms to the synthetic datasets. The p-values for the Wilcoxon test applied to ACOC and MACOC results are: Synthetic 1 (2.394e-13) and Synthetic 2 (7.734e-10)—statistical significant improvements are indicated by a ▲ symbol.

MACOC	Min	Max	Median	Mean		SD
Synthetic 1	99.11%	**100.0**%	**99.78**%	99.75%	▲	± 0.0028
Synthetic 2	96.73%	**100.0**%	**98.18**%	**98.75**%	▲	± 0.0121
ACOC	Min	Max	Median	Mean		SD
Synthetic 1	92.67%	**100.0**%	98.89%	98.57%		± 0.0128
Synthetic 2	82.91%	92.27%	95.27%	94.40%		± 0.0314
K-means	Min	Max	Median	Mean		SD
Synthetic 1	56.67%	99.78%	83.42%	79.47%		± 0.1152
Synthetic 2	65.72%	98.00%	76.73%	79.92%		± 0.0792
PAM	Min	Max	Median	Mean		SD
Synthetic 1	**99.78**%	99.78%	**99.78**%	**99.78**%		± 0.0000
Synthetic 2	**98.00**%	98.00%	98.00%	98.00%		± 0.0000

median is more stable than mean, medoids are usually more stable to outliers in the data than centroids. Future studies will be focused on how the algorithm responds to worse conditions, such as large data, more outliers and extremely noisy data.

5 Conclusions and Future Work

This work presented a new ACO-based clustering algorithm, called MACOC, which is focused on a medoid-based approach. MACOC is an adaptation from a previously proposed centroid-based ACOC algorithm to a medoid-based approach. From the ACO perspective, the new algorithm has also improved the use of the pheromone, extending the pheromone matrix to keep the information from different candidate medoid instances across iterations.

The application of the algorithm to synthetic and real-world datasets has shown that MACOC is more robust to noisy information and it defines better cluster boundaries than ACOC. It also showed that MACOC has good general results compared with well-known clustering algorithms.

The future work will be focused on improving the results of MACOC by incorporating a medoid recalculation process. It would be interesting to explore the addition of the medoid selection to the construction graph, allowing ants to share information of the best performing medoid instances.

Table 2. Results of the application of the algorithms to the different datasets extracted from the UCI database. The p-values for the Wilcoxon test applied to ACOC and MACOC solutions are: Iris (0.4439), Wine (4.117e-07), Ver. Col. (3e-05), Bre. Tis. (0.02349)—statistical significant improvements are indicated by a ▲ symbol.

MACOC	Min	Max	Median	Mean		SD
Iris	87.33%	**95.33**%	**90.65**%	**90.67**%		± 0.0187
Wine	69.10%	**72.47**%	**71.91**%	**71.47**%	▲	± 0.0075
Ver. Col.	46.13%	**65.48**%	52.58%	53.26%	▲	± 0.0488
Bre. Tis.	28.30%	**40.57**%	**34.91**%	**35.55**%	▲	± 0.0328
ACOC	Min	Max	Median	Mean		SD
Iris	**89.33**%	93.33%	90.00%	90.13%		± 0.0080
Wine	70.22%	71.35%	70.79%	70.78%		± 0.0026
Ver. Col.	47.74%	54.19%	49.35%	49.43%		± 0.0095
Bre. Tis.	31.13%	**40.57**%	33.96%	34.47%		± 0.0229
K-means	Min	Max	Median	Mean		SD
Iris	58.00%	89.33%	89.33%	82.46%		± 0.1313
Wine	56.74%	69.33%	70.22%	67.26%		± 0.0632
Ver. Col.	**56.13**%	**65.48**%	**56.13**%	**57.81**%		± 0.0339
Bre. Tis.	33.02%	33.96%	33.21%	33.04%		± 0.0025
PAM	Min	Max	Median	Mean		SD
Iris	**89.33**%	89.33%	89.33%	89.33%		± 0.0000
Wine	**70.79**%	70.79%	70.79%	70.79%		± 0.0000
Ver. Col.	48.71%	48.71%	48.71%	48.71%		± 0.0000
Bre. Tis.	**33.96**%	33.96%	33.96%	33.96%		± 0.0000

Acknowledgments. This work has been partly supported by: Spanish Ministry of Science and Education under project TIN2010-19872 and Savier – an Airbus Defense & Space project (FUAM-076914 and FUAM-076915).

References

1. Ashok, L., Messinger, D.W.: A spectral image clustering algorithm based on ant colony optimization. vol. 8390, pp. 83901P–83901P–10 (2012), `http://dx.doi.org/10.1117/12.919082`
2. Bache, K., Lichman, M.: UCI machine learning repository (2013), `http://archive.ics.uci.edu/ml`
3. Blum, C., Socha, K.: Training feed-forward neural networks with ant colony optimization: An application to pattern classification. In: Proceedings of HIS'05. pp. 233–238. IEEE Computer Society, Washington, DC, USA (2005), `http://dx.doi.org/10.1109/ICHIS.2005.104`
4. Borrotti, M., Poli, I.: Naïve bayes ant colony optimization for experimental design. In: Kruse, R., Berthold, M.R., Moewes, C., Gil, M.A., Grzegorzewski, P., Hryniewicz, O. (eds.) Synergies of Soft Computing and Statistics for Intelligent Data Analysis, Advances in Intelligent Systems and Computing, vol. 190,

pp. 489–497. Springer Berlin Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-33042-1_52
5. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977), http://web.mit.edu/6.435/www/Dempster77.pdf
6. Ding, S.: Feature selection based f-score and aco algorithm in support vector machine. In: Knowledge Acquisition and Modeling, 2009. KAM '09. Second International Symposium on. vol. 1, pp. 19–23 (Nov 2009)
7. França, F., Coelho, G., Zuben, F.: bicACO: An Ant Colony Inspired Biclustering Algorithm. In: Ant Colony Optimization and Swarm Intelligence, LNCS, vol. 5217, pp. 401–402. Springer Berlin Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-87527-7_45
8. Hruschka, E., Campello, R., Freitas, A., de Carvalho, A.: A survey of evolutionary algorithms for clustering. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 39(2), 133–155 (march 2009)
9. Jafar, O.M., Sivakumar, R.: Ant-based clustering algorithms: A brief survey. International Journal of Computer Theory and Engineering 2, 787–796 (2010)
10. Kao, Y., Cheng, K.: An aco-based clustering algorithm. In: Ant Colony Optimization and Swarm Intelligence, Lecture Notes in Computer Science, vol. 4150, pp. 340–347. Springer Berlin Heidelberg (2006), http://dx.doi.org/10.1007/11839088_31
11. Kaufman, L., Rousseeuw, P.: Clustering by Means of Medoids. Reports of the Faculty of Mathematics and Informatics (1987), http://books.google.co.uk/books?id=HK-4GwAACAAJ
12. Larose, D.T.: Discovering Knowledge in Data. John Wiley & Sons (2005)
13. Macqueen, J.B.: Some methods of classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. pp. 281–297 (1967)
14. Martens, D., Baesens, B., Fawcett, T.: Editorial survey: swarm intelligence for data mining. Machine Learning 82(1), 1–42 (2011)
15. Menéndez, H.D., Barrero, D.F., Camacho, D.: A genetic graph-based approach for partitional clustering. Int. J. Neural Syst. 24(3) (2014)
16. Orgaz, G.B., Menéndez, H.D., Camacho, D.: Adaptive k-means algorithm for overlapped graph clustering. Int. J. Neural Syst. 22(5) (2012)
17. Otero, F., Freitas, A., Johnson, C.: Inducing decision trees with an ant colony optimization algorithm. Applied Soft Computing 12(11), 3615–3626 (2012)
18. Otero, F., Freitas, A., Johnson, C.: A New Sequential Covering Strategy for Inducing Classification Rules With Ant Colony Algorithms. IEEE Transactions on Evolutionary Computation 17(1), 64–76 (2013)
19. Parpinelli, R., Lopes, H., Freitas, A.: Data mining with an ant colony optimization algorithm. IEEE Trans. on Evolutionary Computation 6(4), 321–332 (2002)
20. Schaeffer, S.: Graph clustering. Computer Science Review 1(1), 27–64 (2007)
21. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics 1(6), 80–83 (1945)
22. Zhang, X., Chen, X., He, Z.: An aco-based algorithm for parameter optimization of support vector machines. Expert Syst. Appl. 37(9), 6618–6628 (Sep 2010), http://dx.doi.org/10.1016/j.eswa.2010.03.067