

# Ant Clustering Algorithm

Urszula Boryczka

Institute of Computer Science, University of Silesia, Sosnowiec, Poland

## Abstract

Among the many bio-inspired techniques, ant clustering algorithms have received special attention, especially because they still require much investigation to improve performance, stability and other key features that would make such algorithms mature tools for data mining. Clustering with swarm-based algorithms is emerging as an alternative to more conventional clustering methods, such as e.g. k-means etc. This approach mimics the clustering behavior observed in real ant colonies.

As a case study, this paper focus on the behavior of clustering procedures in this new approach. The proposed algorithm is evaluated in a number of well-known benchmark data sets. Empirical results clearly show that ant clustering algorithm performs well when compared to another techniques.

**Keywords:** data mining, cluster analysis, ant clustering algorithm

## 1 Introduction

Clustering is a form of classification imposed over a finite set of objects. The goal of clustering is to group sets of objects into classes such that similar objects are placed in the same cluster while dissimilar objects are in separate clusters. Clustering (or classification) is a common form of data mining and has been applied in many fields including data compression, texture segmentation, vector quantization, computer vision and various business applications. Clustering algorithms can be classified into partitioning and hierarchical algorithms. Partitioning algorithms create a partitioning of objects into a set of clusters. Hierarchical algorithms construct a hierarchical decomposition of objects. The hierarchical decomposition is represented by a tree strategy that separates the objects into small subsets until each consists only of sufficiently similar objects. There exist a large number of clustering algorithms in the literature including k-means in MacQueen (1967), k-medoids in Kaufman and Russeeuw (1990), *CACTUS* in Ganti *et al.* (1999), *CURE* in Guha *et al.* (1998), *CHAMELEON* in Karypis *et al.* (1999) and *DBSCAN* in Ester *et al.* (1996). No single algorithms is suitable for all types of objects, nor all algorithms appropriate for all problems, however, the k-medoids algorithms have been shown in Kaufman and Russeeuw (1990) to be robust to outliers, compared with centroid-based clustering. The drawback of the k-medoids algorithms is the time complexity of determining the medoids. In this paper, a novel ant-based clustering algorithm is proposed to improve the performance of many k-medoids-based

algorithms. A new version of *ACA* algorithm is inspired from the behavior of real ants. The paper is organized as follows: the section 2 gives a detailed description of the biological inspirations and first experiments. The section 3 presents this algorithm. The section 4 presents the experiments that have been conducted to set the parameters of *ACA* regardless on the data sets. The last section concludes and discusses future evolutions of *ACA*.

## 2 Biological inspirations and algorithms

Clustering and sorting behavior of ants has stimulated researches to design new algorithms for data analysis and partitioning. Several species of ants cluster corpses to form a “cemetery”, or sort their larvae into several piles. This behavior is still not fully understood, but a simple model, in which ants move randomly in space and pick up and deposit items on the basis of local information, may account for some of the characteristic features of clustering and sorting in ants Bonabeau *et al.* (1999).

In several species of ants, workers have been reported to form piles of corpses — cemeteries — to clean the nests. Chretien (1996) has performed experiments with the ant *Lasius Niger* to study the organization of cemeteries. Other experiments on the ant *Phaidole pallidula* are also reported in Deneubourg *et al.* (1991). Brood sorting is observed by Franks and Sendova-Franks (1992) in the ant *Leptothorax unifasciatus*. Workers of this species gather the larvae according to their size. Franks and Sendova-Franks (1992) have intensively analyzed the distribution of brood within the brood cluster.

Deneubourg *et al.* (1991) has proposed two closely related models to account for the two above-mentioned phenomena of corpse clustering and larval sorting in ants. General idea is that isolated items should be picked up and dropped at some other location where more items of that type are present. Let us assume that there is only one type of item in the environment. The probability  $p_p$  for a randomly moving, unladen agent to pick up an item is given by

$$p_p = \left( \frac{k_1}{k_1 + f} \right)^2$$

where:

- $f$  is the perceived fraction of items in the neighborhood of the agent,
- $k_1$  — is a threshold item.

The probability  $p_d$  for a randomly moving loaded agent to deposit an item is given by:

$$p_d = \left( \frac{f}{k_2 + f} \right)^2$$

where:

- $k_2$  is another threshold constant.

Franks and Sendova-Franks (1992) have assumed that  $f$  is computed through a short-term memory that each agent possesses, it is simply the number  $N$  of items encountered during these last  $T$  time units, divided by the largest possible number of items that can be encountered during this time.

Gutowitz (1993) has suggested the use of spatial entropy to track the dynamics of clustering. The spatial entropy  $E_s$  at scale  $s$  is defined by:

$$E_s = \sum_{I \in S} P_I \log P_I$$

where  $P_I$  is the fraction of all objects on the lattice that are found in s-patch  $I$ .

Oprisan *et al.* (1996) proposed a variant of Deneubourg basic model (hereafter called BM), in which the influence of previously encountered objects is distributed by a time factor.

Bonabeau (1997) also explored the influence of various weighting functions, especially those with short-term activation and long-term inhibition.

Lumer and Faieta (1994) have generalized Deneubourg *et al.*'s BM to apply it to exploratory data analysis. The idea is to define a distance or dissimilarity  $d$  between objects in the space of object attributes:

- if two objects are identical then  $d(o_i, o_j) = 0$ ,
- when two objects are not identical then  $d(o_i, o_j) = 1$ .

The algorithm introduced by Lumer and Faieta (hereafter LF) consists of projecting the space of attributes onto some lower dimensional space, typically of dimension  $z = 2$ . Let us assume that an ant is located at site  $r$  at time  $t$ , and finds an object  $o_i$  at that site. The "local density"  $f(o_i)$  with respect to object  $o_i$  is given by

$$f(o_i) = \begin{cases} \frac{1}{s^2} \sum_{o_j \in Neigh(s \times s)(r)} [1 - \frac{d(o_i, o_j)}{\alpha}], & \text{when } f > 0 \\ 0, & \text{otherwise} \end{cases}$$

where:

- $f(o_i)$  is a measure of the average similarity of object  $o_i$  with the other objects  $o_j$  present in the neighborhood of  $o_i$ ,
- $\alpha$  is a factor that defines the scale for dissimilarity: it is important for it determines when two items should or should not be located next to each other.

Lumer and Faieta (1994) define picking up and dropping probabilities as follows:

$$p_p(o_i) = \left( \frac{k_1}{k_1 + f(o_i)} \right)^2$$

$$p_d(o_i) = \begin{cases} 2f(o_i) & \text{when } f(o_i) < k_2 \\ 1, & \text{when } f(o_i) \geq k_2 \end{cases} \quad (1)$$

where  $k_1, k_2$  are two constants that play a role similar to  $k_1$  and  $k_2$  in the BM.

### 3 Ant Clustering Algorithm — ACA

The ant clustering algorithms are mainly based on versions proposed by Deneubourg, Lumer and Faieta. A number of slight modifications have been introduced that improve the quality of the clustering and, in particular, the spatial separation between clusters on the grid. Recently Handl and Meyer (2002) extended Lumer and Faieta's algorithm and proposed an application to the classification of Web documents. The model proposed by Handl and Meyer has inspired us to use this idea to classical cluster analysis. The basic idea is to pick up or drop a data item on the grid.

We have employed a modified version of the "short-term memory" introduced by Lumer and Faieta (1994). Each ant has a permission to exploit its memory according to these rules: if an ant situated at grid cell  $p$ , and carrying a data item  $i$ , it uses its memory to proceed to all remembered positions, one after the other. Each of them is evaluated using the neighbourhood function  $f^*(i)$  for finding a dropping site for the currently carried data item  $i$ .

For picking and dropping decisions the following threshold formulae are used:

$$p_{pick}^*(i) = \begin{cases} 1, & \text{if } f^*(i) > 1 \\ \frac{1}{f^*(i)^2}, & \text{else} \end{cases}$$

$$p_{drop}^*(i) = \begin{cases} 1, & \text{if } f^*(i) \geq 1 \\ \frac{1}{f^*(i)^4}, & \text{else,} \end{cases}$$

where  $f^*(i)$  is a modified version of Lumer and Faieta's neighbourhood function:

$$f^*(i) = \begin{cases} \frac{1}{\sigma^2} \sum_j [1 - \frac{d(i,j)}{\alpha}], & \text{if } f^* > 0 \\ & \text{and } (1 - \frac{d(i,j)}{\alpha}) > 0 \\ 0, & \text{otherwise} \end{cases}$$

- $\frac{1}{\sigma^2}$  — a neighborhood scaling parameter,
- $\alpha$  — a parameter scaling the dissimilarities within the neighbourhood function  $f^*(i)$ ,
- $d(i, j)$  — a dissimilarity function.

Ant-based clustering algorithm requires a number of different parameters to be set, which have been experimentally observed. Parameters of this algorithm we can divide into two groups:

1. To be independent of the data,
2. To be set as a function of the size of the data set.

The first group includes:

- the number of agents, which be set to be 10,
- the size of the agents' short-term memory, which we equally set to 10,
- the initial clustering phase (from  $t_{start}$  to  $t_{end}$ :  $t_{start} = 0.45 \cdot N$ ,  $t_{end} = 0.55 \cdot N$ , where  $N$  denote the number of iterations,

- we replace the scaling parameter  $\frac{1}{\sigma^2}$  by  $\frac{1}{N_{occ}}$  after an initial clustering phase, where  $N_{occ}$  is the actual observed number of occupied grid cells within the local neighbourhood.

The employed distance function is the Euclidean measure for the initial testing and the Cosine and Gower measures for the real data analysis.

Several parameters should be selected in dependence of the size of the data set tackled. Given a set of  $N_{max}$  items, the grid should offer a sufficient amount of “free” space to permit the quick dropping of data items. This can be achieved by:

- using a square grid with resolution of  $\sqrt{10 \cdot N_{max}} \times \sqrt{10 \cdot N_{max}}$ ,
- the step should permit sampling of each possible grid position within one move, which is obtained by setting it to stepsize:  $\sqrt{20 \cdot N_{max}}$ ,
- the number of iterations:  $\sqrt{2000 \cdot N_{max}}$ , with a minimal number of 1000000.

During the sorting process,  $\alpha$  determines the percentage of data items on the grid that are classified as similar, such that: a too small choice of  $\alpha$  prevents the formation of clusters on the grid; on the other hand, a too large choice of  $\alpha$  results in the fusion of individual clusters, and in the limit, all data items would be gathered within one cluster.

The scheme for  $\alpha$ -adaptation used in this application is a part of a self-adaptation of agents activity. A heterogenous population of ants is used — with its own parameter  $\alpha$ . An agent considers an adaptation of its own parameter after it has performed  $N_{active}$  moves. During this time, it keeps track of the failed dropping operations  $N_{fail}$ . The rate of failure is determined as  $r_{fail} = \frac{N_{fail}}{N_{active}}$  where  $N_{active}$  is fixed to 100. The agent’s parameter  $\alpha$  is then updating using the rule:

$$\alpha = \begin{cases} \alpha + 0.01, & \text{if } r_{fail} > 0.99 \\ \alpha - 0.01, & \text{if } r_{fail} \leq 0.99. \end{cases}$$

High-level description of the ant clustering algorithm (ACA) is presented below:

---

Algorithm 1: ACA algorithm

---

```

0 /*Initialization Phase*/
1 Randomly scatter  $o_i$  object on the grid file
2 for each agent  $a_j$  do
3   random_select_object ( $o_i$ )
4   pick_up_object  $o_i$ 
5   place_agent  $a_j$  at randomly selected empty grid location
6 end for
7 {*Main loop*}
8   for  $t = 1$  to  $t_{max}$  do
9     random_select_agent ( $a_j$ )
10    move_agent  $a_j$  to new location
11     $i = \text{carried\_object}(\text{agent}a_j)$ 
12    Compute  $f^*(o_i)$  and  $p_{drop}^*(o_i)$ 
13    if drop = True then
14      while pick = False do

```

```
15     i = random_select_object o
16     Compute  $f^*(o_i)$  and  $p_{pick}^*(o_i)$ 
17     Pick_up_object  $o_i$ 
18     end while
19   end if
20 end for
21 end
```

---

## 4 Experimental results

The performance of a clustering algorithm can be judged with respect to its relative performance when compared to other algorithms. we therefore at the beginning choose the k-means algorithm. In our experiments, we run *k-means* algorithm using the correct cluster number  $k$ .

We have applied *ACA* to real world databases from the Machine Learning repository which are often used as benchmark. The data set is useful to show experimentally the efficiency of *ACA* on data with known properties and difficulty. The real data collections used were the Iris data, the Wine Recognition, Ionosphere data and Pima data. Each dataset is permuted and randomly distributed in the sites. Different evaluation functions proposed by Handl *et al.* (2003) are adapted for comparing the clustering results obtained from applying the two clustering algorithms on the test sets. The F-measure Rijsbergen (1979), Dunn Index Halkidi *et al.* (2000) and Rand Index Rijsbergen (1979) are the three measures and their respective definitions also mentioned in (Handl *et al.* (2003)) and each should be maximized. We have also analyzed the Inner Cluster variance — the sum of squared deviations between all data items of their associated cluster centre Handl *et al.* (2003). It is to be minimized.

All runs have been performed for three different dissimilarity measures: Euclidean, Cosine and Gower measures. All presented results have been averaged over 10 runs. Ants (10 agents) were simulated during 1000000 iterations when clustering objects.

The results are mentioned in table 1, 2, 3 and 4. The tables show mean and standard deviations (in brackets) for 1000 000 runs, averaged over 10 runs. In experimental study we show the results reported in details in Skinderowicz (2007). This big number of iterations is a common characteristic for different ant-based clustering algorithms.

The results of ant clustering algorithm are very close to those of k-means on the analysed data sets. The reader should keep in mind that, different from its competitor, ant clustering algorithm has not been provided with the correct number of clusters. We also observed the sensitivity to unequally-sized clusters in analyzed data sets. We show the algorithms' performance on these data sets as reflected by F-measure. While the robust performance of the algorithms across a wide range of data sets has been demonstrated in these tables, our analysis in this report has focused in studying the scheme of adapting the  $\alpha$  values that pose problems to ant clustering algorithms. Importantly, it must be noted that

the cluster method is very sensitive to the choice of  $\alpha$  and correlations over a specific thresholds are only achieved with the proper choice of  $\alpha$ . From some of the results, the *ACA* demonstrated to be incapable of correctly clustering the data in most simulations. The proposed algorithm, however, was capable of appropriately clustering the data in all runs (with strong correlations), but with varying numbers of clusters being found each time the algorithm was run. Despite the sufficient results presented here, there are still several avenues for investigation that deserve to be pursued. For instance, because of too many clusters obtained by *ACA*, a hierarchical analysis of the data sets can be proposed by systematically varying some of the user-defined parameters: the use of set of objects (clusters) instead of a one object on a grid position scheme used here can be performed for an improvement.

## 5 Conclusions

We have presented in this paper a new ant clustering algorithm called *ACA* for data clustering in a knowledge discovery context. *ACA* introduces new ideas and modifications in Lumer and Faieta's algorithm in order to improve the convergence. The main features of this algorithm are the following ones. *ACA* deals with numerical databases. It does not require to establish the number of clusters or any information about the feature of the clusters.

The ant clustering algorithm has a number of features that make it an interesting candidate for augmentation in the context of applications. Firstly, because of its linear scaling behavior it is attractive for use in large data sets, e.g. in information retrieval systems. Secondly — this algorithm deals with the outliers within data sets. In addition ant clustering algorithm is capable to analyse different kind of data which can be divided into clusters of the hardly anticipated shapes on the

TABLE 1: Results of evaluation functions on k-means and *ACA* algorithms for Iris dataset.

| <b>Iris 150</b>                         | <i>k-means</i> | <i>ACA</i>    |
|---|----------------|---------------|
| Clusters                                | 3.000          | 2.960         |
| Rand Index                              | 0.824 (0.002)  | 0.785 (0.022) |
| F-measure                               | 0.821 (0.003)  | 0.773 (0.022) |
| Dunn Index                              | 2.866 (0.188)  | 2.120 (0.628) |
| Variance                                | 0.861 (0.049)  | 4.213 (1.609) |
| Class. err.                             | 0.176 (0.004)  | 0.230 (0.053) |
| The best results (according Rand Index) |                |               |
| Clusters                                | 3.000          | 3.000         |
| Rand Index                              | 0.829          | 0.814         |
| F-measure                               | 0.830          | 0.811         |
| Dunn Index                              | 2.939          | 2.306         |
| Variance                                | 0.899          | 1.486         |
| Class. err.                             | 0.167          | 0.187         |

TABLE 2: Results of evaluation functions on k-means and ACA algorithms for Wine dataset.

| <b>Wine</b>                             | <i>k-means</i> | <i>ACA</i>    |
|---|----------------|---------------|
| Clusters                                | 3.000 (0.000)  | 2.980 (1.140) |
| Rand Index                              | 0.903 (0.008)  | 0.832 (0.021) |
| F-measure                               | 0.928 (0.007)  | 0.855 (0.023) |
| Dunn Index                              | 1.395 (0.022)  | 1.384 (0.101) |
| Variance                                | 6.290 (0.020)  | 8.521 (0.991) |
| Class. err.                             | 0.071(0.007)   | 0.142 (0.030) |
| The best results (according Rand Index) |                |               |
| Clusters                                | 3.000          | 3.000         |
| Rand Index                              | 0.926          | 0.872         |
| F-measure                               | 0.943          | 0.896         |
| Dunn Index                              | 1.327          | 1.436         |
| Variance                                | 6.336          | 8.157         |
| Class. err.                             | 0.056          | 0.101         |

TABLE 3: Results of evaluation functions on k-means and ACA algorithms for Ionosphere dataset.

| <b>Ionosphere</b>                       | <i>k-means</i> | <i>ACA</i>     |
|---|----------------|----------------|
| Clusters                                | 2.000 (0.000)  | 2.560 (0.535)  |
| Rand Index                              | 0.578 (0.002)  | 0.563 (0.017)  |
| F-measure                               | 0.705 (0.002)  | 0.676 (0.037)  |
| Dunn Index                              | 1.211 (0.003)  | 1.031 (0.198)  |
| Variance                                | 23.167 (0.001) | 23.224 (2.224) |
| Class. err.                             | 0.301(0.002)   | 0.300 (0.017)  |
| The best results (according Rand Index) |                |                |
| Clusters                                | 2.000          | 2.000          |
| Rand Index                              | 0.582          | 0.586          |
| F-measure                               | 0.710          | 0.700          |
| Dunn Index                              | 1.212          | 0.841          |
| Variance                                | 23.109         | 23.743         |
| Class. err.                             | 0.296          | 0.291          |

grid files.

The scheme of  $\alpha$ -adaptation proposed originally in J. Handl is not as good as we assumed in our approach. this scaling parameter plays an important role in the clustering process, so its changing scheme of its values should be strongly connected to the effectiveness of the algorithm. This parameter is responsible to the cluster number. If the clusters on a few hierarchical levels exists, this version of ant clustering algorithm will identify the high level connections, so the generated clusters could be recursevily processed.

Future work consists in testing how this model with new ideas of learning process via pheromone updating rules scales with large databases. We are also

TABLE 4: Results of evaluation functions on k-means and ACA algorithms for Pima dataset.

| <b>Pima</b>                             | <i>k-means</i> | <i>ACA</i>      |
|---|----------------|-----------------|
| Clusters                                | 2.000 (0.000)  | 6.400 (1.590)   |
| Rand Index                              | 0.960 (0.020)  | 0.504 (0.013)   |
| F-measure                               | 0.678 (0.029)  | 0.473 (0.070)   |
| Dunn Index                              | 0.983 (0.029)  | 0.752 (0.140)   |
| Variance                                | 74.974 (1.835) | 45.226 (18.880) |
| Class. err.                             | 0.324 (0.023)  | 0.321 (0.016)   |
| The best results (according Rand Index) |                |                 |
| Clusters                                | 2.000          | 5.000           |
| Rand Index                              | 0.581          | 0.536           |
| F-measure                               | 0.709          | 0.623           |
| Dunn Index                              | 0.975          | 0.776           |
| Variance                                | 73.808         | 62.971          |
| Class. err.                             | 0.278          | 0.331           |

considering other biological inspirations from real ants for analysis a clustering problem, for example learning the template and other principles of recognition system.

### Acknowledgements

The research has been partially supported by the project “Decision support — new generation systems” of Innovative Economy Operational Programme 2007–2013 (Priority Axis 1. Research and development of new technologies) managed by Ministry of Regional Development of the Republic of Poland.

### References

- E. BONABEAU (1997), From Classical Models of Morphogenesis to Agent-Based Models of Pattern formation, *Artificial Life*, 3:191–209.
- E. BONABEAU, M. DORIGO, and G. THERAULAZ (1999), *Swarm Intelligence. From Natural to Artificial Systems*, Oxford University Press, New York.
- L. CHRETIEN (1996), *Organisation Spatiale du Matériel Provenant de L’excavation du nid chez Messor Barbarus et des Cadavres d’ouvrières chez Lasius niger (Hymenopterae: Formicidae)*, Ph.D. thesis, Université Libre de Bruxelles.
- J.-L. DENEUBOURG, S. GOSS, N. FRANKS, A. SENDOVA-FRANKS, C. DETRAIN, and L. CHRETIEN (1991), The Dynamics of Collective Sorting: Robot-Like Ant and Ant-Like Robot, in J. A. MEYER and S. W. WILSON, editors, *First Conference on Simulation of Adaptive Behavior. From Animals to Animats*, pp. 356–365.
- M. ESTER, H.-P. KRIEGLER, J. SANDER, and X. XU (1996), A density-based algorithm for discovering clusters in large spatial databases with noise, in E. SIMUODIS, J. HAN, and U. FAYYARD, editors, *Second International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, AAAI Press, Portland, USA.

- N. R. FRANKS and A. B. SENDOVA-FRANKS (1992), Brood Sorting by Ants: Distributing the Workload Over the Work Surface, *Behav. Ecol. Sociobiol.*, 30:109–123.
- V. GANTI, J. GEHRKE, and R. RAMAKRISHNAN (1999), Cactus-clustering categorical data using summaries, in *International Conference on Knowledge Discovery and Data Mining*, pp. 73–83, San Diego, USA.
- S. GUHA, R. RASTOGI, and K. SHIM (1998), Cure: an efficient clustering algorithm algorithm for large databases, in *ACM SIGMOD International Conference on the Management of Data*, pp. 73–84, Seattle, USA.
- H. GUTOWITZ (1993), Complexity — seeking Ants, unpublished report.
- M. HALKIDI, M. VAZIRGIANNIS, and I. BATISTAKIS (2000), Quality scheme assesment in the clustering process, in *Proceedings of the Fourth European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 265–267, Springer Verlag, vol.1910 of LNCS.
- J. HANDL, J. KNOWLEDGE, and M. DORIGO (2003), Ant-based clustering: a comparative study of its relative performance with respect to k-means, average link and ID-som, Technical Report 24, IRIDIA, Universite Libre de Bruxelles, Belgium.
- J. HANDL and B. MEYER (2002), Improved ant-based clustering and sorting in a document retrieval interface, in Springer VERLAG, editor, *PPSN — VII. Seventh international Conference on Parallel Problem Solving from Nature*, pp. 913–923, LNCS, Berlin.
- G. KARYPIS, E.-H. HAN, and V. KUMAR (1999), Chameleon: a hierarchical clustering algorithm using dynamic modeling, *Computer*, 32:32–68.
- L. KAUFMAN and P. RUSSEEUW (1990), *Finding groups in data: an introduction to cluster analysis*, John Wiley and Sons.
- E. LUMER and B. FAIETA (1994), Diversity and adaptation in Populations of Clustering Ants, in *Third international Conference on simulation of Adaptive Behavior: From animals to Animats 3*, pp. 489–508, MIT Press, Cambridge.
- J. MACQUEEN (1967), Some methods for classification and analysis of multivariate observations, in *5th Berkeley Symposium on Mathematics, Statistics and Probability*, pp. 281–296.
- S. A. OPRISAN, V. HOLBAN, and B. MOLDOVEANU (1996), Functional self-Organisation Performing Wide-Sense Stochastic Processes, *Phys. Lett.*, A 216:303–306.
- C. V. RIJSBERGEN (1979), *Information Retrieval, 2nd edition*, Butterworth, London.
- R. SKINDEROWICZ (2007), *Zastosowanie algorytmow mrowkowych do grupowania danych*, Master's thesis, Institute of Computer Science, University of Silesia.