



**HAL**  
open science

# Fast Ant-Inspired Clustering Algorithm for Web Usage Mining

Nicolas Labroche

► **To cite this version:**

Nicolas Labroche. Fast Ant-Inspired Clustering Algorithm for Web Usage Mining. IPMU 2006 - 11th International Conference on Information Processing and Management of Uncertainty, Jul 2006, Paris, France. pp.2668-2675. hal-01144862

**HAL Id: hal-01144862**

**<https://hal.science/hal-01144862>**

Submitted on 22 Apr 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Fast Ant-Inspired Clustering Algorithm for Web Usage Mining

Nicolas Labroche

Université Pierre et Marie Curie-Paris6, UMR7606, LIP6,  
8 rue du Capitaine Scott, F-75015 Paris, France  
nicolas.labroche@lip6.fr

## Abstract

In this paper, we propose a new agglomerative clustering algorithm named *Leader Ant* (LA) that improves the classical Leader clustering algorithm model [14] with a metaphor inspired by the chemical recognition system of ants. In our approach, each object of the data set is associated to the colonial odour of an artificial ant. At each iteration, a randomly chosen ant meets ants from each already existing nest to decide if it integrates this nest or if it creates its own nest. At the end of this iterative meeting process, the nests represent a partition of the initial data set. Similarly to the Leader algorithm, LA processes each data only once, which allows short computation times even on large data sets. LA is compared to other clustering algorithms such as k-means or AntClust [9] on artificial and real data sets. Finally, we briefly describe results obtained when applying LA on real Web usage data from a French museum Web site.

**Keywords:** clustering, artificial ants, Web usage mining.

## 1 Introduction

This paper proposes a new agglomerative clustering algorithm named *Leader Ant* (LA)

that is inspired by the chemical recognition system of ants and whose objectives are: (1) to mine more efficiently (very) large data sets (like Web users sessions), (2) to accept any input data type (numerical vectors, set of visited Web pages, ...) since an adapted normalized similarity measure has been designed, and (3) with minimum parameters settings (like the number of clusters, their expected shapes, ...). As our approach is conducted by an ant metaphor, we present hereafter a brief overview of some ant-based clustering algorithms.

Ant-based algorithms are optimization methods that rely on the modeling of collective behaviors of real ants. Lumer and Faieta [11] have developed a clustering algorithm that reproduces the ability of real ants to sort their cemetery or their brood. This algorithm uses a discrete grid on which data and artificial ants are randomly placed. Each ant can move across the grid and pick up or drop objects according to the similarity between an object that may be already carried by the ant, and the objects that may be in its neighborhood. This method has some limitations: numerous parameters to set and a difficulty to find the expected number of clusters. In [12], the authors propose a hybrid approach named Antclass in which a variant of the previous algorithm is used to generate the clusters seeds for a k-means algorithm. In this approach, the grid is toroidal to avoid undesirable side effects and each ant can drop several objects in the same location of the grid to avoid the problem of heap aggregation. However, AntClass can only handle numerical

data sets because of its hybridization with k-means algorithms.

More recently other computer scientists have proposed new variants such as Ramos et al. [13] with the ACluster algorithm which introduces the notion of bio-inspired spatial transition probabilities, in which artificial ants move on a grid according to pheromone trails whose densities are proportional to the number of objects in the neighborhood. In [6], Handl et al. introduce new mechanisms (short-term memory, adaptive scaling, jump in the 2D-grid, stagnation control, ...) to improve the original Lumer and Faieta's approach in the context of a visual tool for searching Web documents. This concept has been improved recently with the ATTA clustering algorithm [5] which determines automatically the number of clusters and which is robust to partition with overlapping or differing sizes clusters. The main limitation of these approaches is the size of the grid which may be too important for very large data sets (tens of thousands of objects).

In [9], Labroche et al. have proposed a clustering algorithm inspired by the chemical recognition system of ants named AntClust. In this model, each artificial ant possesses an odour representative of its nest membership called "label" and a genome which is associated to a unique object of the data set. The algorithm simulates meetings between artificial ants according to behavioural rules to allow each ant to find the label (or nest) that best fits its genome. AntClust has also been successfully applied to the Web sessions clustering problem.

We propose in this paper a new clustering algorithm named leader Ant that enriches the leader algorithm with an ant metaphor inspired by the chemical recognition system of ants, and that aims at mining very large data sets, with no hypothesis concerning the data type or the expected clusters.

This paper is organized as follows: section 2 introduces the Leader Ant clustering algorithm, section 3 evaluates its performances against other clustering algorithms such as k-Means and AntClust and section 4 shows

the results obtained when applying the Leader Ant algorithm to real Web usage from a French museum Web site. Finally, section 5 concludes and details some perspectives of this work.

## 2 The Leader Ant Algorithm

In the biological system, each ant is characterized by its own odour the label, partially defined by the genome of the ant, and a neuronal template representative of the labels of the nest members, that is learned during meetings. The recognition system relies on the comparison between the template of the ant and the perceived label of the encountered individuals.

The underlying model of the Leader Ant algorithm (LA), although inspired by real ants system, has been adapted to match more specifically the objectives of the clustering problem and for performance purposes. In LA, an artificial ant is described by three parameters:

- the genome is associated to a unique object of the data set;
- the template is the same for all artificial ants and is set to the mean value of  $Nb_{learn}$  similarity values  $s(i, j) \in [0, 1]$  estimated between  $Nb_{learn}$  couple of ants  $i$  and  $j$  randomly chosen.

$$Template \leftarrow \frac{\sum_{Nb_{learn}} s(i, j)}{Nb_{learn}}$$

The template is a real value between 0 and 1 since the similarity measure is normalized between 0 and 1;

- the label reflects the nest membership of each artificial ant. At the beginning, this value is set to zero as no hypothesis is made concerning the initial membership of ants.

LA is a one-pass agglomerative algorithm that iteratively selects at random a new ant  $a$  (that has not been already assigned to a

nest), and determines its label or nest membership by simulating  $Nb_{Meetings}$  meetings with randomly chosen ants from each existing nest  $k \in [0, K]$ .

During these meetings, the ant  $a$  estimates the similarity of its genome with those of ants from the evaluated nest  $k$ . At the end, the similarity  $S(a, k)$  between the ant  $a$  and the nest  $k$  is computed as the mean similarity over the  $Nb_{Meetings}$  meetings.

$$S(a, k) = \frac{\sum_{j=1}^{Nb_{Meetings}} s(a, ant_j^k)}{Nb_{Meetings}}$$

where  $ant_j^k$  is the  $j^{th}$ ,  $j \in [1, Nb_{Meetings}]$  randomly chosen ant from nest  $k$ .

If no nest exists or if the mean similarity value is under the template value, the ant creates its own new nest (or own label). In the opposite case, the ant joins the nest with the highest mean similarity value by setting its label as follows:

$$Label_a \leftarrow \operatorname{argmax}_{k \in [1, K]} S(a, k)$$

Finally, when all ants are assigned to a nest, the smallest nests whose size is under a fixed threshold set to  $MinSizeNest \times N$ ,  $MinSizeNest \in [0, 1]$  can optionally be deleted and their ants reassigned to the other clusters.

## 2.1 Experimental Protocol

### 2.1.1 The data sets

We use artificial and real data sets in our tests. The artificial data sets are named  $Art_{1,2,3,4,5,6}$  and have been generated according to gaussian or uniform laws with distinct difficulties (irrelevant attributes, clusters overlap) (see [12]). The real data sets are extracted from the Machine Learning Repository and are named : *Iris*, *Glass*, *Pima*, *Soybean* and *Thyroid*. We expect them to be more difficult to cluster, as real data may be more unpredictable than artificial ones. The table 1 sums up the main characteristics of these data sets. The fields for each data set

are: the number of objects ( $N$ ), their associated number of attributes ( $M$ ), and the number of clusters ( $K$ ).

Files	N	M	K
<i>Art<sub>1</sub></i>	400	2	4
<i>Art<sub>2</sub></i>	1000	2	2
<i>Art<sub>3</sub></i>	1100	2	4
<i>Art<sub>4</sub></i>	200	2	2
<i>Art<sub>5</sub></i>	900	2	9
<i>Art<sub>6</sub></i>	400	8	4
<i>Iris</i>	150	4	3
<i>Glass</i>	214	9	6
<i>Pima</i>	798	8	2
<i>Soybean</i>	47	35	4
<i>Thyroid</i>	215	5	3

Table 1: Main characteristics of the data sets.

### 2.1.2 The Error Measure

In order to estimate the performance of our algorithm, we use a mesure adapted from the measure developed by Fowlkes and Mallows as used in [8]. It evaluates the differences between two partitions by comparing each pair of objects and by verifying each time if they are clustered similarly or not. Let  $P_{exp}$  be the expected partition and  $P_{out}$  the output partition of a clustering algorithm. The clustering error  $E(P_{exp}, P_{out})$  can be defined as follows:

$$E(P_{exp}, P_{out}) = \frac{2 \times \sum_{(m,n) \in [1,N]^2, m < n} \epsilon_{mn}}{N(N-1)} \quad (1)$$

where  $\epsilon_{mn} =$

- 0 if  $(P_{exp}(m) = P_{exp}(n) \wedge P_{out}(m) = P_{out}(n)) \vee (P_{exp}(m) \neq P_{exp}(n) \wedge P_{out}(m) \neq P_{out}(n))$
- 1 otherwise

with  $N$  the number of objects in the data set.  $P_{exp}(o)$  (resp.  $P_{out}(o)$ ) is the cluster number of the object  $o$  in the partition  $P_{exp}$  (resp.  $P_{out}$ ).

### 2.1.3 The Similarity Measure

We use a similarity measure  $s(x, y)$  derived from the Minkowski distance at order 1 which

handles 2 objects represented as normalized vectors  $x$  and  $y$  with  $m$  attributes each as follows:

$$s(x, y) = 1 - \frac{\sum_{j=1}^m |x[j] - y[j]|}{m}$$

## 2.2 Parameters Settings

The Leader Ant clustering algorithm mainly depends from 3 parameters:

1.  $Nb_{learn}$ : the number of meetings to estimate the template,
2.  $Nb_{Meetings}$ : the number of meetings to estimate the similarity between an ant and a nest,
3.  $MinSizeNest$ : the threshold under which smallest nest are deleted.

To estimate each of these parameters we have conducted experiments on artificial and real data sets. As the results are similar, we only report here the results computed on artificial data sets. We evaluate the mean clustering error (see equation 1) over 50 runs for each data set.

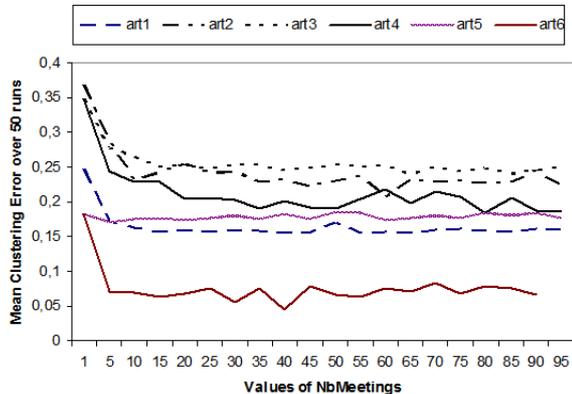


Figure 1: Mean clustering error for values of  $Nb_{Meetings} \in [1, 95]$ , with  $Nb_{Learn} = 0.2 * N$  and  $MinSizeNest = 0.05 * N$

The figure 1 presents the results obtained when  $Nb_{Meetings}$  varies from 1 to 95 meetings with the following parameters:  $Nb_{learn} = 0.2 * N$  and  $MinSizeNest = 0.05 * N$ . According to figure 1, the clustering error decreases rapidly until  $Nb_{Meetings}$  reaches the

value 10. For all values of  $Nb_{Meetings} > 10$ , results seems to be stable for all data sets. Consequently, the value of  $Nb_{Meetings}$  is set to 20 in the next experiments.

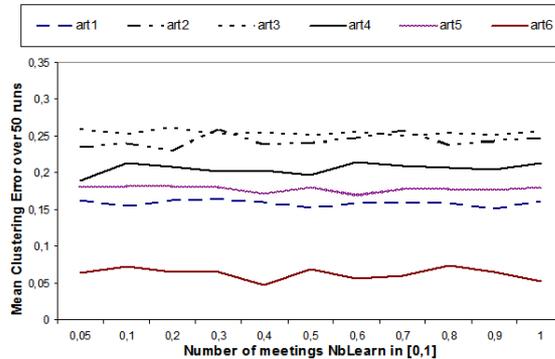


Figure 2: Mean Clustering Error for values of  $Nb_{Learn} \in [0.05, 1]$ , with  $Nb_{Meetings} = 20$  and  $MinSizeNest = 0.05 * N$

The figure 2 shows that, although the learning of an accurate estimation of the template is necessary to achieve good results, this estimation can be obtained with relatively few meetings. Consequently,  $Nb_{Learn}$  is set to  $10\% * N$  in the next experiments.

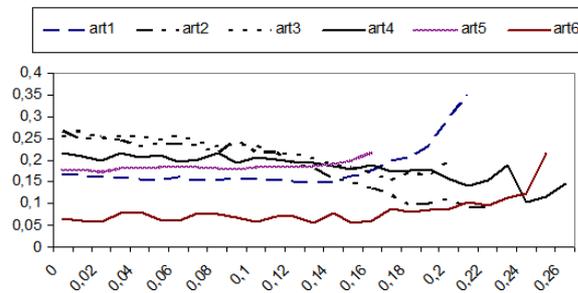


Figure 3: Mean Clustering Error for values of  $MinSizeNest \in [0, 0.31]$ , with  $Nb_{Meetings} = 20$  and  $Nb_{Learn} = 0.2 * N$

As expected, the figure 3 shows that the minimum size of a cluster  $MinSizeCluster$  mainly depends on the number of clusters in the data sets and their respective sizes. As the clustering error takes into account the deviation between the theoretical number of clusters and the number of clusters effectively found, the results are generally best when the nest deletion heuristic based on the  $MinSizeNest$  value provides the best estimation of the theoretical number of clusters. As expected, the

$MinSizeNest$  value that best estimates the number of clusters is not the same for every data sets because of their structures. However, the results are relatively stable, exception made of  $Art_2$  and  $Art_4$  which have 2 theoretical clusters and thus may need to delete more of the smallest nests. Thus the choice of a value for  $MinSizeCluster$  depends partly on the ratio performances/computation time the user needs, and partly on the inner structure of the data set. The figure 4 indicates for each artificial data set, the mean number of ants that are reassigned to a new nest for values of  $MinSizeNest \in [0, 0.31]$ . We propose to set  $MinSizeNest = 0.05$  in the next experiments to favorize the computation time.

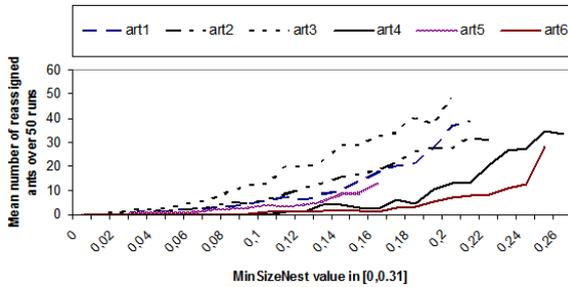


Figure 4: Mean number of reassigned ants versus the values of  $MinSizeNest \in [0, 0.31]$ , with  $Nb_{Meetings} = 20$  and  $Nb_{Learn} = 0.2 * N$

### 3 Evaluation of performances

This section presents the comparative evaluation of performances between Leader Ant ( $LA$  in tables 2 and 3) and other clustering algorithms such as AntClust ( $AC$  in table 2 and 3) and the well-known k-means algorithm ( $KM$  in table 2). The comparative results are presented in the table 2 which indicates for each data set, the mean clustering error values  $E$  (see equation 1), the mean number of clusters found  $\#$  clusters (and their respective standard deviation values). The table 3 presents the mean computation times in milliseconds computed over 50 runs on each data set for Antclust and leader Ant. Both programs are written in Java, and the computation times have been measured on a 1.1GHz ULV Pentium M processor with 1.24 Go RAM in the same experimental

Data sets	AC	LA
$Art_1$	204.98	26.26
$Art_2$	999.06	134.14
$Art_3$	1251.84	170.58
$Art_4$	62.54	8.12
$Art_5$	914.38	115.6
$Art_6$	212.82	27.2
$Glass$	79.7	11.28
$Iris$	40.02	5.3
$Pima$	683.7	135.86
$Soybean$	13.12	1.54
$Thyroid$	71.88	9.36

Table 3: Mean computation times for each data set and each method computed over 50 runs in milliseconds ( $AC = AntClust$ ,  $LA = leader Ant$ ).

conditions.

The table 2 shows that the clustering error values of AntClust and Leader Ant are comparable: on the one hand, AntClust ( $AC$ ) performs better than the Leader Ant ( $LA$ ) algorithm for data sets  $Art_2$ ,  $Art_3$  and  $Art_4$ , but on the other hand  $LA$  tends to be more efficient than  $AC$  on data sets  $Art_1$ ,  $Art_5$ ,  $Art_6$  and  $Iris$ . The error values are similar for the other real data sets. These results can be explained by the fact that the clustering error measure takes into account the ability of the algorithm to estimates the expected number of clusters and that according to the **# clusters** columns, AntClust better evaluates this number of clusters. This over-estimation of the number of clusters in the leader Ant algorithm is due to the choice of  $MinSizeNest = 0.05 * N$  as previous experiments showed. As  $LA$  is not favored by the error measure, we can conclude that for  $Art_1$ ,  $Art_6$  and  $Iris$  data sets, the Leader Ant algorithm performs better than AntClust. k-means has similar or best performances than  $LA$ . There may be three main reasons to explain that: first k-means computes the mean of each cluster when  $LA$  only estimates it, second, k-means is a multi-pass algorithm when  $LA$  builds its partition in one pass over the objects and third, k-means is set with the right number of clusters and is favored by our er-

Data sets	$E$			# Clusters		
	AC	LA	KM	AC	LA	KM
<i>Art<sub>1</sub></i>	0.29 ± 0.04	0.16 ± 0.02	0.11 ± 0.00	4.26 ± 1.07	5.04 ± 0.7	3.98 ± 0.14
<i>Art<sub>2</sub></i>	0.06 ± 0.02	0.26 ± 0.04	0.04 ± 0.00	2.06 ± 0.24	4.58 ± 0.7	2.00 ± 0.00
<i>Art<sub>3</sub></i>	0.14 ± 0.02	0.25 ± 0.03	0.22 ± 0.02	2.26 ± 0.53	5.5 ± 0.76	3.84 ± 0.37
<i>Art<sub>4</sub></i>	0.13 ± 0.05	0.21 ± 0.08	0.00 ± 0.00	3.22 ± 0.65	3.66 ± 0.69	2.00 ± 0.00
<i>Art<sub>5</sub></i>	0.34 ± 0.04	0.18 ± 0.03	0.09 ± 0.02	4.82 ± 1.34	4.98 ± 0.8	8.10 ± 0.75
<i>Art<sub>6</sub></i>	0.13 ± 0.03	0.07 ± 0.06	0.01 ± 0.04	4.28 ± 0.67	3.84 ± 0.47	4.00 ± 0.00
<i>Glass</i>	0.4 ± 0.03	0.34 ± 0.05	0.32 ± 0.01	4.3 ± 1.16	3.78 ± 0.84	6.88 ± 0.32
<i>Iris</i>	0.22 ± 0.00	0.17 ± 0.04	0.14 ± 0.03	2.42 ± 0.64	3.1 ± 0.54	2.96 ± 0.20
<i>Pima</i>	0.45 ± 0.01	0.48 ± 0.02	0.44 ± 0.00	7.12 ± 2.29	6.22 ± 1.07	2.00 ± 0.00
<i>Soybean</i>	0.12 ± 0.04	0.14 ± 0.1	0.09 ± 0.08	3.76 ± 0.48	3.48 ± 0.76	3.96 ± 0.20
<i>Thyroid</i>	0.2 ± 0.03	0.23 ± 0.09	0.18 ± 0.00	4.26 ± 1.07	3.26 ± 0.72	3.00 ± 0.00

Table 2: Mean clustering error  $E$ , number of clusters (and their standard deviations) for each data set and each method computed over 50 runs.

ror measure.

The theoretical complexity of AntClust is linear with the number of objects to cluster (in  $O(NK)$ ) when the number of cluster  $K \ll N$ . The leader Ant algorithm has also a theoretical complexity in  $O(NK)$  (when  $K \ll N$ ) as each of the  $N$  artificial ants is only considered once and compared to the clusters that already exist (and whose number is bounded by  $K$ ). The table 3 indicates for each data set, the mean computation time in milliseconds over 50 runs. It clearly shows that *LA* outperforms *AC* by running from approximately 5 (for data set *Pima*) to 8.5 (for data set *Soybean*) times faster. This difference is partially due to the relative complexity of AntClust compared to the leader Ant algorithm, in which artificial ants behavior is mainly conducted by a single rule to decide if an ant joins an existing nest or if it builds its own. Furthermore, AntClust is a succession of “sub-algorithms” (ants initialization with individual template learning, random meetings with other ants, nests selection and nest finalization) whereas the leader Ant algorithm is more simple (with only one pass over the data set). Finally, the Java structures in AntClust, although optimized, seems to be much more memory consuming, but further experiments should be conducted to definitively concludes.

## 4 Application to the Web Usage Mining problem

The Web Usage Mining aims at understanding the navigation of users on Web sites by inferring their goals and their motivations from the stream of requests made during their navigation sessions. Several works already focused on the discovery of typical Web browsing patterns with clustering algorithms [14, 7, 4, 10, 1, 3]. These studies generally use numerical vector representations of Web users navigations, in which each component is associated to a Web page and indicates the number of times it has been accessed during the session (hits vectors). The problem is that these vectors are often large with many components set to zero as the corresponding Web pages have not been visited. As it is not possible to store efficiently all these sessions in the RAM memory of a computer at the same time, we propose hereafter a simple approach to normalize and compare hits vectors.

### 4.1 Hits vectors normalization and comparison

We define the hits vectors on a subset of the Web site urls equals to the union of the set of urls visited at least once during each compared session. Then, we create each session as a numerical vector defined on this new set of urls and we normalize each component of this new vector by its maximum value in order

to apply the usual similarity measure.

## 4.2 Experiments and results

The experiments have been conducted on Web users sessions reconstructed from the pre-processed Web log file of the Cap Sciences museum Web site (<http://www.cap-sciences.net>). Our Web log file contains 39004 sessions that visited 3775 distinct urls (after filtering a priori uninteresting documents such as images); the Web log file was recorded during one month and a half (during march and april 2005).

The Cap Sciences Web site is a complex Web site to analyse since it contains two main parts that are closely related: Cap Sciences and Info Sciences; the first aims at promoting the exhibitions of the museum whereas the second is interested in scientific popularization. Moreover, some distinct urls lead to the same content which complicates the interpretation of clustering results.

As the analysis of Web user sessions generally produces (very) small clusters that may be representative of a minority trend in the Web users behaviors, we set  $MinSizeNest = 0.01$  in our experiment. Our algorithm produces 9 clusters.

The 1<sup>st</sup> cluster contains 31733 sessions which corresponds to more than 81% of the total number of Web sessions. The Web sessions of this cluster mainly visit the “index” pages of the CapSciences part of the Web site but also some of the other pages more related to the editorial content such as pdf files and events page.

The 2<sup>nd</sup> cluster contains 3243 sessions. These are short sessions (with mainly less than 4 hits) that only visited the index page of the Info Sciences Web site.

The other clusters contain between approximately 400 to 1300 sessions and each visited only one url exclusively. In other words, web sessions of these clusters may have visited (depending on their session length) several times the same url consecutively within 30 minutes. It is interesting to notice that

2 clusters which “visited” an index url (/default.asp) and a tourism link which are easily accessible, have very short sessions. On the other hand the 5 other clusters are only focused on the download pages for pdf documents and have much longer sessions (until 44 hits in the same session).

These results can be easily explained: our vector normalization heuristic represents a session by giving each accessed url a value equal to its representativity in this session. Thus, two compared sessions are similar if they have visited the same pages in the same proportion. This mechanism favors clusters interested in only one url and the grouping of the other sessions in the cluster that is not clearly defined. Despite this, the results may allow a webmaster to understand the type of access that are made on her Web site and highlight some of the main users center of interest (the pdf files for example).

## 5 Conclusion and perspectives

This paper proposes a new clustering algorithm (the leader Ant) that improves a classical clustering model with an ant metaphor inspired by the chemical recognition system of ants. It has been designed to be more efficient on large data sets than previous ant-based clustering algorithm such as AntClust, with minimum parameters to set and the possibility to analyse any data type since an adapted similarity measure is available. We have shown, on artificial and real data sets, that the leader Ant algorithm could perform as well as AntClust in term of clustering error (and sometimes better), and outperforms AntClust when considering the computation times (sometimes until 8 times faster). The leader Ant algorithms is also applied to real Web sessions clustering problem and allows to find small meaningful clusters that may help a webmaster to better understand the main center of interests of her Web site.

We are currently working on improving both our clustering algorithm (by studying distinct criteria to compute the similarity between an ant and an existing nest) and the Web usage

mining application (by introducing Web sessions representations as Web pages sequences for example). We also plan to introduce fuzzyness in our model and comparisons with fuzzy c-means [2] in our future experiments.

## References

- [1] R. Baraglia and P. Palmerini. Suggest: A web usage mining system. In *Proceedings of IEEE International Conference on Information Technology: Coding and Computing*, April 2002.
- [2] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.
- [3] B. Davison. Predicting web actions from html content. In *Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia (HT'02)*, pages 159–168, College Park, MD, June 2002.
- [4] Y. Fu, K. Sandhu, and M. Shih. Clustering of web users based on access patterns. In Springer-Verlag, editor, *Proceedings of the 1999 KDD Workshop on Web Mining*, San Diego, CA, 1999.
- [5] J. Handl, J. Knowles, and M. Dorigo. Ant-based clustering and topographic mapping. In *Artificial Life*, volume 12(1), 2005.
- [6] J. Handl and B. Meyer. Improved ant-based clustering and sorting in a document retrieval interface. In J. M. G. et al. (Eds.), editor, *Proc. of the 7th International Conference on Parallel Problem Solving from Nature (PPSN VII)*, Granada, Spain, 7-11 September 2002. Springer-Verlag.
- [7] J. Heer and E. Chi. Identification of web user traffic composition using multi-modal clustering and information scent, 2001.
- [8] J. Heer and E. Chi. Mining the structure of user activity using cluster stability. In *Proceedings of the Workshop on Web Analytics, SIAM Conference on Data Mining*, April 2002.
- [9] N. Labroche, C. Guinot, and G. Venturini. Fast unsupervised clustering with artificial ants. In X. Y. et al. Editors, editor, *Proceedings of the Parallel Problem Solving from Nature 2004 (PPSN VIII)*, pages 1143–1152, Birmingham, England, 18-22 september 2004.
- [10] N. Labroche, N. Monmarché, and G. Venturini. Antclust: Ant clustering and web usage mining. In *Proceedings of the Genetic and Evolutionary Computation Conference (Gecco 2003)*, Chicago, USA, July 12-16 2003.
- [11] E. Lumer and B. Faieta. Diversity and adaptation in populations of clustering ants. In D. Cliff, P. Husbands, J. Meyer, and S. W., editors, *Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 501–508. MIT Press, Cambridge, Massachusetts, 1994.
- [12] N. Monmarché, M. Slimane, and G. Venturini. On improving clustering in numerical databases with artificial ants. In D. Floreano, J. Nicoud, and F. Mondala, editors, *Lecture Notes in Artificial Intelligence*, pages 626–635, Swiss Federal Institute of Technology, Lausanne, Switzerland, 13-17 September 1999. Springer-Verlag.
- [13] V. Ramos and J. Merelo. Self-organized stigmergic document maps: Environment as mechanism for context learning. In *Proceedings of the First Spanish Conference on Evolutionary and Bio-Inspired Algorithms (AEB' 2002)*, Centro Univ. de Mérida, Mérida, Spain, 6-8 February 2002. E. Alba, F. Herrera, J.J. Merelo et al.
- [14] T. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal. From user access patterns to dynamic hypertext linking. In *Proc. of 5th WWW*, pages 1007–1014, 1996.