Advanced Ant Colony Algorithm based on unsupervised classification for the construction of dynamic graph

Kumud, Azad Kumar Shrivastava

Mewar University, Gangrar,
Chittorgarh, India.

Abstract:

We present a new unsupervised classification algorithm hierarchical construction of dynamic graphs called Advanced Ant Colony Algorithm (AACA). It uses the principle of the chemical recognition system observed in real ants applied on Cluster graphs [1,2]. We detail the algorithm and local rules of ant behavior artifacts and the resulting evolution on the cluster graph from the study that was carried out on the mechanisms fundamentals of the chemical recognition system in real ants. We also described all the tests that were used to validate the AACA algorithm, studying in particular the results obtained on games of real digital and binary data, as well as the evaluation indices so judge the quality of the classification obtained. In the display part of the graphs generated, we will present the neighborhood graphs of some databases tested by the algorithm AACA. We conclude by presenting the interests of this approach as well as the extensions possible to propose.

Introduction:

We propose in this paper a new modeling that has been applied to solve a major problem in computer science. This new model is in line with previous work on algorithms Classification based on the behavior of real ants [3] and more generally biological systems [4]. It's about modeling the way ants form groups with the same colonial smell and use this behavior to organize data according to a graph that is built in an evolutionary way. These algorithms can benefit from interesting properties like those of produce a global optimization of the classification avoiding the minimum premises (probabilistic construction) and in a "reasonable" calculation time, to process real data or the need for prior information (number of classes, initial classification) from an initialization by the algorithm K-Means [5,6].

The principle used is as follows: intuitively, each ant data is initially located in a cluster, an ant extracted from the list of ants belonging to a nest cluster [7], decides to leave his group and to integrate to another neighboring nest cluster that is the most similar. The behavior of Ants involves moving from one cluster to another. This behavior is determined in particular by the similarity between the data and the acceptance threshold nest cluster. This results in a dynamic graph structure of data whose properties will allow us to deal with many applications : automatic determination of partitioning (which we use especially as a comparison with other methods), and finally a visual overview of graph. This paper therefore relates to the presentation of the first version of the algorithm non-hierarchical [8] and unsupervised classification [9] named AACA.

We will detail the algorithm to form a graph structure and the local rules of behavior of artificial ants from the study that was performed on the building behavior of a colonial scent in real ants. Then we will present all the tests that were used to validate the AACA algorithm, studying in particular the results obtained on numerical and categorical data sets actual binary data, as well as quality evaluation indices of the classification obtained. We will study in the next chapters the extension of this algorithm that processes data streams in a way incremental. We will then compare the proposed algorithms to the algorithm K-Means, the Hierarchical Ascending Classification (CAH) and the difierent incremental algorithms like D-Stream [10], DenStream [11] and ClusStream [12].

Method:

To obtain a data classification, we will build a graph whose nodes are known a priori from an initialization step by K-Means algorithm and represent clusters of data, and whose arcs are presented to identify the neighborhood relationship between the groups of data. Note that the constructed graph is a complete and dynamic graph will contain data at all nodes. We will show in results of section, visualization of graphs after each step of the algorithm. Each node represents the group of data to be classified. We consider that we have a similarity measure Sim (*Wi;Wj*) which takes as parameter a pair of centroids *CLi* clusters and *CLj*. We make no other assumptions about the representation of data that can be of any type of representation (numerical, categorical), it suffices that we can define a measure of similarity.

Each ant *fi* is located in a node of the graph, it is to say each data of the node to classify. The graph represents the structure in which ants will move to build similar groups of data. Each *fi* ant is simulated, each represents a data initially located in a cluster [13], and each nest represents a cluster. Starting from a first partition that is initialized by the KMeans algorithm, the data is assigned to each class K whose center is the most close in terms of Euclidean distance. This partition represents the structure undirected full graph, the ants will move from one node to another neighbor node, then successively to the other nodes already connected in the graph [14, and so on until all the ants are affected to the most similar clusters. All these displacements and these assignments depend on the value returned by the similarity function Sim (*fi;Wj*) [15,16] between the data, and the center of gravity of the cluster. We want use these behaviors to organize the most similar data in the same group technically called cluster. For each fi we will therefore define the following concepts:

- K clusters CL_j or nodes are initialized by the algorithm K-Means to build the graph structure. This number fixed a priori represents nests formed by f_i ants moving from a nest to another neighbor's nest trying to find the one with the same smell chemical, Each cluster CL_j has a center of gravity W_j representing the center of severity of each cluster *j*.

- An acceptance threshold *Th CLj* , each cluster *CLj* makes it possible to define whether the given *di* is sufficiently similar or suficiently dissimilar other data represented by the cluster;

- consider also that the neighborhood between two clusters is represented by the pheromone rate *Phij* (with *i; j* the indices of the two clusters neighbors *CLi* and *CLj*).

During the construction of the structure, each of the ants will be either:

- Traveling on the graph [17]. In this case, we consider each ant moves to cluster *CLj*: moving from one to another, the substance on the ground a substance called pheromone that creates a certain type of a chemical track. Ants can smell pheromones who play the role of path markers: Ants tend to choose the route with the highest concentration of pheromones.

- Assigned to a new neighbor cluster. An update of prototypes are carried out. The weight of the edge is calculated from a distance, Depending on the nature of the data, which connects the two centers of the two poles (source and target). This weight represents the pheromone rate that will increase, after the assignment, between the source cluster and the one accepted by the new ant as a new target.

AACA algorithm works in the following way (algorithm 1): for a number of fixed iterations, an ant taken from the list of ants goes integrate or move to another nearby nest based on similarity with his neighborhood. As long as there is an ant on the move, we simulate an action for *fi* according to its position on the edge of the graph, that is to say that she decided to leave the source cluster to another more similar one. Initially, ants are placed in K clusters from the algorithm K-Means. They are moving in a structure of the dynamic graph. The similarity threshold is calculated

according to a distance and which can be between 0 and 1. For these two parameters we will quote other parameters put in place for the execution of this algorithm. The ants are sequentially tested. Then for each ant *fi* distinguishes 4 steps:

- K-Means algorithm initialization step: K-Means algorithm [MacQueen 1967] chooses K first points representing the centers of K classes (center of gravity). Delta, a first partition is formed in assigning each datum to the class K in which the center is the most close in terms of distance measurement.

- Decision of the ant to leave its cluster: once in its nest (cluster) each ant has a similarity to the center of the cluster, if Sim (*fi;Wj*) <cluster threshold *CLj* , where *Wj* is the center of *CLj*. The ant decides to leave his cluster and look for another cluster that seems closest.

- Moving the ant: the ant leaves its group, it moves to another cluster by choosing the path that has a stronger pheromone concentration. If it is not accepted by a cluster in its proximity, it moves to another by always taking the path the densest.

- Assignment of the ant fi in the new neighbor cluster: the ant is accepted by the cluster if Sim (*ai;Wj*) fi threshold of the nest (cluster) *CLj* then it updates the identifier of its class, and the destination cluster threshold as well as that of the original cluster. The path that leads the ant to his original cluster to the destination cluster will also be updated.

We have developed a model of behavioral rules of artificial ants for unsupervised clustering using the principles of smell colonial and traces of pheromones in real ants. We validate this approach with a comparative study in the next chapter. Ants go through two main stages: they borrow the path with a maximum pheromone rate, then a possibility to integrate into a new neighboring cluster.

Algorithm 1: General principle of the algorithm by artificial ants.

1: Beginning

2: Initialize the clusters by the K-Means algorithm

3: Initialize ant parameters and *CLj* clusters

- Calculate the similarity Sim (*fi;Wk*) is the distance between the ant and the cluster center of gravity *Wk*

- Calculate the nest acceptance threshold (cluster)

- Specify the identifier of the ant

4: For (a number of iterations) do

5: If (Sim (*fi;Wj*) <the threshold of acceptance of the nest *CLj* then

6: The decision of an ant to leave his cluster CLj) of belonging

7: Moving the ant

8: If (Sim (*fi;Wj*) the threshold of acceptance of the nest CLj) then

9: Assignment of ant to the nearest clutser that seems most similar to him

10: Update (the identifier of the ant fi, Sim (*fi;Wj*), the acceptance threshold *CLJ* and *CLj*, the pheromone rate of the path taken by the *fi* ant)

11: End

Algorithm 2 Detail view of the AACA algorithm.

1: Initialization: compute clusters using the K-Means algorithm

2: Building a complete graph linking all the clusters initialized by KMeans

3: Initialization of the parameters:

α: incremental value of the

ϒ: evaporation value of pheromone

Maximum number of iterations NB = 1000

Each centroid of the cluster Wj, and each threshold of the cluster *ThAACAj*

The pheromone rate for each edge *Phij* = 0

4: Tantque (NB iterations)

5: For each ant fi; *CLj* is the cluster of which it belongs

7: If (Sim (*fi;Wj*) < *ThAACAj*) then

8: Initialize the taboo list with *l*; corresponding to all clusters to explore them

9: success = false

10: So long (unsuccessful and the taboo list is not empty)

11: *fi* moves to another cluster *CLj* , *j* is argmax

19: End If

20: Evaporate the pheromone rate on all the edges of the graph *Phhl*

21: End Tantque

22: End If

23: If (not successful) then

24: *fi* remains in its initial cluster *CLj*

25: End If

26: End For

27: End Tantque

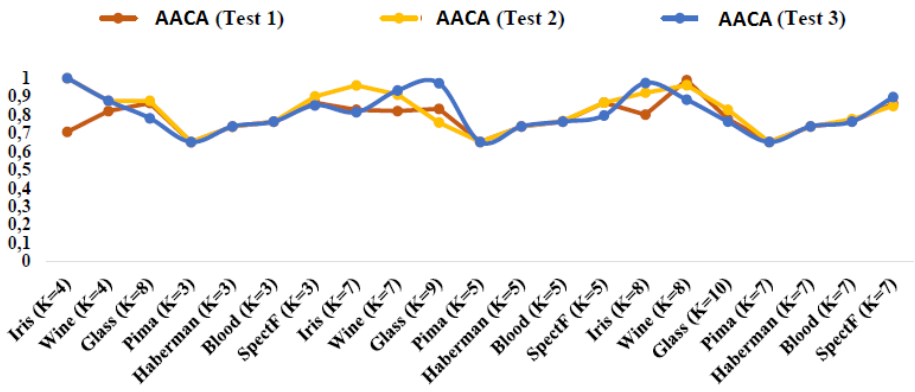

Figure 1: The results of the purity index obtained with AACA (Test 1, Test 2, Test 3) on digital databases after 1000 iterations

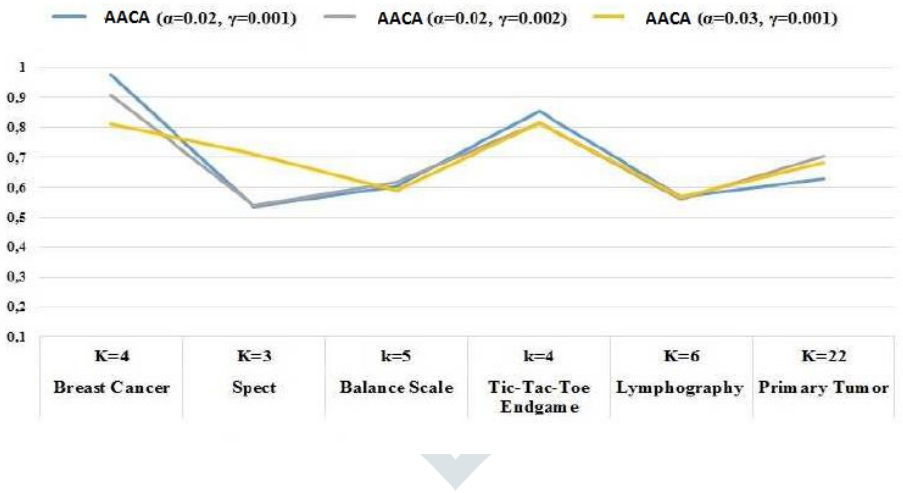

Figure 2: The results of the purity index obtained with AACA 2 (Test 1, Test 2, Test 3) on binary databases after 1000 iterations

Results:

Figures **1**, and **2** represent the results obtained for the tests that we realized. From this information, we can perform several findings. We test our method with difierent values from *fi* and in Figure **1**, we obtain the highest purity value is equal to 1 for the Iris database with K = 4, where AACA (Test 3) works well, with the exception of the Wine database with K = 4. that the AACA results with three tests were comparable: they are about for some databases and close enough to some of others with other databases. The best result in terms of index of AACA (Test 3) with = 0.03 and = 0.001 compared to AACA (Test 1) and AACA (Test 2) on the same number of In Figure 4.4, we obtain the highest purity value is equal to 1 for the Iris database with K = 4, where AACA (Test 3) works well, with the exception of the Wine database with K = 4.

The AACA results with three tests were comparable: they are about equal for some databases and close enough to some of others with other databases. The best result in terms of index of purity is obtained mainly by AACA (Test 3) with fi = 0.03 and = 0.001 compared to AACA (Test 1) and AACA (Test 2) on the same number of digital databases. On the other hand, for binary databases, we find in the figure **2** that the purity obtained by AACA (Test 1) with *fi* = 0.02 and = 0.001 are broadly similar to those obtained by AACA (Test 2) and AACA (Test 3) for both balance scale and lymphography databases. We gets a significant purity value is equal to 0.9755 for the base of breast cancer data with K = 4 after 1000 iterations. The best result is obtained in majority by AACA (Test 1) with fi= 0.02 and = 0.001 for the binary databases by comparing with the other two tests of AACA 2. Iterations on difierent digital databases and tests performed by the AACA 2 algorithm with the difierent values *fi* and (Test 1, Test 2, Test 3). AACA (Test 2) and AACA 2 (Test 3) produced the best results on the basis of Wine data with
K = 4 and Iris with k = 3 in the 1000.

Conclusion:

In this paper we have described a new approach inspired by the principles of chemical recognition in real ants especially the building a common colonial smell in a nest. Initially time, a summary of current knowledge in the field of chemical recognition and colonial identification in societies ant has been proposed. By studying and analyzing this information, a model could be put in place in a second time. By relying on biological behaviors already known and observed by the specialists of field. A computer implementation of this algorithm to be conducted to solve the problem of nonhierarchical classification unsupervised.

The main interest of our approach is that it does not depend on the type of data to be classified; it sufices to define a similarity function for these new data. Indeed we tested our algorithm on numerical, categorical (binary) data. A comparison later will be presented on digital data with other classical methods of classification. However, the model has limitations as the choice of initialization by the K-Means algorithm, the values update of pheromone levels fi and In the next chapter, we introduce an incremental version of the algorithm called AACA 2, which adapts to the coming of new data without redo the entire calculation allowing to manage data.

References:

[1] L.M. Gambardella et M. Dorigo. Ant-Q : A Reinforcement Learning Approach to the Travelling Salesman Problem. In A. Prieditis et S. Russell, editeurs, Proceedings of the Twelfth International Conference on Machine Learning, pages 252{260. Morgan Kaufmann, San Mateo, California, 1995.

[2] S. Garnier. Dfiecisions collectives dans des systfiemes d'intelligencenen essaim. Thfiese doctorat fia l'Universitfie Toulouse III - Paul Sabatier, 2008.

[3] Max H. Garzon. Cayley automata. Theor. Comput. Sci., vol. 108, no. 1, pages 83{102, 1993.
[4] M.H. Garzon, R.J. Deaton et K. Barnes. On Self-Assembling Graphs in vitro. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela et Robert E. Smith, editeurs, Proceedings of the Genetic and Evolutionary Computation Conference, volume 2, pages 1805{1809, Orlando, Florida, USA, 13-17 Juillet1999. Morgan Kaufmann.

[5] D. M. Gordon. The organization of work in social insect colonies. Nature, pages 121{124, 1996. (Citfie en pages 16 et 43.)

[6] P.P. Grassfie. La reconstruction du nid et le travail collectif chez les termites supfierieurs Y.Psychol. pages 370{396, 1939.

[7] P.-P. Grassfie. La reconstruction du nid et les coordinations interindividuelles chez bellicosi- termes natalensis et cubitermes sp. la theorie de la stigmergie : essai d'interpretation des termites constructeurs. Insectes sociaux, IV(1)., pages 41{83, 1959.

[8] N. Guelzim, S. Bottani, P. Bourgine et F. Kepes. Topological and casual structure of the yeast transcriptional regulatory network. Na-161 ture genetics, pages 60{63, 2002.

[9] L. S. Gueron S. A. Self-organization of front patterns in large wildebeest herds. Journal of theoretical biology, pages 541{ 552, 1993.

[10] S. Guha, R. Rastogi et K. Shim. CURE : an eficient clustering algorithm for large databases. In Laura M. Haas et Ashutosh Tiwary, editeurs, Proceedings ACM SIGMOD International Conference on Management of Data, pages 73{84, Seattle, Washington, USA, 6 1998. ACM
Press. (Non citfie.) :1998df :1998df :1998df :1998df

[11] H. Hacid et D-A. Zighed. An Efiective Method for Locally Neighborhood Graphs Updating. In DEXA 2005, pages 930{939, 2005.

[12] H. Hacid et D-A. Zighed. Graphes de Proximitfie pour l'Indexation et l'Interrogation d'Images par le Contenu. In 6fiemes Journfiees Francophones Extraction et Gestion des Connaissances (EGC 06), Lille, volume E-6 of Revue des Nouvelles Technologies de l'Information, pages 11{22, Toulouse, Janvier 2006. Cfiepadufies.

[13] H. Hacid et T. Yoshida. Incremental Neighborhood Graphs Construction for Multidimensional Databases Indexing. In the Canadian Arti-ficial Intelligence Conference, CanAI 2007, LNAI 4509

[14] H. Haken. Self-organization. Scholarpedia. 2008.

[15] J. Halloy, G. Sempo, G. Caprari, C. Rivault, M. Asadpour, F. Tache, I. Said, V. Durier, S. Canonge, J. M. Ame, C. Detrain, N. Correll, A. Martinoli, F. Mondada, R. Siegwart et J. L. Deneubourg. Social Integration of Robots into Groups of Cockroaches to Control Self-Organized
Choices. Science., page 11551158, 2007.

[16] J.A Hartigan. clustering algorithms. John wiley and Sons, New York., 1975. 162

[17] J. A. Hartigan et M. A. Wong. A k-means clustering algorithm. JSTOR : Applied Statistics, vol. 28, no. 1, pages 100{108, 1979. (Non citfie.) :1979ct :1979ct :1979ct :1979ct