An hybridization of an ant-based clustering algorithm with growing neural gas networks for classification tasks *

Marco A. Montes de Oca, Leonardo Garrido and José L. Aguirre
Center for Intelligent Systems
Monterrey Institute of Technology
Eugenio Garza Sada 2501
Monterrey, N.L. México 64849
{A00788072, leonardo.garrido, jlaguirre}@itesm.mx

ABSTRACT

Conventional ant-based clustering algorithms and growing neural gas networks are combined to produce an unsupervised classification algorithm that exploits the strengths of both techiques. The ant-based clustering algorithm detects existing classes on a training data set, and at the same time, trains several growing neural gas networks. On a second stage, these networks are used to classify previously unseen input vectors into the classes detected by the ant-based algorithm. The proposed algorithm eliminates the need of changing the number of agents and the dimensions of the environment when dealing with large databases.

Categories and Subject Descriptors

I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*; I.2.6 [**Artificial Intelligence**]: Learning—*Connectionism and neural nets*

Keywords

Ant-based algorithms, Classification

1. INTRODUCTION

Collective behaviors of social insects are the outcome of a process of self-organization [10]. Bonabeau et al. [3] define self-organization as “a set of dynamical mechanisms whereby structures appear at the global level of a system from interactions among its lower-level components” (p. 9). An example of such behavior can be seen in some termite genera such as *Marcotermes* and *Cubitermes* that create nests of astonishing complexity [14].

*Work supported by the Research Program in Intelligent Agents CAT-011. Tecnológico de Monterrey. Campus Monterrey.

SAC'05 March 13-17, 2005, Santa Fe, New Mexico, USA

In an attempt to explain the dynamics of nest building in these termite genera, Pierre-Paul Grassé [8], proposed that *stigmergy*, the indirect influence on the behavior of others through local environment modifications, was a key factor in that process. He observed that when workers of *Macrotermes bellicosus* were placed in a container with some soil pellets, the insects carried about and put down pellets in an apparently random fashion after an exploration phase where they moved through the container without taking any action. At this stage, a pellet just put down by a termite worker is often picked up and placed somewhere else by another worker. When a pellet is placed on top of another, the resultant structure appears to be much more attractive and termites soon start piling more pellets nearby, making the dropping spot even more attractive [14].

This phenomenon have inspired the creation of clustering algorithms for exploratory data analysis where insects are simulated by simple reactive agents acting in a two-dimensional grid [4, 13].

Neural networks have also been used in cluster analysis [6, 11]. Self-organizing feature maps or SOFM's [12] are particularly attractive for this task as similar input patterns are grouped together adaptively and are represented by a single neuron. However, the maximum number of clusters that a SOFM can learn is limited by the number of neurons, which must be set *before* the training phase. This is one of its major drawbacks for clustering tasks because, in general, the number of clusters is unknown. To overcome this problem, several network architectures that adjust the number of neurons during the training phase have been proposed [1, 5, 7]. In this paper, we propose an hybridization of an ant-based clustering algorithm with growing neural gas networks [7] consisting in a growing neural gas network embedded in each agent participating in the ant-based algorithm. Every time an agent finds data elements in its neighborhood, it trains its network by making it learn both, the topology of data elements in attribute space and their spatial distribution in the two-dimensional grid that serves as the ant environment. In this way, the clusters discovered by the ant-based algorithm become classes for a post-clustering classification phase where the trained networks are used as classifiers. The proposed hybridization makes use of growing neural gas networks because the number of clusters to be found is unknown and the ant-based clustering algorithm dynamically creates and destroys clusters until a steady state is reached.

In sections 2 and 3 we present background information on basic ant-based clustering and growing neural gas networks. In section 4 we present in detail the proposed hybrid model. Section 5 explains the experimental strategy used to test our model. Section 6 summarizes the obtained results. In section 7 we discuss our model in the light of experimental evidence and speculate on future extensions and improvements. Finally, we conclude in section 8.

2. ANT-BASED CLUSTERING

Ant-based clustering was introduced by Deneubourg et al. [4] using a model for spatial sorting. A group of agents exhibiting the same behavior move randomly over a toroidal square grid where objects were initially scattered in a random fashion. The objects can be picked up, moved or dropped in any free location on the grid. An object is picked up with high probability if it is not surrounded by other objects of the same type and is dropped by a loaded agent if its neighborhood is populated by other objects of the same type and the location of the agent has no object on it.

Lumer and Faieta [13] generalized Deneubourg et al. [4] model to apply it to exploratory data analysis through the creation of clusters of related data. All classical clustering algorithms depend on a similarity or dissimilarity measure to determine whether two objects are similar or not. This algorithm is no exception. However, instead of taking as input a similarity or dissimilarity matrix, this algorithm starts with something very similar to what we described in the last paragraph. The probability of picking a data element i is defined as

$$p_p(i) = \left(\frac{k_p}{k_p + f(i)} \right)^2 \tag{1}$$

where k_p is a constant and $f(i)$ is a similarity density measure with respect to element i. Likewise, the probability of dropping a data element is given by

$$p_d(i) = \begin{cases} 2f(i) & \text{if } f(i) < k_d \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

where k_d is a constant. The similarity density $f(i)$ for an element i, at a particular grid location τ, is defined as

$$f(i) = \begin{cases} \frac{1}{s^2} \sum_{j \in Neigh(\tau)} \left(1 - \frac{d(i,j)}{\alpha} \right) & \text{if } f > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where s^2 is the size of the perception area centered at the location of the agent and α is a scaling factor of the dissimilarity measure $d(i,j)$ between elements i and j. If it is assumed that data elements can be represented as points in an n-dimensional space, then the Euclidean distance could be used as dissimilarity measure.

In some cases, however, it may be useful to use a simmilarity measure, such as the cosine metric. In those cases, expression 3 is not directly applicable. The following expression could be used instead

$$f(i) = \frac{1}{s^2} \sum_{j \in Neigh(\tau)} \left(\frac{1}{1 + e^{-S \frac{d(i,j)}{\alpha} + D}} \right) \tag{4}$$

where S is the steepness of the response curve and D serves as a displacement factor. In our experiments we fixed S to 5 because it provides a similarity value close to 0 when the cosine measure is minimum, that is, when the cosine mesure gives a value of -1, and D to 1 because this allows us to better distinguish vectors with separation angles between 0 and $\pi/2$. This expression has the advantage of limiting the range of values α can take to $(0, 1]$, given that $-1 \leq d(i,j) \leq 1$, as is the case for the cosine metric.

3. GROWING NEURAL GAS NETWORK

Introduced by Fritzke [7] as an approach to overcome some of the limitations of conventional self-organizing maps; namely, the *a priori* fixed number of neurons and the problem of "dead" neurons or neurons that do not update their weight vectors due to a misplacing in the input space. Formally, a growing neural gas network consists of

- a set A of units (or nodes). Each unit $c \in A$ has an associated *reference vector* $w_c \in \mathbf{R}^n$. The reference vectors can be regarded as positions in input space of the corresponding units.

- a set N of connections (or edges) among pairs of units. These connections are not weighted. Their sole purpose is the definition of topological structure.

The idea behind the training algorithm is to succesively add new units to an initially small network by evaluating local statistical measures gathered during previous adaptation steps. The network topology is generated incrementally by using a competitive Hebbian learning rule and has a dimensionality that depends on the input data and can vary locally.

The training algorithm begins with two randomly located units and an input signal that is to be learned. The learning rule used to adapt the reference vectors of the unit that is closest in the input space to the current input signal and its topological neighbors is

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \epsilon(\xi - \mathbf{w}^{(t)}) \tag{5}$$

where $\mathbf{w}$ is the reference vector of the adapting unit, ϵ is a constant called *learning rate*, ξ is the input signal being learned. The learning rates for the closest unit and its topological neighbors are different.

The squared error of the nearest unit in turn is accumulated, so that after λ input signals have been learned, a new unit is created half between the two neighboring units with the highest accumulated errors. The deletion of units happens whenever a unit is not topologically connected with any other unit. This occurs after a given number of input signals have not fired a unit. This process continues until a stop criterion is met. The complete training algorithm can be found in [7].

4. HYBRID MODEL

The proposed hybrid model consists of a set of agents, each of which has an embedded growing neural gas network. The agents behavior corresponds to the model described in section 2 with the main difference that in the hybrid model, agents learn the topology of their environment as they explore it.

An agent's growing neural gas network is always adapting itself to reflect the spatial distribution of data on the grid and, while doing so, the agent uses the network to guide its search for favorable dropping locations. Whenever an agent picks up a data object, it classifies it using its own neural gas network to bias its random walk to the location of the winner neuron. The winner neuron should be located near the most similar objects to the one just picked up because the network reflects the spatial distribution of objects in the environment. The main goal of this is to create clusters of higher quality.

After some stopping criterion is met, usually a predefined number of simulation cycles, each agent has a trained neural gas network. However, because individual networks may not cover the whole environment, a collective network is created by taking $\cup_{i=1}^{n} A'_i$ where $A'_i \subset A_i$ and A_i is the set of nodes in the growing neural network of the ith agent. The collective network is used in a second phase to further classify input vectors that were not used during the training phase. Each unit $c'_i \in A'_i$ is in the neighborhood of some data object. In our experiments, the neighboorhood is a square area of 3×3 locations on the grid. If more than one unit is in the data object neighborhood, the closest unit (in attribute space) is selected to be part of the collective network. Therefore, the clusters discovered by the ant-based clustering algorithm are used as the classes into which the new input patterns are going to be classified.

The complete resultant unsupervised classification algorithm is:

1. Initialization phase. (General).
   (a) Randomly scatter data objects on a toroidal square grid.
   (b) Create and randomly place agents with random headings on the grid.
      i. Initialize a growing neural gas network embedded into each agent. (See [7]).
2. Training-clustering phase. (Per agent).
   (a) Move randomly.
   (b) If there are any data objects within perception area then train the embedded network with them. (See [7]).
   (c) Continue with behavior described in section 2 until a stopping criterion is met, with the following change:
      i. If an object has just been picked up, classify it using network.
      ii. Set heading towards the location of winner neuron.
3. Classification phase. (General).
   (a) Extract clusters from the grid. The clusters become classes for classification.
   (b) From all individual networks, determine which neurons will be part of the collective network.
   (c) Classify test input vectors with the collective network.

5. EXPERIMENTAL SETUP

The proposed hybrid algorithm was tested on two classification tasks. The data sets used for this purpose were the Ionosphere and the Image segmentation databases from the UCI Machine Learning Repository [2]. The Ionosphere database consists of 351 instances with 34 continuous attributes each. Instances are classified into two classes, there are 225 positive and 126 negative examples respectively. The training sets used in our experiments were composed of 232 (66.6% of the total) instances. The experiment was repeated 30 times and in every repetition, randomly selected instances for the training and test sets were used. The Image segmentation database has 7 classes and is divided into two sets: the training and test sets. The training set is composed of 210 instances with 30 members of each class. The test set is composed of 2100 instances with 300 members of each class. With this database, the experiments were repeated 30 times too.

To eliminate the bias on similarity measures provoked by different scales within data attributes, we standardized the database. The similarity measure used in our experiments was the cosine metric[1].

For the ant model, expressions 1 and 2 were used to compute the picking and dropping probabilities respectively. The local similarity density function used was 4. Table 1 summarizes all other agent settings used in our experiments. These settings were selected because they gave acceptable results in preliminary tests.

Table 1: Agent settings

Parameter	Value
k_p	0.1
k_d	0.15
α	0.7
Neighborhood size	5×5
Number of agents	20

The embedded growing neural gas networks have a parameter set on their own, we refer the reader to [7] for a detailed explanation of the meaning and effects of all these parameters. Table 2 summarizes the parameter set for the embedded growing neural gas networks used in our experiments.

Table 2: Embedded growing neural gas networks settings

Parameter	Value
Winner neuron learning rate ϵ_b	0.8
Neighboring neurons learning rate ϵ_n	0.005
Maximum edge age	50
Growing threshold λ	150
Local error decreasing rate α	0.5
Global error decreasing rate d	0.995

A learning rate of 0.8 is normally considered too high, however, due to the fact that the ant environment is highly

[1]In preliminary tests, the cosine metric had better performance than Euclidean distance.

dynamic, a learning rate of this characteristics is needed in order to allow the network learn the real data distribution on the grid. A crucial condition for the proper operation of the hybrid algorithm.

In an effort to formally evaluate the clustering quality of ant-based algorithms, Handl et al. [9] applied four validity measures. From these four measures, we used the Rand index and the F-measure, which determine a similarity measure between the known correct classification C and the some other classification P. They are defined as follows:

F-Measure. The harmonic mean of recall and precision, also known as the *F-Measure*. Commonly associated to the information retrieval field, recall and precision are measures that give us some idea of how well a clustering algorithm is identifying the classes present in a database. In the context of classification, recall is defined as $r(i,j) = \frac{n_{ij}}{n_i}$ where n_{ij} is the number of elements of class i in cluster j and n_i is the number of elements of class i. Precision is defined as $p(i,j) = \frac{n_{ij}}{n_j}$ where n_j is the number of elements in cluster j. For a class i and a cluster j the *F-Measure* is defined by

$$F(i,j) = \frac{2\,p(i,j)\,r(i,j)}{p(i,j) + r(i,j)}$$

The overall *F-Measure* for the classification generated by the clustering algorithm is given by

$$F = \sum_i \frac{n_i}{n} \max_j \{F(i,j)\} \qquad (6)$$

where n is the size of the data set. F is limited to the interval $[0,1]$ with a value of 1 with a perfect clustering.

Rand Statistic. It is defined as

$$R = \frac{a+d}{a+b+c+d} \qquad (7)$$

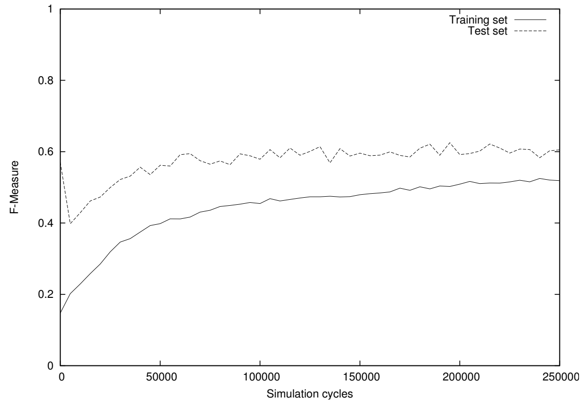
where

- a is the number of pairs where both elements belong to the same class in C and to the same group of the partition P.
- b is the number of pairs where both elements belong to the same class in C and to different groups in P.
- c is the number of pairs where both elements belong to different classes in C and to the same group in P.
- d is the number of pairs where both elements belong to different classes in C and to different groups in P.

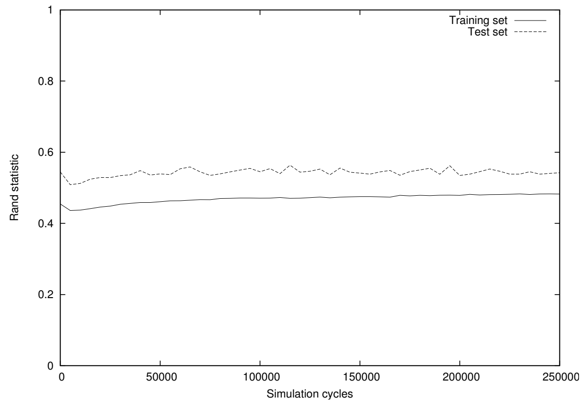
Note that $a+b+c+d = N(N-1)/2$ where N is the total size of the data set. The Rand statistic is limited to the interval $[0,1]$ with a value of 1 with a perfect clustering.

6. RESULTS

Figure 1 shows the $F-$Measure 1(a) and Rand statistic 1(b) scores over time for the algorithm during training and classification phases with the Ionosphere database.



(a) F-Measure scores



(b) Rand statistic scores

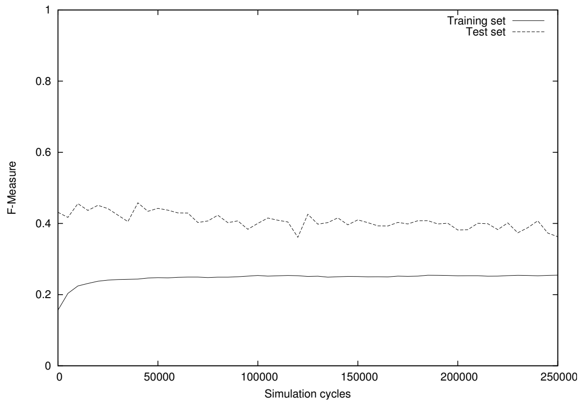
Figure 1: F-Measure and Rand statistic scores over time for the hybrid algorithm. Subfigure 1(a) shows the F-Measure behavior over time. Subfigure 1(b) shows the Rand statistic behavior over time.

Figure 2 shows the $F-$Measure 2(a) and Rand statistic 2(b) scores over time for the algorithm during training and classification phases with the Image segmentation database.

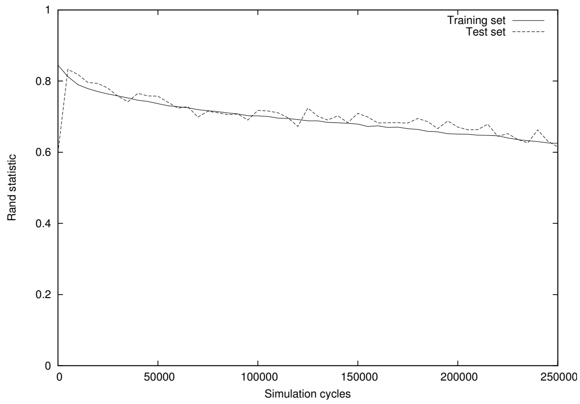
These experiments results show that the classification performance with unseen examples is strongly correlated with the clustering quality that define classes into which new input vector are classified. A result that was expected due to the fact that the ant-based clustering algorithm is the responsible of determining the classes into which the classification is done. The hybridization looks promising from the perspective of ant-based clustering because it could be useful with large databases. By using the hybrid algorithm one could classify, in an unsupervised way, elements of a large database without increasing the number of agents or the environment dimensions, given, of course, that the training set is representative enough of the whole data collection.

7. DISCUSSION

The hybridization proposed in this paper opens the door



(a) F-Measure scores



(b) Rand statistic scores

Figure 2: F-Measure and Rand statistic scores over time for the hybrid algorithm. Subfigure 2(a) shows the F-Measure behavior over time. Subfigure 2(b) shows the Rand statistic behavior over time.

to the application of ant-based algorithms to large database explorations. By clustering a sample of a database, the ant-based clustering algorithm parameters need not be changed whenever the database being explored changes. In particular, the number of agents and the environment dimensions. A collective neural gas network is used instead of a sole network, because networks are not equally trained for all regions of the input space due to the agents random walk. This collective network covers the environment and thus, the final classification considers all the classes discovered by the ant-based algorithm.

There is however, an important drawback. The multiple parameter set needed to fine tune the algorithm. This is consequence of the lack of understanding of the impact in the global behavior of a colony of simulated insect-like agents. This parameter set need to be tuned for specific needs.

8. CONCLUSIONS

We have presented an hybrid algorithm with ant-based clustering algorithms and growing neural gas networks to create an unsupervised classification algorithm. Ant-based clustering discovers natural partitions in data and neural networks learn their distribution in both, the attribute space and the plane, where ants use them to guide their search for favorable dropping locations. This symbiotic relationship overcomes some of the limitations of both techniques.

Results suggest that by using this hybrid approach, ant-based clustering techniques can be used for classification tasks over large databases.

9. REFERENCES

[1] D. Alahakoon, S. K. Halgamuge, and B. Srinivasan. Dynamic self-organizing maps with controlled growth for knowledge discovery. *IEEE Transactions on Neural Networks*, 11(3), 2000.

[2] C. Blake and C. Merz. UCI repository of machine learning databases [http://www.ics.uci.edu/∼mlearn/mlrepository.html], 1998.

[3] E. Bonabeau, M. Dorigo, and G. Theralauz. *Swarm Intelligence. From Natural to Artificial Systems.* Oxford University Press, 1999.

[4] J.-L. Deneubourg, S.Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chretien. The dynamics of collective sorting: Robot-like ants and ant-like robots. In *Proceedings of the 1st International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pages 356–365. MIT Press, 1991.

[5] M. Dittenbach, D. Merkl, and A. Rauber. The growing hierarchical self-organizing map. In *International Joint Conference on Neural Networks.* IEEE Computer Society Press, 2000.

[6] B. S. Everitt, S. Landau, and M. Leese. *Cluster Analysis.* Arnold, 4th edition, 2001.

[7] B. Fritzke. A growing neural gas network learns topologies. In *Advances in Neural Information Processing Systems 7.* MIT Press, 1995.

[8] P.-P. Grassé. La reconstruction du nid et les coordinations inter-individuelles chez bellicositermes natalensis et cubitermes sp. la theorie de la stigmergie: Essai d'interpretation des termites constructeurs. *Insectes Sociaux*, 6(1):41–83, 1959.

[9] J. Handl, J. Knowles, and M. Dorigo. On the performance of ant-based clustering. Design and application of hybrid intelligent systems. Frontiers in Artificial Intelligence and Applications 104. IOS Press, Amsterdam, The Netherlands, 2003.

[10] O. Holland and C. Melhuish. Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life*, 5:173–202, 1999.

[11] A. Jain, M. Murty, and P. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3), 1999.

[12] T. Kohonen, T. S. Huang, and M. R. Schroeder. *Self-organizing maps.* Springer Verlag, 3th edition, 2000.

[13] E. D. Lumer and B. Faieta. Diversity and adaptation in populations of clustering ants. In *Proceedings of the 3th International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3*, pages 501–508. MIT Press, 1994.

[14] E. O. Wilson. *The Insect Societies.* The Belkap Press of Harvard University Press, 1971.