Comparing an Ant-Based Clustering Algorithm with Self-Organizing Maps and K-means

Clodis Boscarioli, Rosangela Villwock and Bruno Eduardo Soares

Western Paraná State University (UNIOESTE)
Avenida Universitária, 2069, Bairro Universitário, CEP: 85.819-110, Cascavel, Paraná, Brazil

Summary
The data analysis involves the performance of different tasks, which can be performed by many different techniques and strategies. The data clustering task, an unsupervised pattern recognition process, is the task of assigning a set of objects into groups called clusters so that the objects in the same cluster are more similar to each other than to those in other clusters. This paper describes three different approaches to Data Clustering using the artificial neural network Self-Organizing Maps, K-means and an Ant-based Algorithm proposal, and the experimental results are discussed comparing their performance.

Keywords:
Ant Colony, Self-Organizing Maps, Experimental Evaluation, Data Clustering

1. Introduction

The exploratory data analysis makes use of different visualization techniques, graphical and quantitative interpretation, aiming to maximize the gathering of the information that is hidden in the structure of the data sets. Among these approaches, there is the data clustering, an unsupervised process of pattern recognition that has a key role in the exploratory data analysis, in the reunion of objects that holds some significant similarity with one another.

The techniques of data clustering, which are not corroborative, but exploratory by nature have precisely the aim to make an optimal separation of the objects of one compilation. The key issue is how to organize the observed data into structures that make sense, or even, how to develop taxonomies able to separate the observed data into different groups that occur naturally in its own data set.

The main objective of this work is to explore, in a methodological and comparative way, the application of algorithms based on Ant Colony and the artificial neural networks Self-Organizing Maps (SOM) in the analysis of numerical data clustering, so that it can be discovered the group structure inherent to the analyzed data set, in case there is one. Also, it is intended to compare these approaches with the results of the K-means algorithm. Davies-Bouldin Index (DBI) [1] was used for validation in all the experiments.

This paper is structured as follows: The Section 2 presents the main data clustering concepts, and the algorithms used for cluster recovery. In the Section 3 the methodology to experimental evaluation was introduced with results their performance compared. Lastly, Section 4 presents the some conclusions, and future directions.

2. Data Clustering

Cluster Analysis is the process of data clustering such that the objects inside a group have a high similarity when compared to the other objects of said group, and high dissimilarity to the objects of other groups. To [2], the task of data clustering is based in two basic ideas: the internal cohesion of the objects (homogeneity) and the external isolation (separation) among the groups.

According to [5], the cluster analysis is a generic denomination for a wide-scale of numerical methods utilized to examine multivariate data, aiming to find the homogenous sets of observations. Given a sample with an n amount of data (or individuals), each one of them measured by p variables, the objective is to look for a scheme that cluster them into g groups. With this cluster, it's possible to identify useful relations between the data, like similarities and differences that were not previously revealed.

It is an unsupervised learning process, since there are no predefined classes or examples which demonstrate that some sort of relation should be valid among the data, or even the presence of tutors of the field to supervise the learning process. Both the optimal number of groups as the particular characteristics that reveal similarities (or differences) should be determined by the process itself.

The cluster analysis is a difficult problem to solve because many critical factors are not included in the given problem, such as proximity measures, definition of the criteria functions, proper algorithms and its initial conditions. Furthermore, it is known that no clustering method can

Manuscript received September 5, 2012
Manuscript revised September 20, 2012

properly handle all types of group structures (with different shapes, sizes and density) [11].
The many existing approaches for cluster analysis can produce different groups for the same kind of data. And if, for the same algorithm, there are any changes in the parameters, or even changes in the order of the presentation of input patterns, the final results can be affected. Therefore, a good comparative evaluation is essential to increase the reliability on the results given by the algorithm for the account of the experts on the field. Furthermore we describe the evaluations used on this work.

2.1 Ant-Based Clustering Algorithms

Clustering based on Ants was initially suggested by [4]. In it, ants were represented as simple agents that moved randomly on a square grid. The patterns were scattered within this grid and could be picked, transported and dropped by the agents (ants). These operations are based on similarity and on the density of the patterns that were distributed in the agent's local vicinity; isolated patterns or those that are surrounded by dissimilar ones are more likely to be picked and dropped in a neighborhood of similar.
According to [20], the general idea with this algorithm is to have similar data in the original n-dimensional space in neighboring regions of the grid, this is, data that are neighbors in the grid indicate similar patterns in the original space.
Ant-based Clustering Algorithms are inspired mainly in the versions proposed by [4] and [6]. According to [1], several modifications were introduced to improve the quality of clusters and, in particular, the spatial separation between clusters in the grid.
The ant-based clustering analysis used here was based on the basic algorithm by [4] and some strategies described by [6] with modifications proposed by [18], in the following three steps:

i. Initial stage:

- The patterns are randomly scattered on the grid.
- Each ant randomly chooses a pattern to load and it's placed in a random position on the grid.

ii. Distribution stage:

- Each ant is randomly selected and it moves randomly along the grid.
- The ant probabilistically decides if it'll unload its pattern in this position. The pattern is only unload in the randomly chosen position if this probability is higher than the probability of unload this pattern in its current position.
- If the decision is negative, another ant is randomly chosen and the distribution stage starts over again.
- If the decision is positive, the ant unloads the pattern in its current position in the grid, if it's free.
- If that cell of the grid is occupied, the pattern must be unloaded in another cell in a nearby neighborhood, which must be free, by means of a random research. The evaluation of the probability of unloading the pattern in the new position is made, and the patter is only discharged in a neighbor cell if the probability of unloading the pattern in this position is still higher than the probability of unloading this pattern in its current position. If in any free nearby position the probability of unloading the pattern is higher than the probability of unloading this pattern in the current position, the pattern in not unloaded and the process starts over by the choice of another ant.
- The ant randomly searches for a new pattern to carry (among the free patterns), it goes to its position on the grid, makes the evaluation of the neighborhood function and probabilistically decides whether to carry this pattern or not.
- This choosing process of a free pattern in the grid is performed until the ant finds a pattern to carry.
- The carried pattern for an ant will be replaced in case this pattern is not unloaded in 100 consecutive interactions. Another pattern is randomly chosen, but it is only carried if the probability of carrying this pattern was higher than 0.13397. In case that there is no pattern with such a probability, the last randomly chosen pattern is carried by the ant.

The modifications proposed in [18] are described below. During the study of the Ant-based Clustering, it was observed that many of the changes in position of patterns occur unnecessarily. It is considered an unnecessary change when a pattern is among similar ones on the grid and, in this case, there is no need to change this pattern to another position. Aiming to avoid these unnecessary changes, was introduced a comparison of the probability of dropping a pattern in the position chosen randomly with the probability of dropping this pattern at its current position. The pattern is only dropped at the position chosen randomly if this probability is greater than the probability of dropping this pattern at its current position.
The occurrence of fusion of close clusters on the grid was also observed. When a decision to drop a pattern is positive and the cell where that pattern should be dropped is occupied, a free random position close to this one is searched for. However, this new position may also be close to another pattern cluster on the grid. This may be one reason for the merger of close clusters. As an alternative to prevent the merger of close clusters on the grid, in this paper was proposed an assessment of the probability for

the new position. The pattern is only dropped at the position chosen randomly if this probability is greater than the probability of dropping this pattern at its current position. All free neighboring positions are evaluated. If at no free neighboring position the probability of dropping the pattern is higher than the probability of dropping the pattern at its current location, the pattern is not dropped and the process starts again by choosing another ant.

Another issue observed in the Ant-based Clustering is that an ant can carry a pattern that is among similar ones on the grid. An ant only carries a pattern when it is not among similar ones on the grid. However, since the ant carries a pattern until it is drawn to attempt to drop the pattern, changes occur in this neighborhood and then can it leave it among the similar ones. Therefore, this ant is inactive because the operation of dropping the pattern is not performed. In this case, it was proposed to replace the pattern picked by an ant, if this pattern is not dropped in 100 consecutive iterations. The new pattern was chosen by lot, but it was only picked by the ant if the probability of carrying this pattern is greater than 0.13397. The value 0.13397 was defined by making the pick probability (ppick) equal to the drop probability (pdrop). If there is no pattern with a picking probability higher than 0.13397, the ant picks the last pattern drawn. This could also be a stopping criterion.

iii. Clustering Stage:

– The process starts when each pattern forms one group.

After calculating the distance between every group, must fuse (bind) the two groups with the shortest distance (these distances between the groups are defined by the distance of the grid).

2.2 SOM-based Clustering Algorithms

SOM is based in a map of neurons, whose weighs are adapted to the similar input vectors present in a training set [10]. During the training, the SOM behaves as a flexible network, which folds inside clouds formed by the data vectors involved in the training. Due the neighbor relation, neighbor neurons are dragged in the same direction, indicating that the codebook vectors of the neighbor neurons become similar during the learning process. For each neuron is registered its proper value, since a proximity function to the input data.

A SOM consists of M neurons located on a regular low dimensional grid, usually in two dimensions (2-D). The lattice of the 2-D grid is either hexagonal or rectangular. Each input vector $x(n)$ is defined as a real vector $x(n)=\{\delta_1, \delta_2, \ldots, \delta_d\}^T \varepsilon R^d$. The SOM algorithm is iterative. Each neuron (or unit) i has a d-dimensional weight vector (or codebook), as it is also called, $w_i = [w_{i1}, \cdots, w_{id}]^T \in R^d$. Initially, in $t = 0$, the weight vector is initialized randomly preferably from the domain of the input vectors [10]. At each training step t, a sample data vector $x(n)$ is randomly chosen from a training set (N). General distances between $x(n)$ and all weight vector w_i are computed. The winning neuron, denoted by c, is the neuron with the weight vector closest to $x(n)$ or the best match unit (BMU), as Equation 1.

$$c = \arg\min_i \{d(x(n), w_i)\}, \qquad i \in \{1, \ldots, M\} \tag{1}$$

A set of neighboring neurons of the winning neuron is denoted as N_c, which decreases its neighboring radius of the wining neuron at each training step. The $h_{ic}(t)$ is defined as the neighborhood kernel function around the winning neuron c at time t. The neighborhood kernel function is a non-increasing function of time t and of the distance of neuron i from the wining neuron c in the 2-D output space. The kernel can be taken as a Gaussian function [10].

SOM does the reduction of the data set to codebook vectors which will be used by other techniques, whether are they visualization or data clustering. Besides the reduction of the data set for analysis, another advantage of SOM is that it is not necessary to recalculate the map for each new input data, therefore, if the statistics can be assumed as stationary, new data can directly be mapped to the codebook vector representative of the item of the nearest data to the old model.

It must be made very clear that, with exception of approaches that on SOM is applied a segmentation algorithm, the exit of a SOM map supplies only the representation of the data through topological organization of the neurons. This result is not always passive of a direct visual analysis, being, therefore, difficult to suggest the structure of existing groups since this mapping topologically ordered of SOM not to be enough to carry through an analysis of groupings, what refers to the application of other techniques on the SOM so that the results are, of some form, observable.

To discovery information from clustering, a post-processing technique is applied on the BMU's neurons. For such, several approaches exist, being distinguished the visual representations, and the application of clustering techniques on the neurons.

One of the SOM objectives is to represent input patterns of high dimensionality with codebook vectors, so that they can be visualized, of a facilitated form, in a map of lower dimension, generally 2-D, once that the limited number of visual dimensions is one of the problems of the visualization of multidimensional data. For cases where previously definite labels exist, it is feasible the interpretation of existence of groups. However, without the existence of these labels, or exactly when the groups are not linearly separable, it is not possible to affirm that

neurons, although neighbors represent data that are contained in a same cluster, and not even the amount of groups present.

The SOM algorithm has been, for some years, used as base for the development of some algorithms for data clustering, as in [7]. In [12], a SOM in two layers was considered, to group data. The second layer of the SOM takes as entrance the codebook topologically ordered in the first layer. Some works that specifically occupy themselves with the task of clustering data through SOM without previous knowledge of the number of the groups desired, and considered correlated to this research, they are [16], [21] and [15]. The methodology of [16] is sufficiently known and cited in literature, reason for which it will be used here. This approach has two phases: first it uses SOM and after, applies K-means on the codebook vectors to cluster the data.

2.3 K-means Algorithm

K-means is one of the most popular clustering algorithms [8], based in partitioning, so that giving a database with n objects and k groups to form, it organizes the objects in k parts ($k \leq n$), where each part represents a group.

It is based on centroids, using the geometric center of each group to represent it, and the groups are formed so as to optimize a criterion of objective partitioning, commonly called as the similarity function, as a distance measurement. Thus, the objects allocated within a group are similar and the objects allocated within a different group as dissimilar, regarding the attributes that make up the database. The algorithm can be described as follows:

```
Input: Data set and a value for k.
Output: Clustered data set.
Select k points, randomly, as initial
centroids.
REPEAT
  Assign each point to the centroid closest
to it;
  Recalculate the centroid for each group;
UNTIL {Stabilize}
```

3. Experimental Evaluation

The algorithms were implemented in MatLab [13] and the following database of the UCI Machine Learning Repository [14] were used: Iris Plants, Wisconsin Breast Cancer, Dermatology, Pima Indians Diabetes and Abalone Databases, it was assumed that the number of groups to be formed is the number of existing classes in each one of them: Iris Plants, 3 groups; Wisconsin Breast Cancer, 2 groups; Dermatology, 6 groups; Pima Indians Diabetes, 2 groups; and Abalone, 29 groups. Both the K-means and the other two algorithms were performed 10 times to each database, and the best clustering (lowest DB-index) was chosen.

For the clustering made through the ant-based algorithms the following parameter was used: in the definition of the grid size, it was chosen the number of cells equal to 10 times the number of patterns and 10 ants were used (p=10). It was used square neighborhood and the radius of the initial neighborhood was set equal to 1, value which was incremented during the initial stage. During the final stage, the value decreases in order to "relax" the neighborhood size when the ants aren't able to unload the patterns they're carrying. The value of the neighborhood radius was only considered as equal to 1. After some preliminary tests the initial parameter α ($\alpha 0$) was set to 0.8. The dissimilarity measure used was the Euclidian distance. The distance matrix was calculated and standardized. In the recovery of the groups the Ward's method was used. More implementation details are described in [18].

For the clustering made through SOM, the SOM Toolbox was used [17]. First, a large number of prototypes was formed by the SOM, which was combined to form clusters. Neurons that do not represent any element of the data set were excluded. In the second phase, the weight vectors of the network were used to discover clusters. The following parameterization was adopted: two-dimensional and rectangular map, linear initialization, Euclidean neighborhood function, number of neurons $5\sqrt{n}$, where n is the length of the dataset and batch training algorithm, using the default parameters of the SOM-toolbox. The K-means algorithm requires the number of the groups to be formed before your execution, such that the actual number of classes of each database was used there. Given the sensitivity to initialize the K-means algorithm, it was run 100 times to each value of k, with random initialization. The best partition was selected by an error criterion with respect to the centroid.

For the validation, the Davies-Bouldin index was used in every execution, for all the methodologies. This index indicates the similarity among the groups and can be used in the validation of the data partitions and in the relative comparison between different segmentations of a data set. Lower values of DBI indicate the better clustering result with more compacted groups, and higher values of DBI indicate more dispersed groups.

The Table 1 shows the comparison results. The results of the Ant Colony were satisfactory for the smaller databases, but not for the others, while the results of the other two algorithms remained more homogenous and closer together. For the Pima Database the results were not so good, which also happened for the K-means too. For the Abalone

Database, the Ant Colony results were worst those other algorithms, but this database has some singletons (groups with only one data item).

Table 1: Comparison the Experimental results between three methodologies

Databases/ Average DBI	Iris	Wisconsin	Derma-tology	Pima	Abalone
Ant Colony	0.787	1.864	3.802	4.600	29.599
SOM	0.749	0.695	0.839	0.723	0.887
K-means	0.846	1.444	1.502	2.174	1.034

Although, the ant-based algorithm proposed has not presented superiority in our experiments, it has presented improvements regarding the other approach involving clustering by ant colony, described by [1]. Here we compare that algorithm improved with 2-dimensional SOM and K-means. In [19] was presented a comparison of Ant-based clustering algorithm with *Ward's method*, One-dimensional Self-Organizing Maps, and the Modified Ant-based clustering algorithm proposed in [1].

For these experiments, the SOM-based algorithm had the best clustering results considering the DBI, an internal index validation. The vector quantization SOM preserves the topology of the input data but reduce its size as well as the organization of prototype vectors facilitates the data clustering task.

4. Conclusion

An empirical comparsion between an ant-based algorithm, 2D-SOM and K-means was presented. The SOM neural network was chosen because it has been widely used for the tasks of clustering and topological mapping, simultaneously, and K-means was chosen for it is a classical approach of data clustering.

The Ant-based clustering algorithm is a relatively new metaheuristic and it has been receiving special attention, mostly because it still requires a lot of investigation to improve its performance, its stability and other characteristics necessaries to transform this algorithm into a mature tool for data mining. And also, for being able to automatically discover the number of groups in the data.

K-means only works properly for spherical groups, with results strongly dependent of the initial centroids. It is sensitive to outliers and it requires prior knowledge of the number of existing groups. SOM and Ants, according to [6], are not limited to the discovery of homogenous groups in the data, but they capture neighborly relations in a two-dimensional visualization of an area of high dimensional data. However, no algorithm dominates the others in every situation, and it's expected that no strategy has a better performance than another strategy when tested on a large set of databases with different characteristics.

We continue the research optimizing the Ant-based clustering algorithm there discussed, investigating other clustering methodologies, and with other evaluations.

References

[1] BORYCZKA, U. Finding clusters in data: Cluster analysis with ants. Applied Soft Computing, v. 9, p. 61-70 (2009).

[2] CORMACK, R. M. A Review of Classifications. Journal of the Royal Statistical Society. v. Series A, n. 134, p. 321-353 (1971).

[3] DAVIES, D. L.; BOULDIN, D. W. A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence. v. PAMI-1, n. 2, p. 224-227 (April, 1979).

[4] DENEUBOURG, J. L.; GOSS, S.; FRANKS, N.; SENDOVA-FRANKS, A.; DETRAIN, C.; CHRÉTIEN, L. The dynamics of collective sorting: Robot-like ants and ant-like robots. In Proceedings of the First international Conference on Simulation of Adaptive Behavior: From Animals to Animals 1, p. 356–365. Cambridge (1991), MA: MIT Press.

[5] EVERITT, B. S.; LANDAU, S.; MORVEN, L. Cluster Analysis. Hodder Arnold Publishers: 4th Ed, London (2001).

[6] HANDL, J.; KNOWLES, J.; DORIGO, M. Ant-Based Clustering and Topographic Mapping. Artificial Life, v. 12, n. 1, p. 35-61 (2006).

[7] HUNTSBERGER, T.; AJJIMARANGSEE, P. Parallel Self-organizing Feature Maps for Unsupervised Pattern Recognition. Int. Journal General Systems. v. 16, p. 357-372 (1989).

[8] JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data Clustering: A Review. ACM Computing Surveys. v. 31, n. 3, p. 264-323 (September, 1999).

[9] JOHNSON, R.A.; WICHERN, D.W. Applied Multivariate Statistical Analysis. Fourth Edition. New Jersey: Prentice Hall (1998).

[10] KOHONEN, T. Self-Organizing Maps. 3rd Edition. Springer-Verlag. Nova York, USA (2001).

[11] KOTSIANTIS, S. B.; PINTELAS, P. E. Recent Advances in Clustering: A Brief Survey. Transactions on Information Science and Applications, v. 1, n. 1, p. 73-81 (2004).

[12] LAMPINEN, J.; OJA, E. Clustering Properties of Hierarchical Self-Organizing Maps. Journal of Mathematical Imaging and Vision. n. 2-3, v. 2, p. 261-272 (1992).

[13] MATLAB R2010 – The MatWorks, MATLAB. The MathWorks Inc., Natick (2010).

[14] NEWMAN, D. J.; HETTICH, S.; BLAKE, C. L.; MERZ, C. J. UCI – Repository of Machine Learning Databases. University of California, Irvine, Dept. of Information and Computer Sciences, (1998), http://archive.ics.uci.edu/ml/.

[15] TASDEMIR, K.; MERÉNYI, E. Considering Topology in the Clustering of Self-Organizing Maps. Proceedings of the 5th Workshop on Self-Organizing Maps (WSOM'2005), p. 439-446, Paris, France (September, 2005).

[16] VESANTO, J.; ALHONIEMI, E. Clustering of the Self-organizing Map. IEEE Transactions on Neural Networks. n. 3, v. 11, p. 586-600 (May, 2000).

[17] VESANTO, J.; HIMBERG J. ALHONIEMI, E.; PARHANKANGAS, J. Self-organizing map in matlab: The som toolbox. *Proceedings of the Matlab DSP Conference 1999*, Digital Signal and Image Processing – DSP Group, p. 35-40, Espoo, Finlândia (November 1999), http://citeseer.ist.psu.edu/vesanto00selforganizing.html.
[18] VILLWOCK, R. Clustering Techniques and Hierarchy in the KDD Context – Application to Temporal Data-Structural Geotechnical Instrumentation of Itaipu Hydroelectric Power Plant. 125 p. PhD. Thesis in Numerical Methods in Engineering – Exact Science Sector, Parana Federal University, Curitiba (2009). (In Portuguese).
[19] VILLWOCK, R.; STEINER, M. T. A.; SIQUEIRA, P. H. Pattern Clustering via Ants Colony, Ward Method and Kohonen Maps. IJCSNS International Journal of Computer Science and Network Security, v. 12, n. 6, p. 81-91 (June, 2012).
[20] VIZINE, A. L.; DE CASTRO, L. N.; HRUSCHKA, E. R.; GUDWIN, R. R. Towards improving clustering ants: an adaptive ant clustering algorithm. Informatica, v. 29, p. 143-154 (2005).
[21] WU, S.; CHOW, T. W. S. Self-Organizing-Map Based Clustering Using a Local Clustering Validity Index. Neural Processing Letters, Kluwer Academic Publishers, v. 17, p. 253-271, Netherlands (2003).

Bruno Eduardo Soares completed his undergraduate studies in Computer Science at Western Paraná State University, at Cascavel city, State of Paraná, in Brazil, in 2012. He is programmer at the Verifik System Company. His recent interest including Database Systems and Data Mining, more specifically, Cluster Analysis.

Clodis Boscarioli completed his undergraduate studies in Informatics at Ponta Grossa State University, at Ponta Grossa city, State of Paraná, in Brazil in 1996. He got his Master's degree in Informatics in 2002 at Federal University of Paraná, at Curitiba city, State of Paraná, in Brazil, in 2002, and his Ph.D.'degree in Electrical Engineering at São Paulo University at São Paulo city, State of São Paulo, in Brazil, in 2008. He is an Adjunct Professor at the Western Paraná State University, since 2000. His recent interest includes Database Systems and Data Mining, Human Computer Interaction, and Business Intelligence.

Rosangela Villwock completed her undergraduate studies in Mathematics at Western Paraná State University, at Cascavel city, State of Paraná, in Brazil, in 1997. She got her Master's degree in this same university in 2003 and her Ph.D.' degree in Numerical Methods in Engineering at Federal University of Paraná, at Curitiba city, State of Paraná, in Brazil, in 2009. She is an Adjunct Professor at the Western Paraná State University, since 1999. Her recent interest includes Data Mining and Metaheuristics, in special, Ant Colonies.