Adaptive batch SOM for multiple dissimilarity data tables

Anderson B. dos S. Dantas*, Francisco de A. T. de Carvalho*
**Center of Informatics – Federal University of Pernambuco (CIn/UFPE)*
Av. Prof. Luiz Freire, s/n - Cidade Universitaria, CEP 50740-540, Recife – PE – Brazil
Email: {absd,fatc}@cin.ufpe.br

***Abstract*—This paper introduces a clustering algorithm based on batch Self-Organizing Maps to partition objects taking into account their relational descriptions given by multiple dissimilarity matrices. The presented approach provides a partition of the objects and a prototype for each cluster, moreover the method is able to learn relevance weights for each dissimilarity matrix by optimizing an adequacy criterion that measures the fit between clusters and the respective prototypes. These relevance weights change at each iteration and are different from one cluster to another.**

***Keywords*-Clustering; Self-Organizing Maps; Relational data; Relevance weight; Multiple dissimilarity matrices**

I. Introduction

Kohonen's SOM (Self-Organizing Maps) are a special type of unsupervised neural network which has clustering and visualization properties [1]. SOM can be considered as an algorithm that maps a high dimensional data space in a one, two or three-dimensional space. This projection enables partitioning the inputs into similar groups and has the property to preserve the topology of the data.

There are two common representations of the objects upon which clustering can be based: feature data and relational data. When each pair of objects is represented by a relationship, then we have relational data. The most common model of relational data is the case when we have a matrix of dissimilarity data $R = [r_{il}]$, where r_{il} is the pairwise dissimilarity (often a distance) between objects i and l.

References [2] and [3] propose an adaptation of a batch SOM algorithm to dissimilarity data. Reference [4] presents a clustering algorithm that is able to partition objects taking into account simultaneously, their relational description given by multiple dissimilarity matrices.

This work proposes an approach based on batch SOM algorithm with adaptive weights to classify objects using multiple dissimilarity matrices. These matrices can be obtained using different sets of variables and a fixed dissimilarity function, using a fixed set of variables and different dissimilarity functions or using different sets of variables and dissimilarity functions. The influence of the different dissimilarity matrices is not equally important in the definition of the clusters in the final consensus partition [5]. Thus, in order to obtain a meaningful partition from all dissimilarity matrices, it is necessary to learn cluster-dependent relevance weights for each dissimilarity matrix.

This paper is organized as follows. Section II describes a batch Self-Organizing Map algorithm for dissimilarity data proposed by [2]. Section III presents the algorithm proposed by this paper, the batch Self-Organizing Map for multiple dissimilarity data tables. To show the usefulness of this clustering algorithm, experiments with real-value data sets are considered in section IV. Finally, Section V gives the conclusions.

II. Batch Self-Organizing Map for dissimilarity data

The batch SOM algorithm for dissimilarity data (B-SOM for dissimilarity data) introduced by [2] stems from the original Kohonen's SOM [6]. The main difference is on the data which are represented by a dissimilarity relationship. Let $E = \{e_1, \ldots, e_n\}$ be a set of n objects and a dissimilarity measure $d(e_i, e_l)$ between the objects e_i and e_l. Each neuron c is represented by a reference vector (or prototype) $g_c = e_j,\ e_j \in E$. In classical SOM each referent vector evolves in the entire input space $\mathbb{R}^p$, in this approach each neuron has a finite number of representations.

The batch training algorithm is an iterative two-step algorithm (affectation and representation steps, discussed later) in which the whole set of input (noted E) is presented to the map before any adjustments be achieved.

The concept of neighbourhood is taken into account through kernel functions K which are positive and such that $\lim_{|x|\to\infty} K(x) = 0$ [6]. Moreover, K^T that is parameterized by T, is the neighbourhood kernel function that defines the influence region around each neuron r. The smaller the value of T, the fewer the neurons that belong to the neighbourhood of a given neuron r.

When T is kept fixed, the algorithm is performed iteratively in two steps: an affectation step, in which each element e_i is associated to the "closest" neuron representing a referent vector followed by a representation step, in which new prototypes representing each cluster are selected.

III. Adaptive batch Self-Organizing Map for multiple dissimilarity data tables

In this section we present an adaptive clustering algorithm based on both SOM for dissimilarity data [2] and the clustering algorithm for multiple dissimilarity data tables [4]. The main objective of the model proposed is to map objects taking into account its relational descriptions given by multiple dissimilarity data tables.

This approach is an adaptive batch SOM algorithm (AB-SOM) that uses different adaptive weights on the dissimilarity matrices. Those weights change at each algorithm iteration and are different from one cluster to another and aims to measure the relevance of each matrix in the process of clustering. Previous works [7] [8] give adaptive versions of stochastic SOM algorithm. The computation of these vectors of weights in this algorithm is inspired from the approach used to compute a weight for each variable in each cluster in the dynamic clustering algorithm based on adaptive distances [9].

Let $E = \{e_1, \ldots, e_n\}$ be a set of n objects and let p dissimilarity nxn matrices $(\mathbf{D}_1, \ldots, \mathbf{D}_j, \ldots, \mathbf{D}_p)$ where $\mathbf{D}_j[i, l] = d_j(e_i, e_l)$, and $d_j(e_i, e_l)$ gives the dissimilarity between the objects e_i and e_l on the dissimilarity matrix $\mathbf{D}_j$. Assume that the prototype g_r of the cluster C_r is an object of the set E, i.e., $g_r \in E\, \forall r = 1, \ldots, c$.

The adaptive batch SOM training algorithm for multiple dissimilarity data tables introduced in this paper is an iterative three-step algorithm (affectation, representation and weighting steps) which aims to minimize the following cost function:

$$J = \sum_{i=1}^{n} \sum_{r=1}^{c} K^T(\delta(f^T(e_i), r)) D_{\lambda_r}(e_i, g_r) \qquad (1)$$

where the kernel function K is described by: $K^T(\delta) = \exp(-\frac{\delta^2}{T^2})$ and $d_j(e_i, g_r)$ is the dissimilarity between an example $e_i \in C_r$ and the cluster prototype $g_r \in E$ parameterized by relevance weight vector $\lambda_r = (\lambda_{r1}, \ldots, \lambda_{rj}, \ldots, \lambda_{rp})$ where λ_{rj} is the weight between the dissimilarity matrix $\mathbf{D}_j$ and the cluster C_r.

The relevance weight matrix λ is composed by c relevance weight vectors $\lambda_r = (\lambda_{r1}, \ldots, \lambda_{rj}, \ldots, \lambda_{rp})(k = 1, \ldots, c)$. The weight matrix changes at each iteration of the algorithm and the weight vectors are different for one cluster to another. When T is kept fixed, the algorithm alternates three steps:

Step 1: Definition of the Best Prototypes

In this step, the partition $P = (C_1, \ldots, C_c)$ of E into c clusters and the relevance weight matrix λ are fixed. The prototype $g_r \in E$ of cluster C_r, which minimizes the clustering criterion J, is computed according to:

$$g_r = arg \min_{e \in E} \sum_{i=1}^{n} K^T(\delta(f^T(e_i), r)) \sum_{j=1}^{p} \lambda_{rj} d_j(e_i, e) \qquad (2)$$

Step 2: Definition of the Best Relevance Weight Matrix

In this step, the partition $P = (C_1, \ldots, C_c)$ of E and the vector of prototypes $g = (g_1, \ldots, g_c)$ are fixed. The element j of the relevance weight vector $\lambda_r = (\lambda_r 1, \ldots, \lambda_r p)$, which minimizes the clustering criterion J under $\lambda_r j > 0$ and $\prod_{j=1}^{p} \lambda_r j = 1$, is calculated according to the equation:

$$\lambda_{rj} = \frac{\{\prod_{h=1}^{p} (\sum_{i=1}^{n} K^T \delta^2(f^T(e_i), r) d_h(e_i, g_r))\}^{\frac{1}{p}}}{\sum_{i=1}^{n} K^T \delta^2(f^T(e_i), r) d_j(e_i, g_r)} \qquad (3)$$

Step 3: Definition of the Best Partition

In this step, the vector of prototypes $g = (g_1, \ldots, g_c)$ and the relevance weight matrix λ are fixed. The cluster C_r, which minimize the criterion J, is updated according the following expression:

$$f^T(e_i) = arg \min_{1 \le h \le c} \sum_{r=1}^{c} K^T(\delta(h, r)) \sum_{j=1}^{p} \lambda_{rj} d_j(e_i, g_r) \qquad (4)$$

A. The AB-SOM algorithm with Relevance Weight Matrix

1) Initialization.
   Fix: the number c of clusters, δ, the number of iterations N_{iter}, T_{min}, T_{max}; Set $T \leftarrow T_{max}$; Set $t \leftarrow 0$;
   Randomly select c distinct objects $g_l \in E$;
   Set the Relevance Weight Matrix λ where $\lambda_l = (\lambda_{l1}, \ldots, \lambda_{lp}) = (1, \ldots, 1)$;
   Set the map $M(c, \mathbf{G}^0)$, where $\mathbf{G}^0 = (g_1^{(0)}, \ldots, g_c^{(0)})$;
   Assign each object e_i the the closest prototype in order to obtain the partition $P = (C_1, \ldots, C_c)$ where C_l is constructed according to the equation 4;

2) Step 1: definition of the best prototypes.
   The partition P and the relevance weight matrix λ are fixed;
   Set $T = T_{max} * (\frac{T_{min}}{T_{max}})^{\frac{t}{N_{iter}-1}}$;
   Compute the prototype $g_l \in E$ of cluster C_l according to equation 2.

3) Step 2: definition of the best relevance weight matrix.
   The vector of prototypes g and the partition $P = (C_1, \ldots, C_c)$ are fixed;
   For each l, compute the component of the weight vector λ_l according to equation 3.

4) Step 3: definition of the best partition.
   The vector of prototypes g and the relevance weight matrix λ are fixed;
   Assign each individual e_i to its nearest neuron according to equation 4;

5) Stopping criterion If $T = T_{min}$ then STOP; otherwise set $t = t + 1$ and go to 2 (Step 1).

IV. EXPERIMENTS

To illustrate the usefulness of the proposed algorithm, we used databases from the UCI Machine Learning Repository.

Each attribute in the dataset is represented by a dissimilarity matrix. For the batch Self-Organizing Map for dissimilarity data algorithm, there is only one matrix available. Some measures will be considered in order to compare the results furnished by the clustering methods, they are: the corrected Rand index (CR) [10], the F-measure [11] and the overall error rate of classification (OERC) [12]. In addition, the confusion matrix will be presented for each experiment.

The variable parameters for the experiments are: topology of the map, the values of T_{min} and T_{max} and the total number of iterations N_{iter}. The topology of the map defines the maximum number of clusters that will be furnished by the algorithm. The values used in the experiments are the following. Topology: 2x5; T_{min}: 0.3; T_{max}: 3.0; N_{iter}: 500. Each experiment was repeated 50 times with the same values and was selected the one with the minimum adequacy criterion.

A. *Wine data set*

This data set has results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars, providing three classes of wine. The classes 1, 2 and 3 have, respectively 59, 71 and 48 examples. Each example is described by 13 real value attributes.

Table I
WINE: CONFUSION MATRIX FURNISHED BY THE B-SOM ALGORITHM

Cluster/Class	1	2	3	Majority Class
0,0	0	20	10	2
0,1	1	11	16	3
0,2	7	3	5	1
0,3	10	0	0	1
0,4	6	0	0	1
1,0	0	25	2	2
1,1	0	6	9	3
1,2	5	5	6	3
1,3	16	1	0	1
1,4	14	0	0	1

Table II
WINE: CONFUSION MATRIX FURNISHED BY THE AB-SOM FOR MULTIPLE DISSIMILARITY DATA TABLES

Cluster/Class	1	2	3	Majority Class
0,0	0	5	16	3
0,1	0	1	4	3
0,2	0	21	0	2
0,3	2	22	0	2
0,4	6	10	0	2
1,0	0	0	15	3
1,1	0	0	13	3
1,2	0	9	0	2
1,3	24	2	0	1
1,4	27	1	0	1

Tables I and II give the confusion matrices furnished, respectively by the batch SOM algorithm and the adaptive approach for multiple dissimilarity tables. The last column (Majority Class) represents the number of the class with most objects in a given cluster. Observing this column it is possible to recognize regions in which objects belonging to the same a-priori class are in neighbors clusters. For example, in table II, the clusters (0,0), (0,1), (1,0) and (1,1), on the left side of the map, have more objects from class 3.

The results for CR index are 0.42 and 0.31 for, respectively, the adaptive approach for multiple dissimilarity tables and the batch SOM. For the F-measure the results are 0.52 and 0.45, respectively. And for the OERC, the results are 0.09 and 0.27, respectively. The adaptive SOM for multiple dissimilarity table performs better for the wine data set.

Table III gives the final relevance weight matrix furnished by the adaptive SOM algorithm for multiple dissimilarity data tables. Values above 1 (bold in table) denote a strong influence of a given dissimilarity matrix on the cluster. For example, for the cluster (0,0) the most relevant matrix is the matrix 7 (3.05).

B. *Iris data set*

This data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. Each example is described by 4 attributes.

Table IV
IRIS: CONFUSION MATRIX FROM B-SOM ALGORITHM

Cluster/Class	1	2	3	Majority Class
0,0	0	1	15	3
0,1	0	0	11	3
0,2	0	0	9	3
0,3	5	0	0	1
0,4	32	0	0	1
1,0	0	26	1	2
1,1	0	7	7	2/3
1,2	0	0	7	3
1,3	0	16	0	2
1,4	13	0	0	1

Table V
IRIS: CONFUSION MATRIX FROM AB-SOM FOR MULTIPLE DISSIMILARITY DATA TABLES

Cluster/Class	1	2	3	Majority Class
0,0	13	0	0	1
0,1	0	0	9	3
0,2	0	17	0	2
0,3	0	17	1	2
0,4	0	4	14	3
1,0	37	0	0	1
1,1	0	2	0	2
1,2	0	9	0	2
1,3	0	1	6	3
1,4	0	0	20	3

Tables IV and V give the confusion matrices furnished, respectively by the batch SOM algorithm and the adaptive approach for multiple dissimilarity tables. In table IV, the bottom of the map has most objects from class 3 (clusters

Table III
WINE: FINAL RELEVANCE WEIGHT MATRIX

Cluster/Matrix	1	2	3	4	5	6	7	8	9	10	11	12	13
0,0	**1.03**	0.27	**1.31**	**1.07**	0.48	0.90	**3.05**	0.33	**1.87**	**1.25**	0.57	**2.07**	**2.02**
0,1	0.99	**1.05**	0.55	**1.09**	0.27	0.69	0.79	0.52	0.41	0.92	**1.81**	**3.60**	**8.64**
0,2	**1.28**	0.52	0.41	0.43	**1.47**	0.60	**2.42**	0.90	**1.57**	**5.17**	0.18	0.51	**5.66**
0,3	0.69	0.33	0.93	**2.64**	**1.32**	0.48	**2.29**	**1.22**	**0.45**	**2.39**	0.36	**4.02**	0.62
0,4	**0.77**	**4.19**	0.27	0.99	0.16	**1.92**	0.84	**1.96**	0.24	**3.04**	**3.72**	**2.15**	0.37
1,0	**1.02**	0.37	0.63	0.89	0.47	**1.68**	**2.42**	0.25	**2.79**	0.45	**1.43**	5.12	0.99
1,1	0.46	0.21	0.79	0.90	**1.85**	0.49	**5.07**	0.86	0.24	0.36	**2.21**	**5.23**	**3.48**
1,2	0.42	0.61	0.51	**1.35**	0.16	**2.93**	**3.12**	0.24	**2.27**	**2.96**	**1.33**	**1.72**	**1.01**
1,3	0.77	0.55	0.69	0.16	0.29	**1.38**	**3.13**	**2.14**	**1.12**	**1.95**	**1.01**	**3.90**	0.86
1,4	0.48	**6.01**	0.37	0.33	0.61	**1.64**	**1.65**	0.94	0.40	**1.50**	**1.74**	**2.78**	0.60

(1,0), (1,1), (1,2) and (1,3)) and in table V, class 1 is mostly present on the left side of the map (clusters (0,0) and (1,0)).

Table VI
IRIS: FINAL RELEVANCE WEIGHT MATRIX

Cluster/Matrix	1	2	3	4
0,0	0.63	0.18	**3.40**	**2.62**
0,1	0.80	0.81	**1.45**	**1.04**
0,2	0.34	0.26	**2.95**	**3.79**
0,3	0.14	0.29	**10.14**	**2.26**
0,4	0.58	0.32	**4.99**	**1.05**
1,0	0.25	0.07	**5.06**	**11.38**
1,1	**1.59**	0.52	0.95	**1.26**
1,2	0.35	0.50	**1.50**	**3.73**
1,3	0.25	**1.53**	**1.47**	**1.76**
1,4	0.51	0.26	**2.59**	**2.83**

The results for CR index are 0.54 and 0.50 for, respectively, the adaptive approach for multiple dissimilarity tables and the batch SOM. For the F-measure the results are 0.64 and 0.62, respectively. And for the OERC, the results are 0.04 and 0.06, respectively. The adaptive SOM for multiple dissimilarity table performs better for this data set.

Table VI gives the final relevance weight matrix. In this table, the weights for the matrices 3 and 4 have greater values then the other matrices, denoting a greater influence on the objects affectation to the clusters.

V. CONCLUSION

This paper introduced a clustering algorithm based on Self-Organizing Maps that is able to partition objects taking into account their relational descriptions given by multiple dissimilarity matrices.

This algorithm is an adaptive batch SOM approach for multiple dissimilarity tables. The algorithm starts from an initial partition and alternates three steps until a stop criterion is reached. These iterative steps intend to optimize an adequacy criterion that measures the fit between the partition and their prototypes. The relevance weights change at each iteration and are different from one cluster to another.

The usefulness of this algorithm was illustrated by an experimental evaluation using real-value data sets extracted from the UCI Machine Learning Repository.

ACKNOWLEDGMENT

The authors thank the Brazilian agencies CNPq and FACEPE for their financial support.

REFERENCES

[1] K. Gopalakrishnan, S. Khaitan, and A. Manik, "Enhanced clustering analysis and visualization using kohonen's self-organizing feature map networks," *International Journal of Computational Intelligence*, vol. 4, pp. 64–71, 2008.

[2] A. Golli, B. Conan-Guez, and F. Rossi, "A self-organizing map for dissimilarity data," in *Classification, Clustering, and Data Mining Applications*. Springer Berlin Heidelberg, 2004, pp. 61–68.

[3] B. Conan-Guez, F. Rossi, and A. E. Golli, "Fast algorithm and implementation of dissimilarity self-organizing maps," *Neural Networks*, vol. 19, pp. 855–863, 2006.

[4] F. de Carvalho, Y. Lechevallier, and F. de Melo, "Partitioning hard clustering algorithms based on multiple dissimilarity matrices," *Pattern Recognition*, 2011, doi: 10.1016/j.patcog.2011.05.016.

[5] H. Frigui, C. Hwang, and F. Chung-Hoon Rhee, "Clustering and aggregation of relational data with applications to image database categorization," *Pattern recognition*, vol. 40, no. 11, pp. 3053–3068, 2007.

[6] F. Badran, M. Yacoub, and S. Thiria, "Self-organizing maps and unsupervised classification," in *Neural networks: methodology and applications*. Springer-Verlag, 2005, pp. 379–442.

[7] J. A. Kangas, T. K. Kohonen, and J. T. Laaksonen, "Variants of self-organizing maps," *IEEE Transactions on Neural Networks*, 1990.

[8] N. Grozavu, Y. Bennani, and M. Lebbah, "From variable weighting to cluster characterization in topographic unsupervised learning," in *Proceedings of International Joint Conference on Neural Networks (IJCNN'09).*

[9] E. Diday and G. Govaert, "Classification automatique avec distances adaptatives," *R.A.I.R.O. Informatique Comput. Sci.*, vol. 11, no. 4, pp. 329–349, 1977.

[10] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, pp. 193–218, 1985.

[11] C. J. van Rijsbergen, *Information Retrieval.* London: Butterworths, 1979.

[12] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen, *Classification and Regression Trees.* Chapman and Hall/CRC, 1984.