A New Self-Organizing Map for Dissimilarity Data

Tien Ho-Phuoc
GIPSA-lab, France

Anne Guerin-Dugue
GIPSA-lab, France

INTRODUCTION

The **Self-Organizing Map** (Kohonen, 1997) is an effective and a very popular tool for data clustering and visualization. With this method, the input samples are projected into a low dimension space while preserving their topology. The samples are described by a set of features. The input space is generally a high dimensional space R^d. 2D or 3D maps are very often used for visualization in a low dimension space (2 or 3).

For many applications, usually in psychology, biology, genetic, image and signal processing, such vector description is not available; only pair-wise **dissimilarity** data is provided. For instance, applications in Text Mining or ADN exploration are very important in this field and the observations are usually described through their proximities expressed by the "Levenshtein", or "String Edit" distances (Levenshtein, 1966). The first approach consists of the transformation of a dissimilarity matrix into a true **Euclidean distance** matrix. A straightforward strategy is to use "Multidimensional Scaling" techniques (Borg & Groenen, 1997) to provide a feature space. So, the initial vector SOM algorithm can be naturally used. If this transformation involves great distortions, the initial vector model for SOM is no longer valid, and the analysis of dissimilarity data requires specific techniques (Jain & Dubes, 1988; Van Cutsem, 1994) and Dissimilarity Self Organizing Map (DSOM) is a new one.

Consequently, adaptation of the **Self-Organizing Map** (SOM) to dissimilarity data is of a growing interest. During this last decade, different propositions emerged to extend the vector SOM model to pair-wise dissimilarity data. The main motivation is to cope with large proximity databases for data mining. In this article, we present a new adaptation of the SOM algorithm which is compared with two existing ones.

BACKGROUND

Basically, there are two main approaches to the SOM extension dealing with **dissimilarity** data. The first one uses a probabilistic framework, as for example in Graepel & Obermayer (1999) where a topographic mapping of proximity is derived by simulated annealing. The second approach uses directly the initial SOM framework to adapt the two usual steps (affectation, representation) to dissimilarity data, as for example in Kohonen & Somervuo (1998, 2002), in El Golli, Conan-Guez & Rossi (2004), and in Ambroise & Govaert, (1996).

Our work is inspired by this last approach and we have compared our proposal to the algorithms proposed by Kohonen (Kohonen & Somervuo, 1998) (Kohonen & Somervuo, 2002) and by El Golli et al. (El Golli, Conan-Guez & Rossi, 2004). Three metrics for quality estimate (quantization and neighborhood) are used for comparison. Numerical experiments on artificial and real data show the quality of the algorithm. The strong point of the proposed algorithm comes from a more accurate prototype estimate which is one of the most difficult parts of Dissimilarity SOM algorithms.

The major difficulty of the DSOM is the constraint on the output data representation. For (vector) SOM algorithm, there is a latent data model for each output **prototype** (a spherical distribution whose the prototype is the barycentre). For DSOM, there is no data model for each output prototype. One referent observation is explicitly associated to each output prototype instead of its tuning by the barycentre processing. This referent is usually chosen among the input observations at the end of an optimization process. Consequently, several prototypes can unfortunately share the same referent and these collisions provide great distortions in the output map. To avoid this difficulty, we propose here

an implicit referent for each prototype which is adapted during training iterations. So there is no collision during learning phase and consequently, the projection quality is greatly enhanced.

ADAPTATION OF SOM FOR DISSIMILARITY DATA

This article presents a new DSOM algorithm for dissimilarity data. We will first present DSOM algorithms which have been directly derived from the initial SOM framework. In the next parts, we will present in detail our proposed algorithm and some experiments to show its effectiveness in comparison with the other DSOM algorithms.

Description of DSOM Algorithms

Basically the starting point of the DSOM algorithm is the "batch" algorithm of the initial vector SOM. Let us recall this "batch" algorithm. At each iteration, the entire dataset is presented. We consider a dataset X of N observations, $X = \{o_i, i = 1..N\}$. The SOM is configured with C nodes (neurons) a priori interconnected on the output map where $\delta(c,l)$ is the distance between the nodes c and l. At iteration t, each node is represented by a prototype ω_c^t in the input space. After an initialization step, an affectation step and a representation step are sequentially processed at each iteration. The role of the former is to assign to each observation o_i, the best matching unit ω_{c*}^t, according to the **Euclidean distance**. The affectation function is:

$$c^* = Arg\left[\underset{c}{Min}\left(d^2(o_i,\omega_c)\right)\right] \quad (1)$$

Thus, a partition of the whole dataset is realized. In the latter, the prototype ω_c is adjusted to represent each **partition** X_c as well as possible. This prototype is computed as the weighted average of the input samples. The weights are evaluated through the **neighborhood function** $h^T(.)$ which is a non-increasing function of the distance on the map and controlled by a radius parameter $T(t)$ decreasing with time. At the end, the prototype ω_c is the gravity centre of the partition X_c.

This representation step cannot be directly transposed to dissimilarity data. An alternative implementation is to approximate these C prototypes by referent observations belonging to the initial dataset X. Then, this step becomes very time-consuming: all the input observations are candidate and must be evaluated. Some strategies to reduce the computation time have been proposed (Conan-Guez, Rossi & El Golli, 2006).

Let us notice $D = [d_{ij}]$ $i, j = 1..N$, the dissimilarity data. These dissimilarities describe a non metric space. However, for all the DSOM algorithms, we consider symmetric dissimilarities.

For the DSOM proposed by Kohonen, each **prototype** will be represented by one referent observation, $\omega_c = o_{r(c)}$. During the initialisation step, C observations in the input dataset are randomly assigned to the prototypes. For the affectation step, the affectation function simply uses the input dissimilarity data. Each observation is assigned to the nearest prototype:

$$f(i) = Arg\left[\underset{c}{Min}\left(d_{ir(c)}\right)\right] = Arg\left[\underset{c}{Min}\left(d\left(o_i, o_{r(c)}\right)\right)\right] \quad (2)$$

For the representation step, a new observation $o_{r(c)}$ is assigned to the prototype ω_c minimizing the following cost function:

$$r(c) = Arg\left[\underset{j}{Min}\left(E\left(c,j\right)\right)\right] \quad (3)$$

where $E(c,j)$ is the weighted local distortion if o_j is the referent of the prototype ω_c:

$$E\left(c,j\right) = \sum_{o_i \in X} h^T\left(\delta\left(c, f(i)\right)\right) d^2\left(o_j, o_i\right) \quad (4)$$

The global cost function which is then minimized is the global distortion over all the prototypes:

$$E_g = \sum_{c=1}^{C} E(c, r(c)) \quad (5)$$

For the representation step, different variants are possible. The neighborhood function in Eq. (4) can be simply integrated on the neighborhood of the prototype (the search is realized over the union of the partitions inside an output neighborhood) and not on the weighted dissimilarities. It is the "set Mean search". Also, the exponent '2' in Eq. (4) can be omitted: it is the "set **Median** search".

Different prototypes can share the same referent (collision) when the search of the referent observa-