On-line relational and multiple relational SOM

Madalina Olteanu[a], Nathalie Villa-Vialaneix[a,b]

[a] *SAMM, EA 4543, Université Paris 1, F-75634 Paris, France*
[b] *INRA, UR 875, MIAT, F-31326 Castanet-Tolosan, France*

Abstract

In some applications and in order to address real-world situations better, data may be more complex than simple numerical vectors. In some examples, data can be known only through their pairwise dissimilarities or through multiple dissimilarities, each of them describing a particular feature of the data set. Several variants of the Self Organizing Map (SOM) algorithm were introduced to generalize the original algorithm to the framework of dissimilarity data. Whereas median SOM is based on a rough representation of the prototypes, relational SOM allows representing these prototypes by a virtual linear combination of all elements in the data set, referring to a pseudo-euclidean framework. In the present article, an on-line version of relational SOM is introduced and studied. Similarly to the situation in the Euclidean framework, this on-line algorithm provides a better organization and is much less sensible to prototype initialization than standard (batch) relational SOM. In a more general case, this stochastic version allows us to integrate an additional stochastic gradient descent step in the algorithm which can tune the respective weights of several dissimilarities in an optimal way: the resulting *multiple relational SOM* thus has the ability to integrate several sources of data of different types, or to make a consensus between several dissimilarities describing the same data. The algorithms introduced in this manuscript are tested on several data sets, including categorical data and graphs. On-line relational SOM is currently available in the R package **SOMbrero** that can be downloaded at http://sombrero.r-forge.r-project.org/ or directly tested on its Web User Interface at http://shiny.nathalievilla.org/sombrero.

Keywords: Self-Organizing Map, Dissimilarity, Kernel, On-line, Multiple

Email addresses: madalina.olteanu@univ-paris1.fr (Madalina Olteanu), nathalie.villa@toulouse.inra.fr (Nathalie Villa-Vialaneix)

1. Introduction

In many real-world applications, data can not always be described by a fixed set of numerical attributes. This is the case, for instance, when data are described by categorical variables or by relations between objects (i.e., persons involved in a social network). This issue can be even trickier when the data are composed of several sources of non homogeneous information (e.g., a social network together with attributes on the nodes as in [1, 2]). A common solution to address this kind of issue is to use a measure of resemblance (i.e., a similarity or a dissimilarity) that can handle categorical variables, graphs or focus on specific aspects of the data, designed by expertise knowledge [3]. Many standard methods for data mining have been generalized to non vectorial data, recently including prototype-based clustering, even though, in some cases, the choice of the most relevant dissimilarity remains an open issue (see [4, 5] for a discussion on this topic in the field of social science). The recent paper [6] provides an overview of several methods that have been proposed to tackle complex data with neural networks.

In particular, several extensions of the Self-Organizing Map (SOM) algorithm have been proposed. One approach consists in extending SOM to categorical data by using a method similar to Multiple Correspondence Analysis, [7]. Another approach uses the median principle which consists in replacing the standard computation of the prototypes by an approximation in the original data set. This principle was used to extend SOM to dissimilarity data in [8]. One of the main drawbacks of this approach is that forcing the prototypes to be chosen among the data set is very restrictive; in order to increase the flexibility of the representation, [9] proposes to represent a class by several prototypes, all chosen among the original data set. However this method increases the computational time, while prototypes remain restricted to the original data set and may generate possible sampling or sparsity issues.

An alternative to median-based algorithms relies on a method that is close to the standard algorithm used in the Euclidean case. This method is based on the idea that prototypes may be expressed as linear combinations of the original input data. In kernel SOM framework, this setting is made natural by the use of the kernel, which maps the original data into a (large dimensional) Euclidean space (see [10, 11, 12] for on-line versions and [13]

for the batch version). Several kernels may then be used to handle complex data such as strings, nodes in a graph or graphs themselves [14]. In some cases, the data are solely described by a dissimilarity matrix. [15, 16, 17] give necessary and sufficient conditions for a symmetric matrix to be a distance matrix in an Euclidean space but, as pointed out by [3], the class of similarity/dissimilarity that can be embedded in a Euclidean space is rather limited and does not accommodate on a number of useful measures already developed in the literature. In this case, [18, 19, 20, 21] propose to introduce an implicit "convex combination" of the original data in order to extend the classical batch versions of SOM to dissimilarity data: this approach implicitly uses the embedding of the original data in a pseudo-euclidean space, as defined in [22].

However, batch versions of the SOM algorithm are known, at least for the standard numerical SOM [23], to present several drawbacks such as poor organization and strong dependency on the prototype initialization. This problem may be partially countered using PCA or MDS initializations, but when no good initialization is available, a stochastic (also called on-line) version of the algorithm can be very beneficial. The purpose of the present paper is to introduce and justify the on-line version of relational SOM, as already proposed in [24]. Such an approach leads to a better organization of the map. Additionally, taking advantage of the stochastic scheme, relational SOM is extended to integrate several sources of non homogeneous information by using an adaptive convex combination of dissimilarities. The weights of each dissimilarity are updated during the SOM learning process by an additional stochastic gradient descent step. In the remaining of this manuscript, Section 2 describes the on-line extension of the relational SOM algorithm, already studied in [24], while Section 3 describes how this approach can be used to integrate multiple information coming either from different data sets or from different dissimilarity measures. Finally, Section 4 illustrates the approach on simulated and real-world data sets and compares it with previous literature. Note that the on-line relational SOM is available in the R [25] package **SOMbrero**, which can be downloaded on R-Forge [26][1] or tested on its **shiny** [27] Web User Interface at `http://shiny.nathalievilla.org/sombrero`.

[1] `http://sombrero.r-forge.r-project.org/`

2. On-line dissimilarity SOM

Let us recall that the Self-Organizing Map (SOM) algorithm aims at mapping n input data $x_1, \ldots, x_n$ into a low dimensional grid composed of U units. A prototype p_u, valued in the same space as the input data, is associated to each unit $u \in \{1, \ldots, U\}$ of the grid. The grid induces a natural distance d on the map: for every pair of neurons (u, u'), $d(u, u')$ is usually defined as the length of the shortest path between u and u' (although other topologies are sometimes used, including the standard Euclidean distance on the grid). The algorithm aims at clustering together similar observations and also at preserving the original topology of the data set on the map (i.e., close observations are clustered into close units on the map, distant observations are clustered into distant units on the map). In order to do so, an iterative process is performed by alternating two steps. The original algorithm for numerical vectors may be resumed as follows:

- an *assignment step* where one observation (on-line version) or all observations (batch version) is/are affected to the closest prototype (in the sense of the Euclidean distance):

$$f(x_i) = \arg \min_{u=1,\ldots,U} \|x_i - p_u\|,$$

- a *representation step* where all prototypes are updated according to the new assignment. For the on-line version of the algorithm, this step is performed by mimicking a stochastic gradient descent scheme:

$$p_u^{\text{new}} = p_u^{\text{old}} + \mu H\left(d\left(f(x_i), u\right)\right)\left(x_i - p_u^{\text{old}}\right), \tag{1}$$

  where H is the neighborhood function verifying the assumptions $H : \mathbb{R}^+ \to \mathbb{R}^+$, $H(0) = 1$ and $\lim_{x\to+\infty} H(x) = 0$, and μ is a training parameter. Generally, H and μ are supposed to be decreasing with the number of iterations during the training procedure.

The original SOM algorithm described above does not posses a cost function and is not exactly a gradient descent, at least not in the continuous case. However, when the size of the neighborhood is fixed and with a modified assignment step, [28] proved that SOM is minimizing the following energy function:

$$\mathcal{E}\left((p_u)_u\right) = \sum_{u=1}^{U} \int \delta_{u,f(x)} \sum_{l=1}^{U} H(d(l,u)) \|x - p_l\|^2 P(dx) \ ,$$

where $\delta_{u,f(x)} = \begin{cases} 1, & \text{if } f(x) = u \\ 0, & \text{otherwise} \end{cases}$.

2.1. SOM for dissimilarity data

In the case where the input data take values in an arbitrary input space $\mathcal{G}$, a natural Euclidean structure is not necessarily associated with $\mathcal{G}$. Instead, the dissemblance between the observations can be described by a dissimilarity measure $\mathbf{\Delta} = (\delta_{ij})_{i,j=1,\dots,n}$ such that $\mathbf{\Delta}$ is non negative ($\delta_{ij} \geq 0$), symmetric ($\delta_{ij} = \delta_{ji}$) and null on the diagonal ($\delta_{ii} = 0$). In this case however, the assignment step cannot be carried out straightforwardly since the distances between the input data and the prototypes are not be directly computable.

Several extensions of the SOM algorithm have been proposed in this context: [8] proposes the "median SOM" where the prototypes are chosen among the input data $(x_i)_i$ in a batch framework. The assignment step is then similar to the Euclidean framework, with the dissimilarity replacing the Euclidean norm. The representation step simply finds the prototypes that minimize the energy of the map by an exhaustive search among the input data. [9, 29] extend this work by using several observations instead of a unique one for each prototype and by proposing a fast implementation of the algorithm. Nevertheless, choosing the prototypes among the input data is very restrictive and using several observations for each prototype strongly increases the computational time needed to train the map.

To overcome this difficulty, the solution proposed by [18, 19, 20, 21] is to rely on the pseudo-euclidean framework: indeed, [22] pointed out that any data described by a symmetric dissimilarity matrix can be embedded in a space consisting of the orthogonal direct sum of two Euclidean spaces, for which the inner product operation is definite positive on the first space and definite negative on the second. Relying on this framework, and similarly to the kernel SOM approach [19], the prototypes are supposed to be symbolic convex combinations of the original data (actually, convex combinations of their implicit embedding in the pseudo-euclidean space): $p_u \sim \sum_i \beta_{ui} x_i$

with $\sum_i \beta_{ui} = 1$ and $\beta_{ui} \geq 0$[2]. If β_u denotes the vector $(\beta_{u1}, \ldots, \beta_{un})$, the "distance" in the assignment step can be written in terms of Δ and β_u only:

$$\delta(x_i, p_u) \equiv \mathbf{\Delta}_i \beta_u - \frac{1}{2} \beta_u^T \mathbf{\Delta} \beta_u. \tag{2}$$

where $\mathbf{\Delta}_i$ is the *i-th* row of $\mathbf{\Delta}$ (the formula is justified and proved in Appendix A). This algorithm, called relational SOM, was proposed in the batch framework where the representation step consists in updating the convex combination by a mean calculation:

$$\beta_{ui} = \frac{H(d(f(x_i), u))}{\sum_{i'} H(d(f(x_{i'}), u))}.$$

This approach is very similar to the batch kernel SOM described in [12, 13]. In kernel SOM, the Euclidean framework is justified by the definition of a kernel $K : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ that implicitly maps the data into a Hilbert space where the inner product is directly available via the kernel. Actually, batch kernel SOM and batch relational SOM are equivalent for a dissimilarity defined from the kernel by:

$$\delta(x_i, x_j) := K(x_i, x_i) + K(x_j, x_j) - 2K(x_i, x_j). \tag{3}$$

Reciprocally, if the dissimilarity matrix Δ can be embedded in a Euclidean space (i.e., if it fulfills the condition given in [15, 16, 17] which is that the matrix with elements $s_{ij} = \left(\delta(x_i, x_n)^2 + \delta(x_j, x_n)^2 - \delta(x_i, x_j)^2\right)/2$ is positive, or, similarly, if the matrix with elements

$$s(i,j) = -\frac{1}{2}\left(\delta^2(x_i, x_j) - \frac{1}{n}\sum_{k=1}^{n} \delta^2(x_i, x_k) - \frac{1}{n}\sum_{k=1}^{n} \delta^2(x_k, x_j) + \frac{1}{n^2}\sum_{k,k'=1}^{n} \delta^2(x_k, x_{k'})\right)$$

as proposed in [30], is positive), then relational SOM is equivalent to kernel SOM used with the matrix $(s_{ij})_{ij}$, which, in this case, is a kernel. However, as explained in [3], some useful dissimilarities (e.g., shortest path lengths in graphs or optimal matching dissimilarities for sequences of events, [31, 32])

[2]Note that this sum has no real meaning, most of the times, as $\mathcal{G}$ is not necessarily equipped with a + operation neither with a multiplication by a scalar. It simply implicitly refers to the + operation in the underlying pseudo-euclidean space: the formal definition of p_u is given in Appendix A.

do not fulfill the required conditions allowing them to be embedded in a Euclidean space. In these cases, the dissimilarity can be turned into a kernel using various pre-processings, as described in [33] but then, relational SOM and kernel SOM are no longer identical.

2.2. On-line relational SOM

As explained in [23], although batch SOM possesses the nice properties of being deterministic and of usually converging in a few iterations, it has several drawbacks such as organizing the map rather poorly, producing unbalanced classes and being strongly dependent on the initialization. Hence, using the same ideas as [18, 20], we introduce the on-line relational SOM, which generalizes the on-line SOM to the case of dissimilarity data. The proposed method is described in Algorithm 1. In this algorithm, only one observation,

Algorithm 1 On-line relational SOM

1: For all $u = 1, \ldots, U$ and $i = 1, \ldots, n$, initialize β_{ui}^0 such that $\beta_{ui}^0 \geq 0$ and $\sum_i^n \beta_{ui}^0 = 1$.
2: **for** t=1,...,T **do**
3: Randomly choose an input x_i
4: *Assignment step*: find the unit of the closest prototype

$$f^t(x_i) \leftarrow \arg\min_{u=1,\ldots,U} \left(\left(\beta_u^{t-1}\mathbf{\Delta}\right)_i - \frac{1}{2}(\beta_u^{t-1})^T \mathbf{\Delta} \beta_u^{t-1} \right)$$

5: *Representation step*: $\forall\, u = 1, \ldots, U$,

$$\beta_u^t \leftarrow \beta_u^{t-1} + \mu(t) H^t(d(f^t(x_i), u)) \left(\mathbf{1}_i - \beta_u^{t-1}\right)$$

where $\mathbf{1}_i$ is a vector with a single non null coefficient at the ith position, equal to one.
6: **end for**

randomly chosen, is assigned to a unit of the map at each iteration step. The representation step is drawn from Equation (1) by using a similar approach to update the prototypes' coordinates $(\beta_{ui})_{ui}$. Note that the constraints on $(\beta_{ui})_{ui}$ are preserved since:

- $\sum_i \beta_{ui}^t = 1$ (as demonstrated in Appendix B);

- $\beta_{ui}^t \geq 0$ for any u and i as long as $\mu(t)$ is small enough ($\mu(t)H^t$ must simply be smaller than 1).

This latter condition is easy enough to handle. In our experiments, the parameters of the algorithm are chosen according to [34]: the neighborhood H^t decreases in a piecewise linear way, starting from a neighborhood which corresponds to the whole grid up to a neighborhood restricted to the neuron itself; $\mu(t)$ vanishes at the rate of $1/t$.

2.3. Discussion on the algorithm: relations to previous algorithms, complexity and convergence

If the dissimilarity matrix is a Euclidean distance, then the on-line relational SOM is exactly identical to the standard numerical SOM as long as the prototypes of the original SOM are initialized in the convex hull of the original data (i.e., the initial prototypes can be written $p_u^0 = \sum_i \beta_{ui}^0 x_i$). Similarly, the on-line relational SOM is identical to on-line kernel SOM as described in [10, 11, 12] for a dissimilarity defined from a kernel K by Equation (3) or if the dissimilarity fulfills one of the conditions in [15, 16, 17].

Moreover, if one wants to generalize dissimilarities to non-symmetric relations (such as, for example, graph-based comparisons of protein fingerprint graphs), a dissimilarity matrix computed as the half-sum of pairwise relations may be considered as the input for the algorithm.

In order to illustrate the performances of the on-line relational SOM compared to the batch implementation, 500 points are considered, sampled randomly from the uniform distribution in $[0, 1]^2$. The dissimilarity is computed as the length of the shortest path in the graph induced by the Delaunay triangulation (this graph is displayed in Figure 1). Note that this dissimilarity is not exactly equivalent to the Euclidean $\mathbb{R}^2$-metric, since it is not even Euclidean. The batch version of relational SOM and the on-line version of relational SOM were trained on identical 10×10 grid structures. The algorithms were trained either with identical initializations, or with a PCA initialization[3] for the batch SOM, which is the standard initialization used to alleviate the initialization dependency of this algorithm. Results are available in Figure 1 and clearly show a much better organization of the prototypes in the final grid provided by the on-line version of the algorithm. When the

[3]Dissimilarity PCA was used and then properly re-scaled to satisfy the condition $\sum_i \beta_{ui} = 1$.

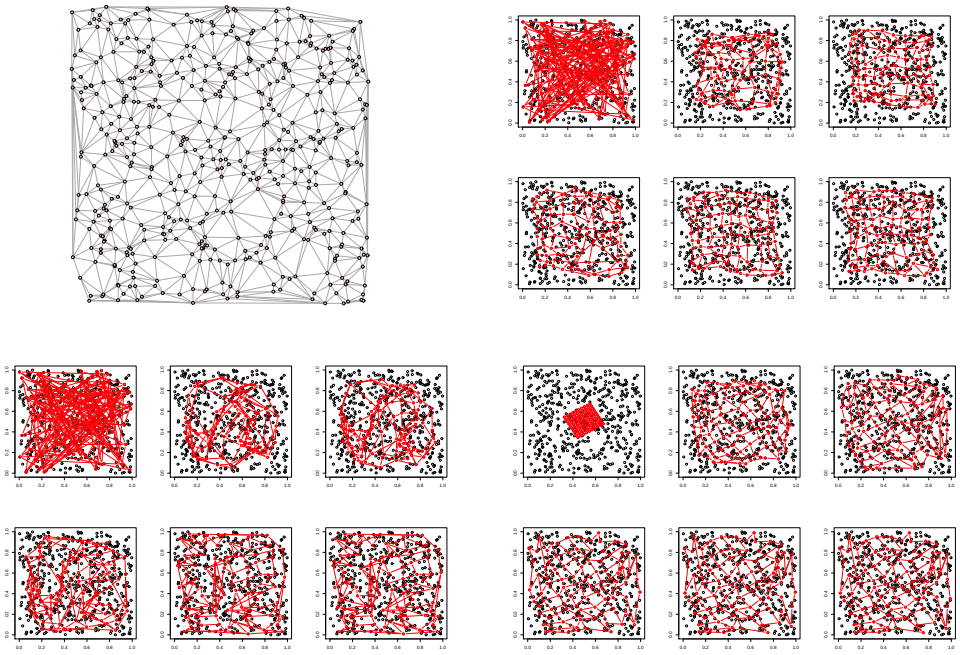


Figure 1: 500 points sampled from the uniform distribution in $[0,1]^2$ and their Delaunay graph (top left) and map organizations obtained by relational on-line SOM (top right) and relational batch SOM (bottom left) with random initialization and by relational batch SOM with dissimilarity-PCA initialization (bottom right).

prototypes are initialized with a PCA, the organization of the map produced by the batch kernel SOM is much better but still slightly worse than the one obtained with the on-line version and a random initialization. This visual effect is confirmed when calculating the topographic error [35]: this error quantifies the continuity of the map with respect to the input space metric by counting the number of times the second best matching unit of a given observation belongs to the direct neighborhood of the best matching unit for this observation. A topographic error equal to 0 means that all second best matching units are in the direct neighborhood of the winner neurons and thus that the original topology of the data is well preserved on the map. In this simple example, it is equal to 0.01 for the on-line relational SOM, to 0.176 for the batch relational SOM with PCA initialization and to 0.264 for the batch relational SOM with random initialization. Hence, the classical initialization dependency of the batch version of the algorithm, as already

shown in [23] also holds for the relational approach. In particular, when no good initialization is present (i.e., when PCA or MDS are bad initialization strategies), the on-line version can then be very beneficial. Finally, the complexity of the on-line and batch versions are similar (of order $\mathcal{O}(Un^2)$) with usually a smaller number of iterations needed to stabilize the batch version: the convergence of the batch version is attained with quadratic speed while the on-line version converges with a linear speed. However, the better organization of the map compensates for this small loss in computational time. Finally, let us remark that formally speaking, in the pseudo-Euclidean setting, the convergence of both algorithms (on-line and batch) is even not guaranteed (saddle points can be present instead of local optima, as pointed out in [21]) but, in practical applications, divergence was never observed.

3. Integrating multiple dissimilarities

In some specific applications, the user is interested in simultaneously analyzing several sources of information: a graph together with additional information known on its nodes, numerical variables measured on individuals together with factors describing these individuals... This situation is often referred to as "multiple view" data and such data are quite common in a number of fields: gene clustering from expression profiles and ontology information [36] in biology, node clustering in a social network taking into account attributes that describe the nodes [37, 1] in social sciences, and molecules clustering from fingerprints and spatial structures [38] in chemistry. In other specific applications, the data set can be described by several dissimilarities, each encoding specific features of the data but none of them being acknowledged as more informative than the others: in social sciences for instance, the choice of a good dissimilarity to describe the resemblance between two event time series is still an open issue [4, 5].

The combination of all sources of information or of several dissimilarities is a challenging problem that aims at increasing the relevance of the clustering. In clustering, this issue has already been tackled by different approaches: some rely on clustering ensembles, combining together the clusterings obtained from each view or from each dissimilarity into a consensus clustering [39]. A more complex strategy, described in [40], iteratively updates the different clusterings using a global log-likelihood approach until they converge to a consensus. Other authors propose to concatenate all data/views prior to the clustering. If kernels are available, this method is known as multi-

ple kernel clustering: the different kernels are combined by using a convex combination and the coefficients of the convex combination are optimized together with the clustering [41, 42]. In a similar way, if the data are described by numerical variables belonging to different feature groups, [43] proposes to weight each group and to optimize simultaneously the clustering and the weights of the groups.

For the SOM algorithm as well, a few articles tackle related issues: in particular, [44] combines numeric and binary variables to produce a single map by optimizing two quantization energies in parallel and [1, 2] use a multiple kernel framework to integrate various information.

In the present section, we use a similar approach by combining different dissimilarities in a convex combination. We propose an algorithm which learns an optimal combination on-line, by minimizing the energy function.

3.1. Computing a multiple dissimilarity

Suppose now that the observations $x_1, \ldots, x_n$ are not described by a single dissimilarity matrix $\mathbf{\Delta}$, but by D dissimilarity matrices $\mathbf{\Delta}^1, \ldots, \mathbf{\Delta}^D$, where $\Delta^d = \left(\delta^d(x_i, x_j)\right)_{ij}$. The dissimilarities can be either different dissimilarities computed on the same data or dissimilarities computed from different variables measured on the same individuals (e.g., a dissimilarity that measures proximities between nodes in a graph and a dissimilarity that measures proximities between the node labels, see Section 4.3 for an example).

Similarly to the multiple kernel approach described in [45] or in [1] (for multiple kernel SOM), we propose to combine all the dissimilarities into a single one, defined as a convex combination:

$$\delta_{ij}^{\alpha} = \sum_{d=1}^{D} \alpha_d \delta_{ij}^{d} \tag{4}$$

where $\alpha_d \geq 0$ and $\sum_{d=1}^{D} \alpha_d = 1$. In the Euclidean framework, this approach is strictly equivalent to the multiple kernel SOM approach because $\|x_i - x_j\|_d^2 = \langle x_i - x_j, x_i - x_j \rangle_d$ (multiple kernel is a convex combination of dot products whereas Equation (4) is based on a convex combination of squared distances).

3.2. On-line multiple relational SOM

If the (α_d) are given, relational SOM based on the dissimilarity introduced in Equation (4) aims at minimizing (over $(\beta_u)_u$) the following energy function

$$\mathcal{E}((\beta_u)_u, (\alpha_d)_d) = \sum_{u=1}^{U} \sum_{i=1}^{n} H\left(d\left(f(x_i), u\right)\right) \delta^{\alpha}\left(x_i, p_u(\beta_u)\right),$$

where $\delta^{\alpha}\left(x_i, p_u(\beta_u)\right)$ is defined as in Equation (2) by

$$\delta^{\alpha}\left(x_i, p_u(\beta_u)\right) \equiv \mathbf{\Delta}_i^{\alpha} \beta_u - \frac{1}{2} \beta_u^T \mathbf{\Delta}^{\alpha} \beta_u \tag{5}$$

with $\mathbf{\Delta}^{\alpha} = \sum_d \alpha_d \mathbf{\Delta}^d$

When there is no a-priori on the $(\alpha_d)_d$, we propose to include the optimization of the convex combination within the on-line algorithm which trains the map. This idea is similar to the one proposed in [46] for optimizing a kernel parameter in vector quantization algorithms. More precisely, a stochastic gradient descent step is added to the original on-line relational SOM algorithm to optimize the energy $\mathcal{E}((\beta_{ui})_{ui}, (\alpha_d)_d)$, over both $(\beta_{ui})_{ji}$ and $(\alpha_d)_d$. To perform the stochastic gradient descent step on the (α_d), the computation of the derivative of

$$\mathcal{E}|_{x_i} = \sum_{u=1}^{U} H\left(d\left(f(x_i), u\right)\right) \delta^{\alpha}\left(x_i, p_u(\beta_u)\right)$$

(the contribution of the randomly chosen observation $(x_i)_i$ to the energy) with respect to α is needed. Since

$$\frac{\partial}{\partial \alpha_d}\left[\delta^{\alpha}(x_i, p_u)\right] = \delta^d(x_i, p_u),$$

we have

$$\mathcal{D}_{id} = \frac{\partial \mathcal{E}|_{x_i}}{\partial \alpha_d} = \sum_{u=1}^{U} H\left(d\left(f(x_i), u\right)\right) \left(\mathbf{\Delta}_i^d \beta_u - \frac{1}{2} \beta_u^T \mathbf{\Delta}^d \beta_u\right).$$

Following an idea similar to that of [45], the SOM is trained by performing, alternatively, the standard steps of the SOM algorithm (i.e., assignment and representation steps) and a gradient descent step for the $(\alpha_i)_i$. The methodology is described in Algorithm 2.

Algorithm 2 On-line multiple dissimilarity SOM

1: For all $u = 1, \dots, U$ and $i = 1, \dots, n$, initialize β_{ui}^0 such that $\beta_{ui}^0 \geq 0$ and $\sum_{i=1}^n \beta_{ui}^0 = 1$.

2: For all $d = 1, \dots, D$, initialize $\alpha_d^0 \in [0, 1]$ st $\sum_d \alpha_d^0 = 1$. **return** $\delta^{\alpha,0} \leftarrow \sum_d \alpha_d^0 \delta^d$.

3: **for** t=1,...,T **do**

4: Randomly choose an input x_i

5: *Assignment step*: find the unit of the closest prototype

$$f^t(x_i) \leftarrow \arg \min_{u=1,\dots,U} \delta^{\alpha,t-1}\left(x_i, p_u(\beta_u)\right)$$

where $\delta^{\alpha,t-1}\left(x_i, p_u(\beta_u)\right)$ is defined as in Equation (5).

6: *Representation step*: update all prototypes according to the new assignment: $\forall\, u = 1, \dots, U$,

$$\beta_u^t \leftarrow \beta_u^{t-1} + \mu(t) H\left(d\left(f(x_i), u\right)\right)\left(\mathbf{1}_i - \beta_u^{t-1}\right)$$

7: *Gradient descent step*: update the convex combination parameters: $\forall\, d = 1, \dots, D$,

$$\alpha_d^t \leftarrow \alpha_d^{t-1} + \nu(t)\mathcal{D}_d^t$$

where $\mathcal{D}_d^t$ is the descent direction and update $\delta^{\alpha,t}$

$$\delta^{\alpha,t} \leftarrow \sum_d \alpha_d^t \delta^d.$$

8: **end for**

To ensure that the gradient step respects the constraints on α ($\alpha_d \geq 0$ and $\sum_d \alpha_d = 1$), the following strategy is used: similarly to [47, 48, 45], the gradient $\left(\frac{\partial \mathcal{E}^{t-1}|_{x_i}}{\partial \alpha_d}\right)_d$ is reduced and projected such that the non-negativity of α is ensured. The following modified descent step is thus used:

$$\widetilde{\mathcal{D}}_d = \begin{cases} 0 & \text{if } \alpha_d = 0 \text{ and } \mathcal{D}_d - \mathcal{D}_{d_0} > 0 \\ -\mathcal{D}_d + \mathcal{D}_{d_0} & \text{if } \alpha_d > 0 \text{ and } d \neq d_0 \\ \sum_{d \neq d_0,\ \alpha_d > 0} \left(\mathcal{D}_d - \mathcal{D}_{d_0}\right) & \text{otherwise} \end{cases}$$

The descent step $\nu(t)$ is decreased with the standard rate of ν_0/t with an initial ν_0 small enough to ensure the positivity constraint on $(\alpha_d)_d$.

4. Applications

In this section, several applications, on simulated or real-life data sets, illustrate the performances of the proposed methods. Section 4.1 compares on-line and batch relational SOM on a DNA barcoding data set, Section 4.2 compares the use of dissimilarities and kernels for mapping two political graphs into a grid, Section 4.3 illustrates the efficiency of the use of a multiple dissimilarity approach on a simulated data set and, finally, Section 4.4 applies the multiple relational SOM to a large data set of categorical time series and shows that the multiple relational SOM approach can be used to interpret which dissimilarities produce the most relevant clusters.

4.1. Comparison between on-line and batch relational SOM on a genetic data set

This first experiment aims at providing a comparison between on-line and batch relational SOM. It is performed on a data set that contains 465 input data issued from ten unbalanced sampled species of Amazonian butterflies. This data set was previously used by [49] to demonstrate the synergy between DNA barcoding and morphological-diversity studies. The notion of DNA barcoding comprises a wide family of molecular and bioinformatics methods aimed at identifying biological specimens and assigning them to a species. According to the vast literature published during the past years on the topic, two separate tasks emerge for DNA barcoding: on the one hand, assign unknown observations to known species and, on the other hand, discover undescribed species, [50]. The second task is usually approached with the Neighbor Joining algorithm [51] which constructs a tree similar to a dendrogram. When the sample size is large, the trees become rapidly unreadable. Moreover, they are quite sensitive to the order in which the input data are presented. Unsupervised learning and visualization methods are used to a very limited extent by the DNA barcoding community, although the information they bring may be quite useful. Self-organizing maps provide a visualization of the data while bringing out clusters or groups of clusters that may correspond to yet unknown species.

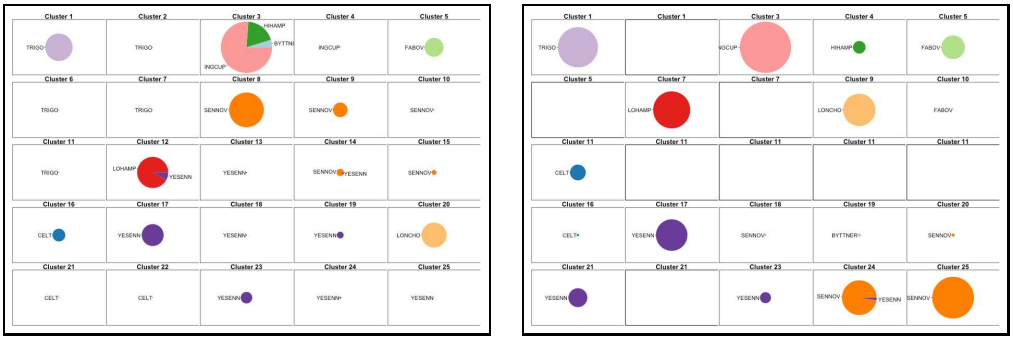
DNA barcoding data are composed of sequences of nucleotides, i.e. sequences of "a", "c", "g", "t" letters in high dimension (hundreds or thousands of sites). Hence, since the data are not Euclidean, dissimilarity-based methods appear to be more appropriate. Specific distances and dissimilarities such as the Kimura-2P [52] are usually computed. Recently, batch median SOM was

tested in [53] on several data sets, amongst which the Amazonian butterflies. Although median SOM provided encouraging results, two main drawbacks emerged. First, since the algorithm was run in batch, the organization of the map was generally poor and highly depending on the initialization. Second, since the algorithm calculates a prototype for each cluster among the data set, it does not allow for empty clusters. Thus, the existence of species or groups of species was difficult to acknowledge. The use of on-line relational SOM overcomes these two issues. Figure 2 contains the maps obtained with median SOM and relational SOM with PCA initialization, both trained in batch versions[4] and Figure 3 illustrates the mapping produced with the on-line relational SOM. The three algorithms were run with identical fixed seeds for the random generators. The clustering quality of median SOM is poor, since several clusters mix together several species. On the contrary, relational SOM allows for empty clusters and thus produces a better mapping, from a clustering point of view: the only mixing class corresponds to a labeling error. Moreover, the empty cells help separating the main groups of species. Clustering may thus be useful in addressing misidentification issues.

Topographic errors were computed for the three mappings in order to assess the quality of the projection. For the online algorithm, the error is 0.0022, for relational SOM with PCA initialization we obtained 0.3682, while the error of median SOM is 0.3094. Hence, the stochasticity of the on-line algorithm allowed for a better organization of the map, compared with batch algorithms.

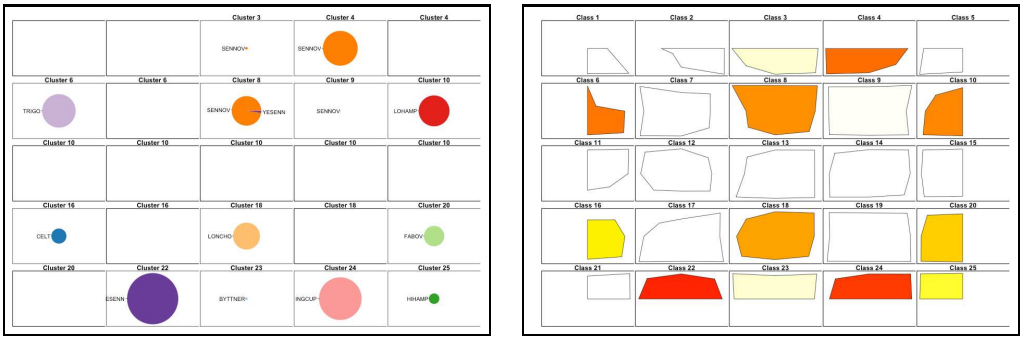
In Figure 3b, distances with respect to the nearest neighbors were computed for each node. The distance between two nodes/cells is computed as the mean dissimilarity between the observations within each class. A polygon is drawn within each cell with vertices proportional to the distances to its neighbors. If two neighbor prototypes are very close, then the corresponding vertices are very close to the edges of the two cells. If the distance between neighbor prototypes is very large, then the corresponding vertices are far apart, close to the center of the cells.

[4]relational batch SOM with random initialization was also tested but, since the results were worse than the ones obtained with PCA initialization, they are not shown in this article.



(a) (b)

Figure 2: Species diversity distribution by cluster (radius proportional to the size of the cluster): Median batch SOM (a) and Relational batch SOM with PCA initialization (b).



(a) (b)

Figure 3: On-line relational SOM results for Amazonian butterflies: (a) Species diversity distribution by cluster (radius is proportional to the cluster size. (b) Distances between prototypes.

4.2. On-line relational SOM and on-line kernel SOM to decipher the structure of political networks

This present section's purpose is to give a comparison of the performances obtained with relational SOM when used with various metrics. More precisely, we will show that for structural data such as graphs, a kernel is not always the most relevant way to extract information from the graph structure, compared to, i.e., the simple similarity based on the length of the shortest path between two nodes.

The data used in this section come from two famous data sets pertaining to the US politics. The first data set is a graph where the nodes are 105 American political books, all published around the presidential election of

2004 and sold by Amazon.com. The edges of this graph encode the fact that two books were co-purchased by a common buyer[5]. All nodes are labeled according to their political affiliation (conservative, liberal or neutral), and this information will be used to validate the results a posteriori.

The second data set is a graph representing the US politics blogosphere, recorded in 2004, for the same presidential election as the previous one, by Adamic and Glance [54]. This data set contains 1 222 nodes which are political blogs and 16 714 edges that represent a hyperlink between two blogs[6]. Again, additional information pertaining the political preference of the blog is also provided (here only conservative or liberal). Both graphs are represented in Figure 4 by a Fruchterman and Reingold [55] force directed placement algorithm and nodes are colored according to the political affiliation of the book or of the blog.

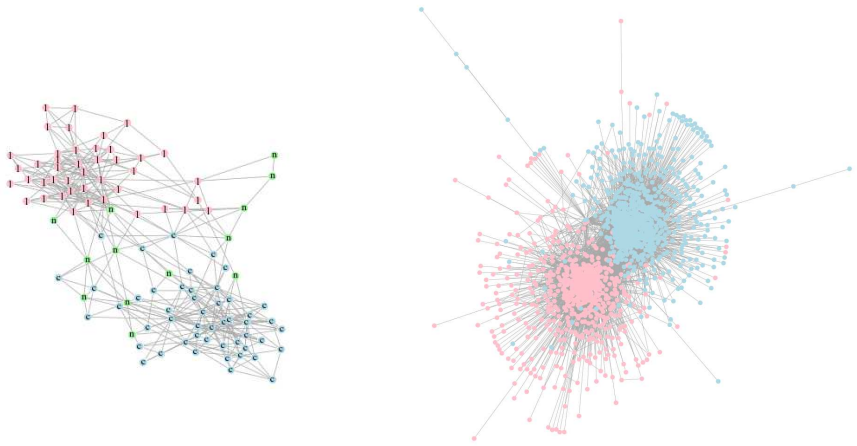


Figure 4: Political books (left) and blogs (right) networks. Nodes are labeled according to the political orientation of the book or of the blog: pink is for conservative, blue for liberal and green for neutral.

Relational SOM was performed to project the nodes of the two graphs on a square grid having dimension 5×5 (books) and 10×10 (blogs). Three different dissimilarities were used to perform this task:

[5]This graph was built by Valdis Krebs and is available for downloading at http://www-personal.umich.edu/~mejn/netdata/polbooks.zip.

[6]The original graph was directed but we only used undirected edges to perform our analysis.

- the length of the shortest path between two nodes. Note that, in general, the length of the shortest path is not a Euclidean distance: for the two graphs described in this section, the condition of [16] is not satisfied;

- a dissimilarity defined as the square of the distance induced by the heat kernel ($K = e^{-\gamma L}$ where L is the Laplacian, [56]), with parameters $\gamma = 0.1$ and 1. In this case, relational SOM is equivalent to kernel SOM as described in [18, 20];

- a dissimilarity defined as the square of the distance induced by the commute time kernel [57].

The performances of the tested methods were assessed using three criteria: the modularity of the obtained partition, the neurons' purity (compared to the political labels) and the topographic error of the map. The modularity [58] is a measure of quality of a partition of the nodes in a graph:

$$\mathcal{Q} = \sum_{u=1}^{U} \sum_{i,j:\, f(x_i)=f(x_j)=u} \left(E_{ij} - \frac{d_i d_j}{2m} \right)$$

where $E_{ij} = 1$ iff there is an edge between nodes x_i and x_j, d_i is the degree of node x_i and m is the number of edges in the graph. The best partition corresponds to the largest modularity. The neurons' purity is a measure of the consistency of the clustering with respect to the political labels: it counts the frequency of the political labels of the nodes that are equal to the majority political label of the node's cluster. The closer to 1 the purity is, the better the clustering is. The last quality criterium, the topographic error of the map [35], quantifies the continuity of the map, with respect to the input-space metric as already explained in Section 2.3. Notice that, as it computes the second best matching unit, the topographic error depends on the metric of the input space itself and tells us if this metric is well preserved on the map.

The results are given in Table 1. In addition, Figures 5 (books) and 6 (blogs) display two of the maps obtained for each data set. First note that the modularity obtained with the SOM algorithm should not be compared with that of a standard node clustering algorithm: the number of clusters used in such maps is often much larger than the optimal number of clusters for the modularity (for instance, the optimal modularity found by the algorithm

Dissimilarity	Shortest path length	Heat kernel $\gamma = 0.1$	Heat kernel $\gamma = 1$	Commute time kernel
	Political books			
modularity	0.25	-0.05	0.08	**0.27**
purity	0.88	0.61	0.72	**0.89**
topo. error	0.048	0.133	**0.038**	**0.038**
	Political blogs			
modularity	**0.08**	0.02	0.00	0.00
purity	**0.93**	0.89	0.79	0.57
topo error	0.303	**0.047**	0.322	0.899

Table 1: Modularity, neurons' purity and topographic error obtained for the data sets "political books" and "political blogs" by relational and kernel SOM algorithms.

described in [59] gives only 10 clusters, that should be compared to the 100 clusters of the map). Nevertheless, this measure of the clustering quality is still valid for comparing different dissimilarities.

For the political books data set, the best map is obtained by using the on-line kernel SOM algorithm with the commute time kernel. The on-line relational SOM with the shortest path dissimilarity obtains comparable performance but the heat kernel gives poor results, whatever the value of γ. For the political blogs, the on-line relational SOM gives good results and manages to discriminate the two groups of blogs quite well, while its topographic error is rather large. On the other hand, the kernel SOM with the heat kernel ($\gamma = 0.1$) has a much better topographic error, while it badly discriminates the labels and produces a bad clustering of the nodes of the graph on the map (as measured by the modularity): this can be explained by the fact that, even though the map properly represents the topographic organization of the input space, the metric used to represent the data may not be the most accurate to emphasize some particular features of the data that can be of a major interest for the user.

In a second step, a hierarchical clustering of the prototypes was performed. Using the symbolic representation of the prototypes as $p_u \sim \sum_{i=1}^{n} \beta_{ui} x_i$, the dissimilarity between two prototypes can be expressed as:

$$\delta(p_u, p_{u'}) := -\frac{1}{2} (\beta_u - \beta_{u'})^T \Delta (\beta_u - \beta_{u'}) \tag{6}$$

and used as an input in the hierarchical clustering algorithm (see [20], The-

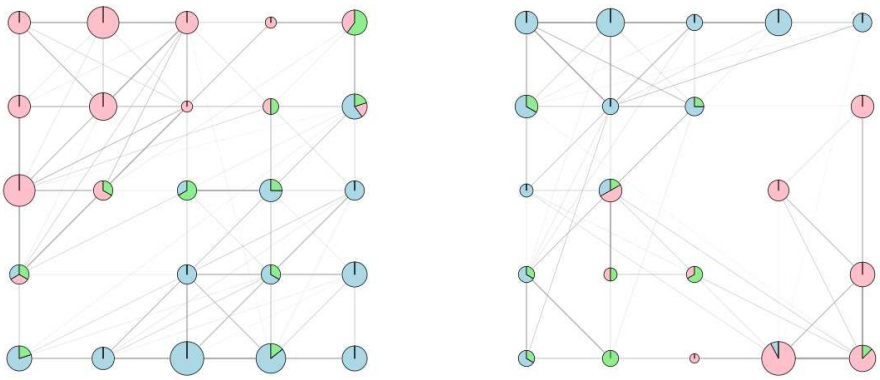


Figure 5: **Political books.** Maps obtained by the on-line relational SOM algorithm with the shortest path dissimilarity (left) and by the on-line kernel SOM with the commute time kernel (right).

orem 1, for a justification of this formula). Only three and two clusters were kept for, respectively, the political blogs and the political books data sets in order to try to retrieve the original labels. The resulting clusters are displayed in Figures 7 and 8. Moreover, the classes' purity and modularity are given in Table 2.

Dissimilarity	Shortest path length	Heat kernel $\gamma = 0.1$	Heat kernel $\gamma = 1$	Commute time kernel
	Political books			
modularity	**0.50**	-0.02	-0.00	0.41
classes' purity	**0.84**	0.49	0.47	0.76
	Political blogs			
modularity	**0.39**	0.04	-0.00	0.00
classes' purity	**0.91**	0.52	0.58	0.52

Table 2: Modularity and purity of the classes obtained by a hierarchical clustering of the prototypes, for the data sets "political books" and "political blogs".

As expected, hierarchical clustering tends to slightly decrease the classes' purity (compared to the neurons' purity) and to strongly increase the modularity. But it also affects which of the dissimilarities seems to represent the data better: for both data sets, the shortest path dissimilarity overcomes the dissimilarities based on kernels. This shows that the use of a kernel is

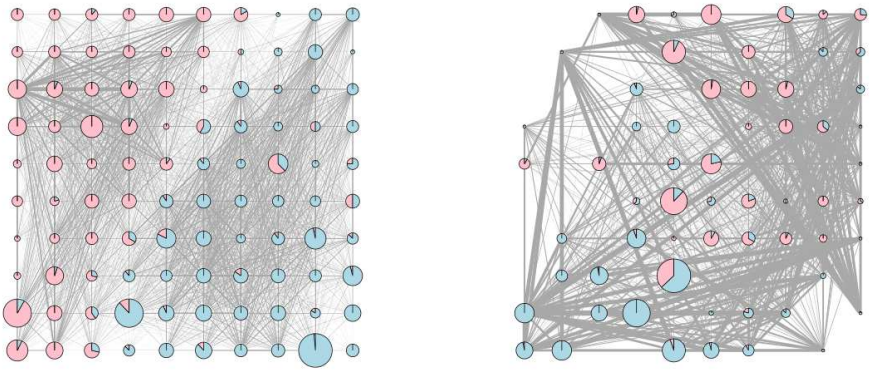


Figure 6: **Political blogs.** Maps obtained by the on-line relational SOM algorithm with the shortest path dissimilarity (left) and by the on-line kernel SOM with the heat kernel $\gamma = 0.1$ (right).

not always the best possible choice for computing similarities/dissimilarities between observations and that allowing the use of a larger family of dissimilarities can be useful in some cases.

4.3. Multiple relational SOM on simulated data

In this section, a simple example is used to test the algorithm and illustrate its behavior in the presence of complementary information. 200 observations, divided into 8 groups (indexed from 1 to 8 in the following), were generated using three different types of data:

- an unweighted graph, simulated similarly as the "planted 3-partition graph" described in [60]. The nodes of the groups 1 to 4 and the nodes of the groups 5 to 8 could not be distinguished in the graph structure: the edges within these two sets of nodes were randomly generated with a probability equal to 0.3. The edges between these two sets of nodes were randomly generated with a probability equal to 0.01;

- numerical data that were two dimensional Gaussian vectors. The variables corresponding to observations of odd groups were simulated by Gaussian vectors with mean $(0, 0)$ and independent components having a variance equal to 0.3 and the variables corresponding to observations of even groups were simulated by Gaussian vectors with mean $(1, 1)$ and independent components having a variance equal to 0.3;

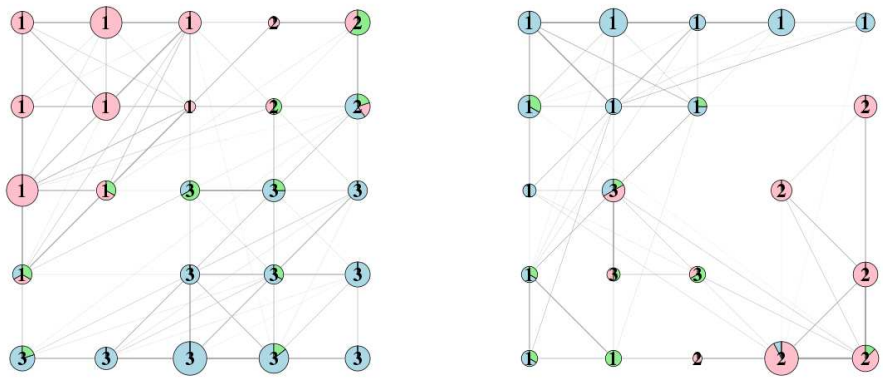


Figure 7: **Political books.** Maps obtained by the on-line relational SOM algorithm with the shortest path dissimilarity (left) and by the on-line kernel SOM with the commute time kernel (right).

- a factor with 2 levels. Observations of groups 1, 2, 5, and 7 were affected to the first level and observations of the other groups to the second level.

Hence, only the combined knowledge of the three data sets gave access to the eight original groups. The multiple relational SOM algorithm was applied to this problem with the shortest path distance for the graph, the standard Euclidean distance for the numerical data and Dice's distance for the factor variable (equal to 0 if the factors are identical between the two observations and to 1 if not). The algorithm was compared with

- a multiple kernel SOM approach as described in [2] where the kernels used were the commute time kernel [57] for the graph and the Gaussian kernel for both the other data sets (the factor was recoded as a numeric variable using its disjunctive form). The parameter of the Gaussian kernel was set as recommended in [61];

- a standard relational SOM approach using one of the three data sets only. It was also compared to the dissimilarity SOM using numerical and factor data or all the three data sets but used as if they were issued from the same data set with a Euclidean distance (when the graph was added to the numerical and factor data, it was under the form of its adjacency matrix).

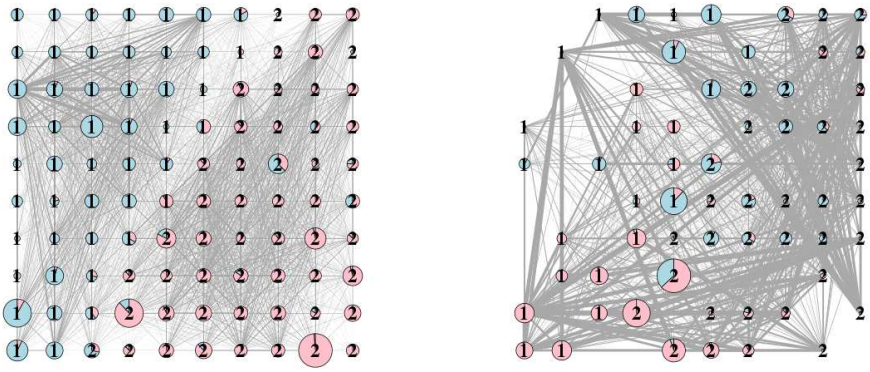


Figure 8: **Political blogs.** Maps obtained by the on-line relational SOM algorithm with the shortest path dissimilarity (left) and by the on-line kernel SOM with the heat kernel $\gamma = 0.1$ (right).

The comparison was performed on 100 different data sets generated as previously described.

The performances of the different approaches were compared using the normalized mutual information [62] with respect to the original classes, the average node purity, taking again as a reference the original classes, and the topographic error [35]. The first two quality measures quantify the adequation between the original classes and the clustering provided by the SOM. The node purity has values between 0 and 1 and is equal to 1 when the two partitions are identical. The last quality measure, the topographic error, does not depend on the original class but it quantifies the continuity of the map, with respect to the input space metric. The results are given in Figure 9, which displays the distributions of the normalized mutual information, the nodes' purity and the topographic error, over the 100 data sets. Figure 10 provides examples of resulting maps.

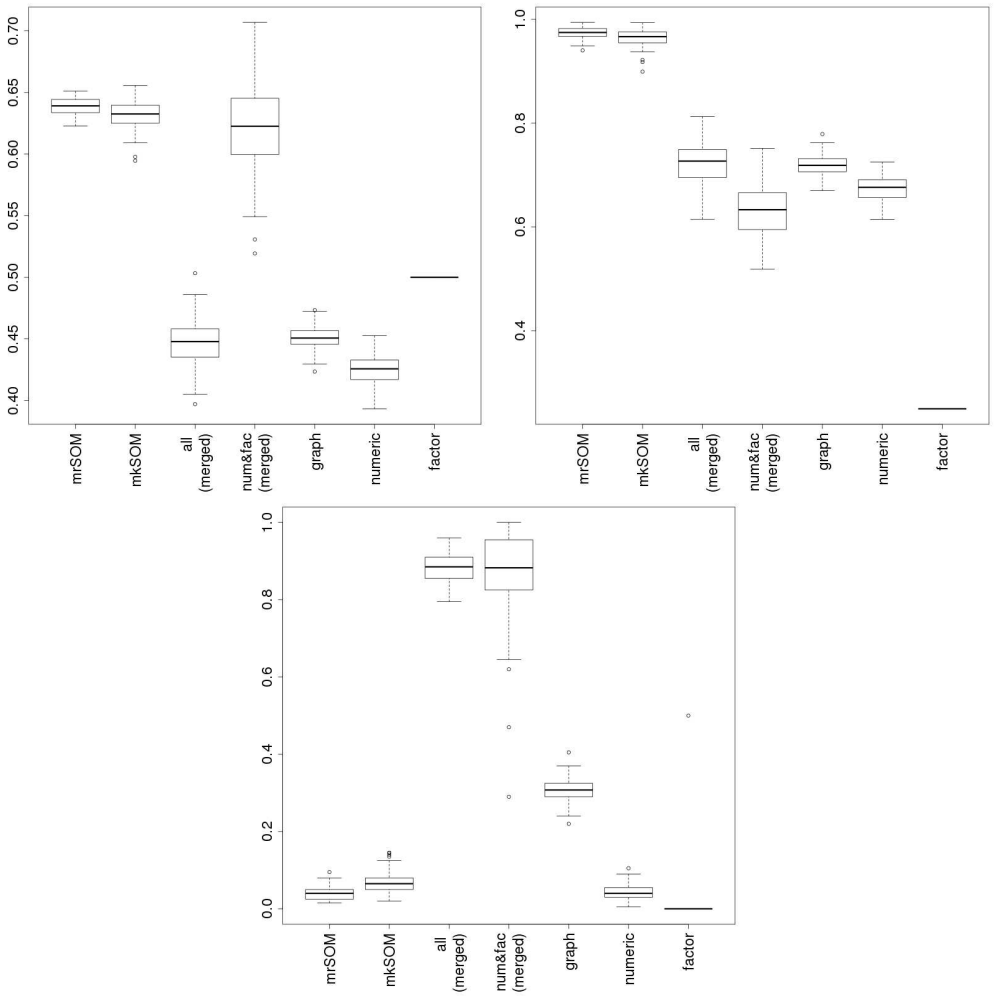


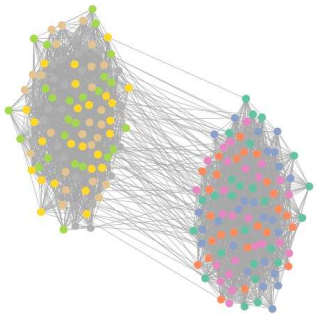
Figure 9: Normalized Mutual Information (top left), neurons' purity (top right) and topographic error (bottom) of multiple relational SOM, multiple kernel SOM and relational SOM used with all or two data sets (num&fac) simply merged in a single data set or with a single data set (graph, numeric or factor).

Taking into account the clustering quality (normalized mutual information), the node purity and the topographic quality, the multiple relational SOM outperforms the other methods. The difference between the use of the shortest-path dissimilarity and a kernel for graph in a similar multiple setting is small but still significant (with p-values smaller than 10^{-9} for Wilcoxon paired tests).

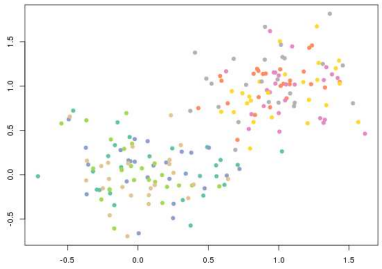
Note that the normalized mutual information gives here a pessimistic vision of the results because it penalizes the fact that the original clusters are separated into several neurons on the grid. This explains the good performance (despite their large variability), in term of normalized mutual information, of the grid built from the numeric variables and the factor only because this latter map contains much more empty clusters as shown in Figure 10. On the contrary, the example of the map resulting from on-line multiple relational SOM in Figure 10 shows a good classification and a good organization according to the three types of information: the eight groups are almost perfectly distinguished by the algorithm.

Also note that the topographic error is not an optimal way to compare the results obtained with data sets that do not contain the same amount of information: indeed, the very good topographic error obtained by the map trained from the numeric data only or the factor only simply means that the topographic properties of these data is well preserved on the map but this cannot be compared to the multiple relation SOM, the multiple kernel SOM or the map trained with all data and a standard SOM: these maps are supposed to preserve the topographic properties of all three sorts of data, which is a harder task than preserving the topographic property of only one sort (numeric, factor, graph) of data. In this case, merging all data in a single data set which is then passed as an input to a numeric SOM leads to a very bad topographic error (approximately 30 times larger than the one obtained with multiple relational SOM or multiple kernel SOM).

The evolution of the α, shown in Figure 10 is also interesting: the Dice's distance, which is the only similarity measure based on a non noisy set of data obtains larger weights than the other two dissimilarities. This is consistent with the fact that these data are indeed the best of the three data sets to distinguish between the original clusters: as shown in Figure 9, the map based on the factor is better in terms of normalized mutual information than the ones based on the numeric variables or on the graph only (its node purity is very low because it perfectly distinguished the data into two clusters where four original clusters are equally mixed).

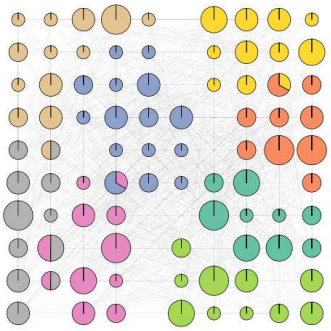


Graph

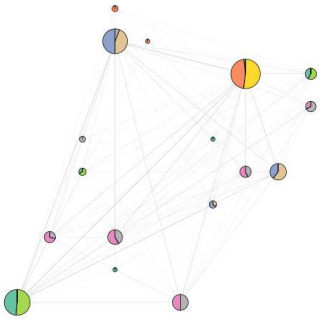


Numerical variables

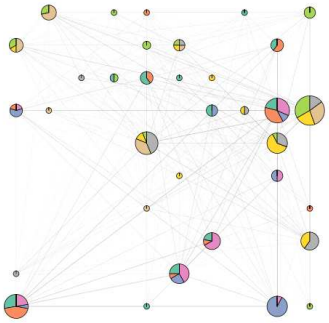
α evolution



multiple rSOM



rSOM
with numeric variables and factor



(simple) rSOM
with all three data sets

Figure 10: **Summary of the experiment**: the original graph and the original distribution of the numerical variables is given at the top of the figure; multiple rSOM results (second row) with the evolution of the α and the resulting map (disks have an area proportional to the number of observations and are colored according to the distribution of the original classes in the corresponding neuron); bottom: maps obtained using numeric variables and factor merged in a single data set and a simple Euclidean distance (left) and using all three data sets but a simple Euclidean distance.

4.4. Multiple relational SOM on real data

The last example illustrates multiple relational SOM on data related to school-to-work transitions. We used the data in the survey "Generation 98"[7]. According to the French National Institute of Statistics (INSEE), 22.7% of young people under 25 were unemployed at the end of the first semester 2012.[8] Hence, it is crucial to understand how the transition from school to employment or unemployment is achieved, in the current economic context. The data set contains information on 16 040 young people having graduated in 1998 and monitored during 94 months after having left school. The labor-market statuses have nine categories, labeled as follows: permanent-labor contract, fixed-term contract, apprenticeship contract, public temporary-labor contract, on-call contract, unemployed, inactive, military service, education. The following stylized facts are highlighted by a first descriptive analysis of the data as shown in Figure 11:

- permanent-labor contracts represent more than 20% of all statuses after one year and their ratio continues to increase until 50% after three years and almost 75% after seven years;
- the ratio of fixed-terms contracts is more than 20% after one year on the labor market, but it is decreasing to 15% after three years and then seems to converge to 8%;
- almost 30% of the young graduates are unemployed after one year. This ratio is decreasing and becomes constant, 10%, after the fourth year.

The dissimilarities between sequences were computed using optimal matching (OM). Also known as "edit distance" or "Levenshtein distance", optimal matching was first introduced in biology by [31] and used for aligning and comparing sequences. In social sciences, the first applications are due to [32]. The underlying idea of optimal matching is to transform one sequence into another using three possible operations: insertion, deletion and substitution. A cost is associated to each of the three operations. The dissimilarity between

[7] available thanks to Génération 1998 á 7 ans - 2005, [producer] CEREQ, [diffusion] Centre Maurice Halbwachs (CMH)

[8] All computations were performed with the free statistical software environment **R** (`http://cran.r-project.org/`, [25]). The graphical illustrations were carried out using the `TraMineR` package [63].

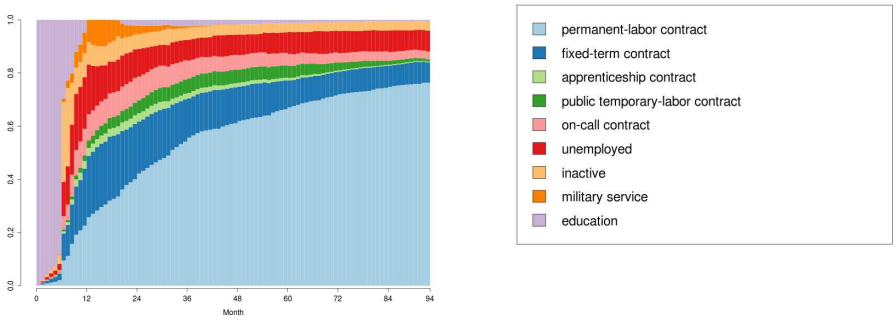


Figure 11: Labor market structure

sequences is then computed as the cost associated to the smallest number of operations which allows to transform the sequences into each other. The method seems simple and relatively intuitive, but the choice of the costs is a delicate operation in social sciences. This topic is subject to lively debates in the literature [4, 5] mostly because of the difficulties to establish an explicit and sound theoretical frame.

In our application, all career paths have the same length, the status of the graduate students being observed during 94 months. Hence, we suppose that there are no insertions or deletions and that only the substitution costs have to be defined for OM metrics. Among optimal-matching dissimilarities, we selected three dissimilarities: the OM with substitution costs computed from the transition matrix between statuses as proposed in [64], the Hamming dissimilarity (HAM, no insertion or deletion costs and a substitution cost equal to 1) and the Dynamic Hamming dissimilarity (DHD as described in [65]).

In order to identify the role of the different dissimilarities in extracting typologies, we considered several samples drawn at random from the data. For each of the experiments below, 50 samples containing 1 000 input sequences each were considered. In order to assess the quality of the maps, two indexes were computed: the quantization error for quantifying the quality of the clustering and the topographic error for quantifying the quality of the mapping, [66]. These quality criteria all depend on the dissimilarities used to train the map but the results are made comparable by using normalized dissimilarities.

Metric	OM	HAM	DHD
α-Mean	0.43111	0.28459	0.28429
α-Std	0.02912	0.01464	0.01523

Metric	OM	HAM	DHD	Optimally-tuned α
Quantization error	92.93672	121.67305	121.05520	114.84431
Topographic error	0.07390	0.08806	0.08124	0.05268

Table 3: Preliminary results for three OM metrics (average over 50 random subsamples): Optimally-tuned α (top table) and Quality criteria for the SOM clustering (bottom table).

The results are listed in Table 3. According to the mean values of the α's, the three dissimilarities contributed to extracting typologies. The Hamming and the dynamical Hamming dissimilarities have similar weights, while the OM with cost-matrix defined from the transition matrix has the largest weight. The mean quantization error computed on the maps trained with the three dissimilarities optimally combined is larger than the quantization error computed on the map trained with the OM metric only. On the other hand, the topographic error is improved in the mixed case. In this case, the joint use of the three dissimilarities provides a trade-off between the quality of the clustering and the quality of the mapping. The results confirm the difficulty to define adequate costs in optimal matching and the fact that the metric has to be chosen according to the aim of the study: building typologies (clustering) or visualizing data (mapping).

Finally, multiple rSOM was trained on the entire data set. The final map is illustrated in Figure 12. Several typologies emerge from the map: a fast access to permanent contracts (clear blue), a transition through fixed-term contracts before obtaining stable ones (dark and then clear blue), a holding on precarious jobs (dark blue), a public temporary contract (dark green) or an on-call (pink) contract ending at the end by a stable one, a long period of inactivity (yellow) or unemployment (red) with a gradual return to employment. The mapping also shows a progressive transition between trajectories of exclusion on the west and quick integration on the east. A more detailed study of this data set is available in [67].

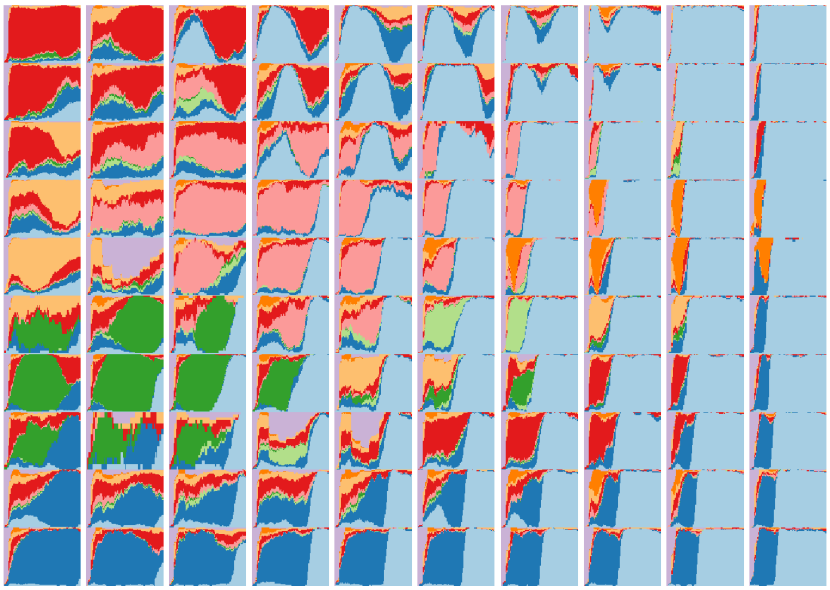


Figure 12: Final map obtained with the OM dissimilarities

5. Conclusion

An on-line version of relational SOM is introduced in this paper. It combines the standard advantage of the stochastic version of SOM (better organization) with relational SOM, which is able to handle data described by dissimilarities. This approach is extended to the case where several dissimilarities are available for the initial data set. Online multiple relational SOM handles several dissimilarities by combining them in an optimal fashion. The algorithm shows good performances, compared to alternative methods, in projecting data described by numerical variables, by categorical variables or by relations and is helpful to understand which dissimilarity is the most relevant when several ones are available. However, in its multiple dissimilarity version, the main drawback of the proposed relational SOM algorithm is related to the computation time: a sparse version should be investigated to allow us to handle very large data sets.

6. Acknowledgements

We thank the anonymous referees for their thorough comments and suggestions and for valuable hints on interesting references.

References

[1] N. Villa-Vialaneix, M. Olteanu, C. Cierco-Ayrolles, Carte auto-organisatrice pour graphes étiquetés, in: Actes des Ateliers FGG (Fouille de Grands Graphes), colloque EGC (Extraction et Gestion de Connaissances), Toulouse, France, 2013.

[2] M. Olteanu, N. Villa-Vialaneix, C. Cierco-Ayrolles, Multiple kernel self-organizing maps, in: M. Verleysen (Ed.), XXIst European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), d-side publications, Bruges, Belgium, 2013, pp. 83–88.

[3] E. Pękalska, R. Duin, The Dissimilarity Representation for Pattern Recognition. Foundations and Applications, World Scientific, 2005.

[4] A. Abbott, A. Tsay, Sequence analysis and optimal matching methods in sociology, Review and Prospect. Sociological Methods and Research 29 (2000) 3–33.

[5] L. Wu, Some comments on "Sequence analysis and optimal matching methods in sociology, review and prospect", Sociological Methods and Research 29 (2000) 41–64.

[6] M. Cottrell, M. Olteanu, F. Rossi, J. Rynkiewicz, N. Villa-Vialaneix, Neural networks for complex data, Künstliche Intelligenz 26 (2012) 1–8.

[7] M. Cottrell, P. Letrémy, How to use the Kohonen algorithm to simultaneously analyse individuals in a survey, Neurocomputing 63 (2005) 193–207.

[8] T. Kohohen, P. Somervuo, Self-organizing maps of symbol strings, Neurocomputing 21 (1998) 19–30.

[9] B. Conan-Guez, F. Rossi, A. El Golli, Fast algorithm and implementation of dissimilarity self-organizing maps, Neural Networks 19 (2006) 855–863.

[10] D. Mac Donald, C. Fyfe, The kernel self organising map., in: Proceedings of 4th International Conference on knowledge-based intelligence engineering systems and applied technologies, 2000, pp. 317–320.

[11] P. Andras, Kernel-Kohonen networks, International Journal of Neural Systems 12 (2002) 117–135.

[12] N. Villa, F. Rossi, A comparison between dissimilarity SOM and kernel SOM for clustering the vertices of a graph, in: 6th International Workshop on Self-Organizing Maps (WSOM), Neuroinformatics Group, Bielefield University, Bielefield, Germany, 2007. doi:10.2390/biecoll-wsom2007-139.

[13] R. Boulet, B. Jouve, F. Rossi, N. Villa, Batch kernel SOM and related laplacian methods for social network analysis, Neurocomputing 71 (2008) 1257–1273.

[14] T. Gärtner, Kernel for Structured Data, volume 72 of *Series in Machine Perception and Artificial Intelligence*, World Scientific, 2008.

[15] I. Schoenberg, Remarks to Maurice Fréchet's article "Sur la définition axiomatique d'une classe d'espace distanciés vectoriellement applicable sur l'espace de Hilbert", Annals of Mathematics 36 (1935) 724–732.

[16] G. Young, A. Householder, Discussion of a set of points in terms of their mutual distances, Psychometrika 3 (1938) 19–22.

[17] N. Krislock, H. Wolkowicz, Handbook on Semidefinite, Conic and Polynomial Optimization, volume 166 of *International Series in Operations Research & Management Science*, Springer, 2012, pp. 879–914.

[18] B. Hammer, A. Hasenfuss, F. Rossi, M. Strickert, Topographic processing of relational data, in: B. U. Neuroinformatics Group (Ed.), Proceedings of the 6th Workshop on Self-Organizing Maps (WSOM 07), Bielefeld, Germany, 2007.

[19] F. Rossi, A. Hasenfuss, B. Hammer, Accelerating relational clustering algorithms with sparse prototype representation, in: Proceedings of the 6th Workshop on Self-Organizing Maps (WSOM 07), Neuroinformatics Group, Bielefield University, Bielefield, Germany, 2007.

[20] B. Hammer, A. Hasenfuss, Topographic mapping of large dissimilarity data sets, Neural Computation 22 (2010) 2229–2284.

[21] B. Hammer, A. Gisbrecht, A. Hasenfuss, B. Mokbel, F. Schleif, X. Zhu, Topographic mapping of dissimilarity data, in: J. Laaksonen, T. Honkela (Eds.), Advances in Self-Organizing Maps (Proceedings of the 8th Workshop on Self-Organizing Maps, WSOM 2011), volume 6731 of *Lecture Notes in Computer Science*, Springer, Espoo, Finland, 2011, pp. 1–15.

[22] L. Goldfarb, A unified approach to pattern recognition, Pattern Recognition 17 (1984) 575–582.

[23] J. Fort, P. Letremy, M. Cottrell, Advantages and drawbacks of the batch kohonen algorithm, in: M. Verleysen (Ed.), Proceedings of 10th European Symposium on Artificial Neural Networks (ESANN 2002), Bruges, Belgium, 2002, pp. 223–230.

[24] M. Olteanu, N. Villa-Vialaneix, M. Cottrell, On-line relational som for dissimilarity data, in: P. Estevez, J. Principe, P. Zegers, G. Barreto (Eds.), Advances in Self-Organizing Maps (Proceedings of WSOM 2012), volume 198 of *AISC (Advances in Intelligent Systems and Computing)*, Springer Verlag, Berlin, Heidelberg, Santiago, Chile, 2012, pp. 13–22. doi:10.1007/978-3-642-35230-0_2.

[25] R Development Core Team, R: A Language and Environment for Statistical Computing, Vienna, Austria, 2012. URL: http://www.R-project.org, ISBN 3-900051-07-0.

[26] S. Theußl, A. Zeileis, Collaborative software development using R-Forge, The R Journal 1 (2009) 9–14. URL: http://journal.R-project.org/.

[27] RStudio and Inc., shiny: Web Application Framework for R, 2013. URL: http://CRAN.R-project.org/package=shiny, R package version 0.6.0.

[28] T. Heskes, Energy functions for self-organizing maps, in: E. Oja, S. Kaski (Eds.), Kohonen Maps, Elsevier, Amsterdam, 1999, pp. 303–315. URL: http://www.snn.ru.nl/reports/Heskes.wsom.ps.gz.

[29] A. El Golli, F. Rossi, B. Conan-Guez, Y. Lechevallier, Une adaptation des cartes auto-organisatrices pour des données décrites par un tableau de dissimilarités, Revue de Statistique Appliquée LIV (2006) 33–64.

[30] J. Lee, M. Verleysen, Nonlinear Dimensionality Reduction, Information Science and Statistics, Springer, New York; London, 2007. URL: http://www.worldcat.org/search?qt=worldcat_org_all&q=9780387393506.

[31] S. Needleman, C. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, Journal of Molecular Biology 48 (1970) 443–453.

[32] A. Abbott, J. Forrest, Optimal matching methods for historical sequences, Journal of Interdisciplinary History 16 (1986) 471–494.

[33] Y. Chen, E. Garcia, M. Gupta, A. Rahimi, L. Cazzanti, Similarity-based classification: concepts and algorithm, Journal of Machine Learning Research 10 (2009) 747–776.

[34] M. Cottrell, J. Fort, G. Pagès, Theoretical aspects of the SOM algorithm, Neurocomputing 21 (1998) 119–138.

[35] G. Polzlbauer, Survey and comparison of quality measures for self-organizing maps, in: J. Paralic, G. Polzlbauer, A. Rauber (Eds.), Proceedings of the Fifth Workshop on Data Analysis (WDA'04), Elfa Academic Press, Sliezsky dom, Vysoke Tatry, Slovakia, 2004, pp. 67–82.

[36] M. Verbanck, S. Le, J. Pagès, A new unsupervised gene clustering algorithm based on the integration of biological knowledge into expression data, BMC Bioinformatics 14 (2013) 42.

[37] D. Combe, C. Largeron, E. Egyed-Zsigmond, M. Géry, Getting clusters from structure data and attribute data, in: Proceedings of International Conference on Advances in Social Networks Analysis and Mining (IEEE/ACM), 2012, pp. 731–733.

[38] P. Labute, Quasar-cluster: a different view of molecular clustering, Technical Report, Chemical Computing Group, Inc., 1998. Http://www.chemcomp.com/journal/cluster.htm.

[39] G. Reza, M. Nasir, I. Hamidah, N. Norwti, A survey: clustering ensembles techniques, in: Proceedings of World Academy of Science, Engineering and Technology, volume 38, 2009, pp. 644–653.

[40] G. Cleuziou, M. Exbrayat, L. Martin, J. Sublemontier, CoFKM: a centralized method for multi-view clustering, in: Proceedings of International Conference on Data Mining, 2009.

[41] B. Zhao, J. Kwok, C. Zhang, Multiple kernel clustering, in: Proceedings of the 9th SIAM International Conference on Data Mining (SDM), Sparks, Nevada, USA, 2009.

[42] J. Zhuang, J. Wang, S. Hoi, X. Lan, Unsupervised multiple kernel clustering, Journal of Machine Learning Research: Workshop and Conference Proceedings 20 (2011) 129–144.

[43] X. Chen, Y. Ye, X. Xu, J. Huang, A feature group weighting method for subspace clustering of high-dimensional data, Pattern Recognition 45 (2012) 434–446.

[44] M. Lebbah, A. Chazottes, F. Badran, S. Thiria, Mixed topological map, in: M. Verleysen (Ed.), Proceedings of the 13th European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 2005, pp. 357–362.

[45] A. Rakotomamonjy, F. Bach, S. Canu, Y. Grandvalet, SimpleMKL, Journal of Machine Learning Research 9 (2008) 2491–2521.

[46] T. Villmann, H. Sven, M. Kästner, Gradient based learning in vector quantization using differentiable kernels, in: P. Estevez, J. Principe, P. Zegers, G. Barreto (Eds.), Advances in Self-Organizing Maps (Proceedings of WSOM 2012), volume 198 of *AISC (Advances in Intelligent Systems and Computing)*, Santiago, Chile, 2012, pp. 193–204.

[47] D. Luenberger, Linear and Nonlinear Programming, Addison-Wesley, 1984.

[48] F. Bonnans, Optimisation Continue, Dunod, 2006.

[49] P. Hebert, E. Penton, J. Burns, D. Janzen, W. Hallwachs, Ten species in one: DNA barcoding reveals cryptic species in the neotropical skipper butterfly astraptes fulgerator, Genetic Analysis 101 (2004) 14812–14817.

[50] R. de Salle, M. Egan, M. Siddal, The unholy trinity: taxonomy, species delimitation and DNA barcoding, Philosophical Transactions of the Royal Society B-Biological Sciences 360 (2005) 1905–1916.

[51] N. Saitou, M. Nei, The neighbor-joining method: a new method for reconstructing phylogenetic trees, Molecular Biology and Evolution 4 (1987) 406–425.

[52] M. Kimura, A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences, Journal of Molecular Evolution 16 (1980) 111–120.

[53] M. Olteanu, V. Nicolas, B. Schaeffer, C. Denys, A. Missoup, J. Kennis, C. Larédo, Nonlinear projection methods for visualizing barcode data and application on two data sets, Molecular Ecology Resources (2013) n/a–n/a.

[54] L. Adamic, N. Glance, The political blogosphere and the 2004 us election: divided they blog, in: Proceedings of the 3rd LINKDD Workshop, ACM Press, New York, NY, USA, 2005, pp. 36–43.

[55] T. Fruchterman, B. Reingold, Graph drawing by force-directed placement, Software, Practice and Experience 21 (1991) 1129–1164.

[56] R. Kondor, J. Lafferty, Diffusion kernels on graphs and other discrete structures, in: Proceedings of the 19*th* International Conference on Machine Learning, 2002, pp. 315–322.

[57] F. Fouss, A. Pirotte, J. Renders, M. Saerens, Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation, IEEE Transactions on Knowledge and Data Engineering 19 (2007) 355–369.

[58] M. Newman, M. Girvan, Finding and evaluating community structure in networks, Physical Review, E 69 (2004) 026113.

[59] A. Clauset, M. E. J. Newman, C. Moore, Finding community structure in very large networks, Physical Review, E 70 (2004) 066111.

[60] A. Condon, R. Karp, Algorithms for graph partitioning on the planted partition model, Random Structures and Algorithms 18 (2001) 116–140.

[61] B. Caputo, K. Sim, F. Furesjo, A. Smola, Appearance-based object recognition using SVMs: which kernel should I use?, in: Proceedings of NIPS workshop on Statistical methods for computational experiments in visual processing and computer vision, Whistler, 2002.

[62] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, Journal of Statistical Mechanics (2005) P09008.

[63] A. Gabadinho, G. Ritschard, N. Müller, M. Studer, Analyzing and visualizing state sequences in R with TraMineR, Journal of Statistical Software 40 (2011) 1–37.

[64] N. Müller, G. Ritschard, M. Studer, A. Gabadinho, Extracting knowledge from life courses: clustering and visualization, in: 10th International Conference DaWaK (Data Warehousing and Knowledge Discovery), volume 5182 of *Lecture Notes in Computer Science*, Berlin: Springer, Turin, Italy, 2008, pp. 176–185.

[65] L. Lesnard, Setting cost in optimal matching to uncover contemporaneous socio-temporal patterns, Sociological Methods et Research 38 (2010) 389–419.

[66] G. Pölzlbauer, Survey and comparison of quality measures for self-organizing maps, in: Proceedings of the Fifth Workshop on Data Analysis (WDA'04), Elfa Academic Press, 2004, pp. 67–82.

[67] S. Massoni, M. Olteanu, N. Villa-Vialaneix, Which distance use when extracting typologies in sequence analysis? An application to school to work transitions, in: International Work Conference on Artificial Neural Networks (IWANN 2013), Puerto de la Cruz, Tenerife, 2013.

Appendix A. Pseudo-euclidean framework and justification of Equation 2

The proof below can be derived directly from Theorem 1 of [20]. It is given in details here, for the sake of clarity.

As explained in [22, 3], if δ is a symmetric dissimilarity matrix, then, there exists two Euclidean spaces $\mathcal{E}$ and $\mathcal{F}$, with positive definite scalar products, and a mapping $\phi : x \in \mathcal{G} \to (\phi|_{\mathcal{E}}(x), \phi|_{\mathcal{F}}(x)) \in \mathcal{E} \otimes \mathcal{F}$ such that

$$\delta(x_i, x_j) = \|\phi|_{\mathcal{E}}(x_i) - \phi|_{\mathcal{E}}(x_j)\|^2_{\mathcal{E}} - \|\phi|_{\mathcal{F}}(x_i) - \phi|_{\mathcal{F}}(x_j)\|^2_{\mathcal{F}}. \tag{A.1}$$

Hence, supposing that p_u can be written as $p_u = \sum_i \beta_{ui}(\phi|_{\mathcal{E}}(x_i), \phi|_{\mathcal{F}}(x_i))$ (which, in the text of the article is written $\sum_i \beta_{ui} x_i$ for the sake of simplicity),

then the right hand side of Equation (2) can be re-written as:

$$\begin{aligned}\boldsymbol{\Delta}_i\beta_u - \frac{1}{2}\beta_u^T\boldsymbol{\Delta}\beta_u &= \sum_l \beta_{ul}\delta(x_l,x_i) - \frac{1}{2}\sum_{ll'}\beta_{ul}\beta_{ul'}\delta(x_l,x_{l'}) \qquad (A.2)\\ &= \Bigg[\sum_l \beta_{ul}\|\phi|_{\mathcal{E}}(x_i) - \phi|_{\mathcal{E}}(x_l)\|^2_{\mathcal{E}} \\ &\qquad -\frac{1}{2}\sum_{ll'}\beta_{ul}\beta_{ul'}\|\phi|_{\mathcal{E}}(x_l) - \phi|_{\mathcal{E}}(x_{l'})\|^2_{\mathcal{E}}\Bigg] - \\ &\qquad \Bigg[\sum_l \beta_{ul}\|\phi|_{\mathcal{F}}(x_i) - \phi|_{\mathcal{F}}(x_l)\|^2_{\mathcal{E}} \\ &\qquad -\frac{1}{2}\sum_{ll'}\beta_{ul}\beta_{ul'}\|\phi|_{\mathcal{F}}(x_l) - \phi|_{\mathcal{F}}(x_{l'})\|^2_{\mathcal{F}}.\Bigg]\end{aligned}$$

But, using that $\sum_l \beta_{ul} = 1$, we obtain

$$\begin{aligned}\Bigg[\sum_l \beta_{ul}\|\phi|_{\mathcal{E}}(x_i) - \phi|_{\mathcal{E}}(x_l)\|^2_{\mathcal{E}} - \frac{1}{2}\sum_{ll'}\beta_{ul}\beta_{ul'}\|\phi|_{\mathcal{E}}(x_l) - \phi|_{\mathcal{E}}(x_{l'})\|^2_{\mathcal{E}}\Bigg] &= \\ \|\phi|_{\mathcal{E}}(x_i)\|^2_{\mathcal{E}} - 2\sum_l \beta_{ul}\langle\phi|_{\mathcal{E}}(x_i),\phi|_{\mathcal{E}}(x_l)\rangle_{\mathcal{E}} + \sum_l \beta_{ul}\|\phi|_{\mathcal{E}}(x_l)\|^2_{\mathcal{E}} &+ \\ -\frac{1}{2}\sum_l \beta_{ul}\|\phi|_{\mathcal{E}}(x_l)\|^2_{\mathcal{E}} - \frac{1}{2}\sum_l \beta_{ul}\|\phi|_{\mathcal{E}}(x_l)\|^2_{\mathcal{E}} + \sum_l\sum_{l'}\beta_{ul}\beta_{ul'}\langle\phi|_{\mathcal{E}}(x_l),\phi|_{\mathcal{E}}(x_{l'})\rangle_{\mathcal{E}} &= \\ \|\phi|_{\mathcal{E}}(x_i) - \sum_l \beta_{ul}\phi|_{\mathcal{E}}(x_l)\|^2_{\mathcal{E}},\end{aligned}$$

which, injected into Equation (A.2), gives

$$\begin{aligned}\boldsymbol{\Delta}_i\beta_u - \frac{1}{2}\beta_u^T\boldsymbol{\Delta}\beta_u &= \|\phi|_{\mathcal{E}}(x_i) - \sum_l \beta_{ul}\phi|_{\mathcal{E}}(x_l)\|^2_{\mathcal{E}} - \\ &\qquad \|\phi|_{\mathcal{F}}(x_i) - \sum_l \beta_{ul}\phi|_{\mathcal{F}}(x_l)\|^2_{\mathcal{F}}\end{aligned}$$

which is the distance, in $\mathcal{E}\otimes\mathcal{F}$, induced by the pseudo-norm defined in Equation (A.1), between $(\phi_{\mathcal{E}}(x_i),\phi_{\mathcal{F}}(x_i))$ and p_u.□

Appendix B. Proof that $\sum_l \beta_{ul}^t = 1$ at any step t of the algorithm

We prove here that $\sum_l \beta_{ul}^t = 1$, $\forall t \geq 0$. Noting that the property is verified for $t = 0$, let us suppose that for a given t, we have $\sum_l \beta_{ul}^t = 1$. Then,

$$\beta_{ul}^{t+1} = \begin{cases} \beta_{ul}^t + \mu(t) H^t(d(f^t(x_i), u))(1 - \beta_{ul}^t) & \text{if } i = l \\ \beta_{ul}^t - \mu(t) H^t(d(f^t(x_i), u))\beta_{ul}^t & \text{otherwise.} \end{cases}$$

and thus, using $\sum_l \beta_{ul}^t = 1$, we have

$$\begin{aligned} \sum_l \beta_{ul}^{t+1} &= \sum_l \beta_{ul}^t + \mu(t) H^t(d(f^t(x_i), u)) - \mu(t) H^t(d(f^t(x_i), u)) \sum_{ul}^t \beta_{ul}^t \\ &= 1 + \mu(t) H^t(d(f^t(x_i), u)) - \mu(t) H^t(d(f^t(x_i), u)) = 1. \square \end{aligned}$$

Appendix C. Equivalence between relational SOM, kernel SOM and standard SOM

This appendix shows that

1. if (x_i) take values in a Euclidean space and if the dissimilarity δ is the Euclidean distance in this space, then the on-line version of relational SOM as presented in Algorithm 1 is exactly equivalent to the on-line version of the standard SOM in this space;
2. if the dissimilarity δ is computed from a kernel K by Equation (3), then the on-line version of relational SOM is exactly equivalent to the on-line version of the kernel SOM, as described in [12].

Let us first prove the first part of the assertion: if the prototypes are initialized in the convex hull of (x_i) then, they can all be written $p_u^0 = \sum_i \beta_{ui}^0 x_i$. As already demonstrated in Appendix A, the assignment step of the on-line relational SOM minimizes $\mathbf{\Delta}_i \beta_u - \frac{1}{2} \beta_u^T \mathbf{\Delta} \beta_u$ which is equal to the squared distance between x_i and p_u in the Euclidean space and proves that the assignment step is identical to the one of the standard SOM.

Then, on-line relational SOM updates the β_{ui}^t by

$$\beta_u^t = \beta_u^{t-1} + \mu(t) H^t(d(f^t(x_i), u)) \left(\mathbf{1}_i - \beta_u^{t-1} \right).$$

Multiplying each β_{ul}^t by x_l gives

$$x_l \beta_{ul}^t = \begin{cases} x_l \beta_{ul}^{t-1} \left(1 - \mu(t) H^t(d(f^t(x_i), u))\right) & \text{if } l \neq i \\ x_i \beta_{ui}^{t-1} + \mu(t) H^t(d(f^t(x_i), u)) \left(x_i - \beta_{ui}^{t-1}\right) & \text{if } l = i \end{cases},$$

and, by summing over l, leads to

$$\sum_l \beta_{ul}^t x_l = \sum_l \beta_{ul}^{t-1} x_l + \mu(t) H^t(d(f^t(x_i), u)) \left(x_i - \sum_l \beta_{ul}^{t-1} x_l \right)$$

with $p_{ul}^t = \sum_l x_l \beta_{ul}^{t-1}$ the representation step in the on-line relational SOM is thus

$$p_u^t = p_u^{t-1} + \mu(t) H^t(d(f^t(x_i), u)) \left(x_i - p_u^{t-1} \right),$$

which is in the convex hull of (x_i) as long as p_u^{t-1} already is, as shown in Appendix B. This is also the representation step of the standard on-line SOM.

Then, the equivalence between kernel SOM and relational SOM follows straightforwardly since kernel SOM is equivalent to standard SOM in the RKHS induced by the kernel and that the square distance in this space is given by Equation (3).□