Multiple kernel self-organizing maps
Madalina Olteanu, Nathalie N. Villa-Vialaneix, Christine Cierco-Ayrolles

To cite this version:
Madalina Olteanu, Nathalie N. Villa-Vialaneix, Christine Cierco-Ayrolles. Multiple kernel self-organizing maps. 21. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2013), Apr 2013, Bruges, Belgium. hal-02747111

HAL Id: hal-02747111
https://hal.inrae.fr/hal-02747111
Submitted on 3 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Multiple Kernel Self-Organizing Maps

Madalina Olteanu¹, Nathalie Villa-Vialaneix² and Christine Cierco-Ayrolles²

1- SAMM, Université Paris 1, Paris, France
2- Unité BIA, INRA de Toulouse, Auzeville, France

Abstract. In a number of real-life applications, the user is interested in analyzing several sources of information together: a graph combined with the additional information known on its nodes, numerical variables measured on individuals and factors describing these individuals... The combination of all sources of information can help him to understand the dataset in its whole better. The present article focuses on such an issue, by using self-organizing maps. The use a kernel version of the algorithm allows us to combine various types of information and automatically tune the data combination. This approach is illustrated on a simulated example.

1 Introduction

In a number of real-life applications, the user is interested in analyzing several sources of information together: a graph combined with the additional information known on its nodes, numerical variables measured on individuals and factors describing these individuals... The combination of all sources of information can help him to understand the dataset in its whole better. The present article focuses on such an issue, by using self-organizing maps (SOM).

SOM has already been extended to the framework of non numerical data, using various approaches. Historically, median SOM, which consists in replacing the standard computation of the prototypes by an approximation in the original dataset, was introduced first. This principle was used to extend SOM to dissimilarity data in [1]. Several variants of the original algorithm have been proposed since: [2, 3, 4, 5] extended the framework of SOM for dissimilarity data by proposing a more flexible representation for the prototypes, either in batch or in online versions. A closely related approach is to rely on kernels to map the original data into an (implicit) Euclidean space where the standard SOM can be used [6, 7, 8]. Several kernels have been designed to handle complex data such as strings, nodes in a graph or several graphs [9].

In the present paper, we propose to extend kernel SOM by using a combination of kernels, each of them dedicated to a particular aspect of the entire dataset. The methodology is described in Section 2 and illustrated in Section 3.

2 Method

The purpose of the present paper is to design a self-organizing map algorithm in which data coming from various sources can be used as inputs in an optimal combination. Let us suppose that we are given $D$ sets of input data,
Taking values in arbitrary spaces \((G_d)_d\). For example, \((x_i^d)\) may represent \(p\)-dimensional numerical vectors or factor (non-numerical) variables describing the \(n\) individuals. More specifically, \((x_i^d)\) may describe the \(n\) nodes in a graph and the kind of relation between them or some text variables naming or describing the data.

**Multiple kernel SOM**

Suppose also that for each of these \(D\) datasets, a kernel is known, i.e., a function \(K_d : G_d \times G_d \rightarrow \mathbb{R}\) such that \(K_d\) is symmetric \((\forall z, z' \in G_d, K_d(z, z') = K_d(z', z))\) and positive \((\forall N \in \mathbb{N}, \forall (z_j)_{j=1}^N \subset G_d\) and \(\forall (\alpha_j)_{j=1}^N \subset \mathbb{R}, \sum_j \alpha_j \alpha_j K_d(z_j, z'_j) \geq 0\). For example, the linear kernel, the Gaussian kernel or the polynomial kernel, among others, may be used for numerical variables. For nodes of a graph, the heat kernel [10], the commute time kernel [11] or any other regularized version of the Laplacian can be used as kernels [12], while for textual variables, string kernels, based on the number of occurrences of common substrings [13] may be considered.\(^1\)

From these \(D\) kernels, a new kernel can be defined by computing a convex combination: \(\Phi(x_i, x'_i) = \sum_{d=1}^D \alpha_d K_d(x_i^d, x'_i^d)\), where \(\alpha_d \in [0, 1]\), \(\sum_{d} \alpha_{d} = 1\) and \(x_i = (x_i^1, \ldots, x_i^n)\) \(\in G = G_1 \times \ldots \times G_d\). Obviously, such an application is also symmetric and positive and thus [14] proves that there is a Hilbert space, \((\mathcal{H}, \langle \cdot, \cdot \rangle)\) (called the feature space) and an application \(\phi : G \rightarrow \mathcal{H}\) (called the feature map) such that \(\Phi(x_i, x'_i) = \langle \phi(x_i), \phi(x'_i) \rangle\).

In the SOM algorithm, the individuals \(i = 1, \ldots, n\) have to be clustered into a low dimensional grid made of \(M\) neurons, \(\{1, \ldots, M\}\). These neurons are related to each other by a neighborhood relationship, \(h\). Each neuron is also represented by prototypes, \((p_j)\), \(j = 1, \ldots, M\) taking values in the previously defined feature space \(\mathcal{H}\). Following the general framework of kernel SOM described in [6, 15, 8], the prototypes are written as a convex combination of the input data in the feature space \(\mathcal{H}\): \(p_j = \sum_l \gamma_{jl} \phi(x_l)\). The (squared-)distance between an individual \(x_i\) and a prototype \(p_j\) is then computed by

\[
\|p_j - \phi(x_i)\|^2 = K(x_i, x_i) - 2 \sum_{l=1}^n \gamma_{jl} K(x_l, x_i) + \sum_{l, l'=1}^n \gamma_{jl} \gamma_{j'l'} K(x_l, x_{l'}) \tag{1}
\]

**Online multiple kernel SOM**

If the \((\alpha_d)\) are given, the standard kernel SOM aims at optimizing (over \((\gamma_{ji})_{ji}\)) the following energy function:

\[
E((\gamma_{ji})_{ji}, (\alpha_d)_d) = \sum_{i=1}^n \sum_{j=1}^M \rho(f(x_i), j) \|\phi^\alpha(x_i) - p_j^\alpha(\gamma_{ji})\|_\alpha^2,
\]

where \(f(x_i) \in \{1, \ldots, M\}\) is the index of the cluster to which \(x_i\) is assigned. The notations \(\phi^\alpha, p_j^\alpha\) and \(\|\cdot\|_\alpha\) are used to emphasize that these quantities depend

\(^1\)Note that this methodology can also be applied to combine several kernels on the same data (e.g., different kinds of string kernels, encapsulating different features of the texts).
on $\alpha$. The novelty of this paper is to include the optimization of the convex combination of kernels into the on-line algorithm which trains the map. More precisely, a stochastic gradient descent step is added to the original on-line SOM algorithm to optimize the energy $E((\gamma_{ji})_{ji}, (\alpha_d)_d)$ over both $(\gamma_{ji})_{ji}$ and $(\alpha_d)_d$.

To perform the stochastic gradient descent step on the $(\alpha_d)_d$, the computation of the derivative of $E|_{x_t} = \sum_{j=1}^M h(f(x_t), j) \|\phi^{\alpha}(x_t) - p_j^{\alpha}(\gamma_j)\|^2_\alpha$ (the contribution of the randomly chosen observation $x_t$ to the energy) with respect to $\alpha$ is needed. As $\partial \partial_{\alpha_d} [K(x_i, x_i')] = K_d(x_d^i, x_d^i)$, we have that

$$D_d := \frac{\partial E}{\partial \alpha_d} = \sum_{j=1}^M h(f(x_t), j) \left( K_d(x_d^j, x_d^j) - 2 \sum_{l=1}^n \gamma_{jl} K_d(x_d^l, x_d^j) + \sum_{l,l'=1}^n \gamma_{jl} \gamma_{jl'} K_d(x_d^l, x_d^{l'}) \right)$$

Following an idea similar to that of [16], the SOM is trained by performing, alternatively, the standard steps of the SOM algorithm (i.e., affectation and representation steps) and a gradient descent step for the $(\alpha_d)_d$. The full process is given in Algorithm 1.

**Algorithm 1** Multiple online kernel SOM

1. $\forall j = 1, \ldots, M$ and $\forall i = 1, \ldots, n$, initialize $\gamma_{ji}^0$ randomly in $\mathbb{R}$ such that $\gamma_{ji}^0 \geq 0$ and $\sum_{i=1}^n \gamma_{ji} = 1$ and $\forall d = 1, \ldots, D$, initialize $\alpha^0_d$ in $[0, 1]$ such that $\sum_{d} \alpha^0_d = 1$.
2. for $t = 1 \rightarrow T$ do
3. Randomly choose an input $x_t \in \{x_i\}_i$
4. Assignment step. Find the unit of the closest prototype
   $$f^t(x_t) \leftarrow \arg \min_{j=1, \ldots, M} \left\| \phi^{\alpha_{t-1}}(x_t) - p_j^{\alpha_{t-1}}(\gamma_{ji}^{t-1}) \right\|_{\alpha_{t-1}}$$
5. Representation step. Update all the prototypes: $\forall j = 1, \ldots, M$,
   $$\gamma_{ji}^t \leftarrow \gamma_{ji}^{t-1} + \mu(t) h(f^t(x_t), j) \left( \delta_{li} - \gamma_{ji}^{t-1} \right)$$
   with $\delta_{li} = 1$ if $l = i$ and 0 otherwise.
6. Gradient descent step. Update the kernel:
   $\forall d = 1, \ldots, D$, $\alpha_d^t \leftarrow \alpha_d^{t-1} + \nu(t) D_d^t$ and $K^t \leftarrow \sum_d \alpha_d^t K_d^t$.
7. end for

To ensure that the gradient step respects the constraints on $\alpha$ ($\alpha_d \geq 0$ and $\sum_d \alpha_d = 1$), the following strategy can be applied: first, the gradient $D_d^t = \left( \frac{\partial E|_{x_t}}{\partial \alpha_d} \right)_d$ is reduced and it is projected so that the non-negativity of $\alpha$ is
ensured (the reduced gradient projection procedure is described in [17, 18, 16]). This leads to the descent direction $D_d$. The descent step is decreased with the standard rate of $\nu_0/t$.

In practice, to ensure similar scales for all kernels, a normalization step is performed before computing the first convex combination: all kernels are scaled to have a unit norm, according to the Frobenius norm. The tuning procedure then automatically finds out the $(\alpha_d)$ that give the best map, i.e., that best organizes the observations according to the global kernel $K$. As illustrated in [19], this can be particularly useful when the data can be described by several kernels, all bringing up a different insight on the similarities between individuals, as it is often the case in social sciences (see [20, 21] for a discussion on this topic).

Finally, the additional cost of tuning the $(\alpha_d)$ is moderate because it is performed online: in practice, we simply multiplied by two the number of steps usually needed to train the map and this choice led to a converging algorithm.

3 Application

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 (numbered from 1 to 8 in the following), were generated using three different kinds of data: 1/ an unweighted graph, simulated similarly as described in [22]. 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; 2/ numerical data coming from two dimensional Gaussian distributions. The variables were simulated by Gaussian vectors with independent components having a variance equal to 0.3 and a mean equal to $(0,0)$ for the odd groups and to $(1,1)$ for the even groups; 3/ 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 datasets gave access to the eight original groups. The online multiple kernel SOM algorithm was applied to this problem with the commute time kernel for the graph and a Gaussian kernel for the numerical data and for the factor variable (re-coded in disjunctive form). The algorithm was compared with a standard kernel SOM approach using one of the three datasets only. It was also compared to kernel SOM using numerical and factor data or all the three datasets but used as if they were issued from the same dataset with a single Gaussian kernel (when the graph was added to the numerical and factor data, it was under the form of its adjacency matrix). The performance of the different approaches were compared by using the normalized mutual information (NMI) [23] compared to the original classes. This measure has values between 0 and 1 and is equal to 1 when the two partitions are identical. The results are given in Figure 1.

The multiple kernel SOM clearly outperformed all the other approaches, in-
Fig. 1: NMI (left) of online kernel SOM when using all datasets in the multiple kernel framework or only one dataset or all datasets in a single kernel framework. Example of a map resulting from the use of the algorithm (right): each node represents a neuron of the map and has an area proportional to the number of observations classified inside. The colors display the proportion of the eight original classes in the clusters.

including the one where all datasets are simply aggregated together. The example of the resulting map (given on the right hand side of the figure) shows a good classification and a good organization according to the three types of information: the eight groups are well distinguished by the algorithm.

4 Conclusion

This article proposes a way to combine multiple sources of heterogeneous information in self-organizing maps. Our approach can help to select the most relevant sources of information or the most relevant ways to describe similarities between observations. The computational cost of such an approach is moderate but can however be prohibitive when run with very large datasets or a large number of kernels. A sparse approach is currently under study for improving this aspect of the algorithm.

References


