Neurocomputing xx (xxxx) xxxx–xxxx

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Efficient interpretable variants of online SOM for large dissimilarity data

Jérôme Mariette[a,*], Madalina Olteanu[b], Nathalie Villa-Vialaneix[a,*]

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

ARTICLE INFO

Communicated by Y. Gu

Keywords:
SOM
Sparse methods
Kernel
Dissimilarity
K-PCA
Nyström

ABSTRACT

Self-organizing maps (SOM) are a useful tool for exploring data. In its original version, the SOM algorithm was designed for numerical vectors. Since then, several extensions have been proposed to handle complex datasets described by (dis)similarities. Most of these extensions represent prototypes by a list of (dis)similarities with the entire dataset and suffer from several drawbacks: their complexity is increased – it becomes quadratic instead of linear –, the stability is reduced and the interpretability of the prototypes is lost.

In the present article, we propose and compare two extensions of the stochastic SOM for (dis)similarity data: the first one takes advantage of the online setting in order to maintain a sparse representation of the prototypes at each step of the algorithm, while the second one uses a dimension reduction in a feature space defined by the (dis)similarity. Our contributions to the analysis of (dis)similarity data with topographic maps are thus twofolds: first, we present a new version of the SOM algorithm which ensures a sparse representation of the prototypes through online updates. Second, this approach is compared on several benchmarks to a standard dimension reduction technique (K-PCA), which is itself adapted to large datasets with the Nyström approximation.

Results demonstrate that both approaches lead to reduce the prototypes dimensionality while providing accurate results in a reasonable computational time. Selecting one of these two strategies depends on the dataset size, the need to easily interpret the results and the computational facilities available. The conclusion tries to provide some recommendations to help the user making this choice.

1. Introduction

1.1. State-of-the art on SOM for (dis)similarity data

Over the years, the self-organizing map (SOM) algorithm [1] was proved to be a powerful and convenient tool for clustering and visualizing data [2–6]. While the original algorithm had been designed for numerical vectors, the available data in the applications became more and more complex, being frequently too rich to be described by a fixed set of numerical attributes only. This is the case, for example, when data are described by relations between objects (individuals involved in a social network) or by measures of resemblance/dissemblance which are context specific (see [7,8] for similarities between categorical sequences, [9] for similarities between microbial diversity distributions, [10] for similarities in gene expression data).

During the past twenty years, the SOM algorithm was extended to handle non numerical data. For example, SOM was adapted to categorical data analysis, by using a method similar to Multiple Correspondence Analysis in [11]. Another solution, called median SOM [12], addressed the issue of data described by pairwise relations (similarities or dissimilarities): in this solution, the standard computation of the prototypes is replaced by an approximation within the original dataset. However, as prototypes are chosen among the data, their representation is very restrictive. In order to increase the flexibility of the prototypes, [13] proposed to represent a class by several prototypes, all chosen among the original dataset. But, this method seriously increases the computational time, while prototypes remain restricted to the original dataset and may generate possible sampling or sparsity issues.

A very different approach to handle relational data consists in relying on a (pseudo-)Euclidean framework, following the results of [14] (for data described by a kernel) or of [15] (for dissimilarity data). This approach was developed in the framework of kernel SOM (see [16] for the online version and [17] for the batch version), and in the framework of relational SOM (see [18] for the online version and [19] for the batch version). Kernel SOM and relational SOM are equivalent if the dissimilarity in relational SOM is the squared distance induced by the kernel. The key idea of this approach is to express prototypes as convex combinations of the images of the original data $(x_i)_{i=1,\dots,n}$ in a (pseudo-)Euclidean space in which the data are (implicitly) embedded

* Corresponding authors.

http://dx.doi.org/10.1016/j.neucom.2016.11.014
Received 5 April 2016; Received in revised form 22 September 2016; Accepted 11 November 2016
Available online xxxx
0925-2312/

by the kernel (or the dissimilarity): as stated in [20,21], this solution yields several drawbacks due to the large dimensionality of the embedding space (which is equal to the number of observations, n). Firstly, the complexity (in n) is strongly increased and becomes at least quadratic. As stressed in [19], algorithms will be slow for datasets with 10,000 observations and impossible to run on a normal computer for 100,000 input data. Secondly, the results are highly unstable: especially in the online (also called stochastic) version of the algorithm, two different runs of the method can provide very different results. Thirdly, one of the most important features of the SOM algorithm is lost: in standard numerical SOM, clusters are represented by a single prototype valued in the data space. These prototypes help to interpret the obtained clusters and thus the overall map organization. In kernel/relational SOM, prototypes are given as n coefficients that correspond to a resemblance with each of the n observations: they do not correspond themselves to an observation in the original data space and as such, prototypes are not much more informative than the clustering itself.

In conclusion, kernel and relational extensions of the standard SOM algorithm are hardly practicable when the dataset is large. This is due partly to the number of observations, but also to the dimensionality of the (embedded) data which is directly related to this number. To address this issue, strategies usually used to handle large datasets or datasets with a high dimensionality are useful and they can even be combined.

1.2. Review of methods for large datasets and high dimensional datasets

Different strategies were developed and are available in the literature to handle large datasets (when the number of observations is large) and high-dimensional datasets (when the dimension of the dataset is large). For large datasets, standard approaches include (i) *divide and conquer approaches* [22–24] in which data are split into several bits of data which are processed separately. The results are aggregated afterwards to obtain a final solution which is supposed to well approximate the solution that would have been obtained if the entire dataset had been processed at once; (ii) *subsampling methods* [25–29], which consist in using a restricted subset (usually carefully designed) of the original data, in order to approximate the solution that could have been obtained with the entire dataset and (iii) *online* updates [30,31], in which the results are updated with sequential steps, each having a low computational cost.

A particular case of the subsampling strategy is the Nyström approximation [32], which consists in sampling a small number of rows/columns in square matrices and in obtaining an approximation of its eigendecomposition at a very reduced computational cost. The eigendecomposition is even exact when the matrix is of low rank (when the size of the subsample is larger than the rank of the matrix). This method is frequently used for kernel and dissimilarity-based algorithms.

For high-dimensional data, the strategies are a bit different and include (i) *sparse approaches* [33,34], in which a subset of the variables is selected to build the final predictive model. This subset can be obtained from sequential exploration (stepwise strategies), from approximation heuristics or by using a sparse penalty term within the model (LASSO); (ii) *dimension reduction (DR)* techniques, that can be linear (PCA for instance or random projections as in [35]) or nonlinear [36]. DR methods embed the data in a small dimensional space and are usually mainly used for visualization and exploratory analysis. However, if the embedding can be obtained at a low cost, it can be used as an approximation of the high-dimensional dataset on which more costly algorithms may be applied. SOM itself is a dimension reduction method but, as stressed before, the computational complexity of its kernel and relational versions is high. Finally, a particular case of DR techniques is *model-based clustering* methods, which use mixture distributions and embed the data in a low-dimensional subspace that is the best suited for clustering (see [37] for a review).

1.3. Kernel/relational extensions for large datasets

Several extensions for kernel and relational data of the standard SOM algorithm, or of related kernel/relational algorithms (such as, *e.g.*, k-means, LVQ, topographic maps…) have already been proposed in the literature. They use ideas coming from the strategies handling large and/or high-dimensional datasets cited above. Most of them seek a simplified/sparse representation of the prototypes and/or a reduced computational time.

In the relational k-means framework [38], proposed a sparse extension of the batch algorithm: every prototype is represented by at most K (K fixed) observations by cluster, that are selected at each step of the algorithm. In the supervised framework [21], used a similar strategy for batch LVQ, by selecting the most representative observations (with different methods to obtain them, including approximation heuristics and L^1 penalty) in every cluster and at each step of the algorithm. A similar method was used in [39], combined with the Nyström approximation of the LVQ algorithm, in order to obtain sparse prototypes at very low computational cost. The Nyström approximation was also used for obtaining faster versions of topographic mapping methods [40,41] and for reducing the computational cost of the clustering. Another subsampling strategy was used in a nonlinear (kernel) DR framework to allow processing large datasets, in [42].

However, these approaches do not lead to a simplified (and thus interpretable) representation of the prototypes. Furthermore, all of them are restricted to the batch framework and most of them are performed after each iteration of a batch algorithm, *i.e.*, after all observations have been processed at least once. An alternative to these methods consists in splitting the data into several subsets on which independent algorithms are trained: in [19], the complexity is reduced using iterative "patch clustering" that mixes "divide and conquer" and "online updates" methods. First, the data are split into B patches of size n_B ($\ll n$, B fixed). A prototype-based clustering algorithm in batch version (neural gas or SOM) is then run on a patch $\mathcal{P}_l$. The resulting prototypes, which may be viewed as compressed representations of the data already seen, are then added as new data points to the next patch, $\mathcal{P}_{l+1}$. Moreover, the full vector of coefficients is replaced by the Q closest input data (Q fixed). With this method, linear time and constant space representation are obtained but the sequential training may influence the final result since all observations are not processed evenly.

In the same line of thoughts [43], propose a bagging approach for kernel SOM. Data are split into B subsamples of size n_B ($\ll n$, B fixed), the online kernel SOM is trained on each subsample and, after training, the most representative Q observations are chosen for each prototype (Q fixed). Eventually, a final map is trained on the resulting most representative observations. In this method, parallel computing techniques can be used for reducing the computational time. However, the results of the B trained SOMs are not used as such but only to select the most representative observations in the dataset.

1.4. Contributions of the article

In the present article, we propose and compare two methods to obtain sparse prototypes and a reduced computational cost in the online SOM algorithm. The first one uses a reduction dimension in the embedding space, which can be efficiently performed with Nyström technique. This method combines ideas coming from the high dimension and the large data problems. However, it is not specific to the online setting. We thus compared it with another approach which takes advantage of the online framework to provide sparse prototypes at all iteration steps of the algorithm: the coefficients are interpreted similarly to an amount of mass and the most important observations