Collaborative learning using topographic maps

Jérémie Sublime*,** Nistor Grozavu**, Guénaël Cabanes**,
Younès Bennani** and Antoine Cornuéjols*

*AgroParisTech, INRA UMR MIA 518
16 rue Claude Bernard, F-75231 Paris Cedex 5, France
jeremie.sublime@agroparistech.fr, antoine.cornuejols@agroparistech.fr,
**Université Paris 13 - Sorbonne Paris Cité
Laboratoire d'Informatique de Paris-Nord - CNRS UMR 7030
guenael.cabanes@lipn.univ-paris13.fr, younes.bennani@lipn.univ-paris13.fr

Résumé. Le clustering collaboratif est un domaine émergeant du machine learning à fort potentiel applicatif, ayant des similarités avec l'apprentissage par ensemble et l'apprentissage par transfert. Dans cette article, nous proposons une méthode permettant de combiner un framework collaboratif avec la structure des Cartes Topographiques (GTM) afin d'obtenir un algorithme permettant de l'apprentissage par transfert entre algorithmes travaillant sur des données similaires. Notre approche a été validée sur plusieurs jeux de données et a montré un fort potentiel.

1 Introduction

Data clustering algorithms learn to recognise the intrinsic structures of a dataset by regrouping similar data into different groups or clusters (Jain et al., 1999). Clustering is a difficult task, and the current exponential growth of the number and the size of datasets rises additional difficulties. In this context, individual clustering algorithms struggle to achieve good performances in a reasonable amount of time. In order to address this problem, some authors recently proposed the use of a Collaborative Clustering framework (Pedrycz et Hirota, 2008; Depaire et al., 2011; Ghassany et al., 2012; Zarinbal et al., 2015). This framework offers several solutions for these specific issues. The fundamental concept of collaboration is that the clustering algorithms operate locally (namely, on individual datasets) but collaborate by exchanging information (Pedrycz, 2002). In short, the goal of collaborative clustering is to have all algorithms improving their results based on the solution proposed by the collaborators.

Depending on the datasets on which the algorithms collaborate, there are three main types of collaboration : Horizontal, Vertical and Hybrid collaboration. The Hybrid Collaboration is a combination of both Horizontal and Vertical Collaboration. The definitions of Horizontal and Vertical Collaboration have been formalized in earlier works (Pedrycz, 2005; Grozavu et Bennani, 2010) and and can be seen as a constrained forms of transfer learning :

— **Horizontal Collaboration** : Several algorithms analyse different representations of the same observations. It can be applied to multi-view clustering, multi-expert clustering, clustering of high dimensional data, or multi-scale clustering.

— **Vertical Collaboration** : Several algorithms analyse different datasets sharing the same descriptors and having similar data distributions. The Collaborators are therefore looking for similar clusters. This is equivalent to knowledge transfer in identical feature spaces and can also be applied to process large datasets by splitting them and processing each subset with different algorithms exchanging information.

In this article we propose to adapt an horizontal collaboration framework (Sublime et al., 2015) for vertical collaboration purposes. The new method is based on the neural network structure of the Generative Topographic Maps (GTM : Bishop et al., 1998). In Section 2 we present the new method, Section 3 presents the experimental results and a conclusion is given is Section 4.

2 Collaborative clustering using the GTM structure

In an earlier work, we proposed a collaborative framework that allows different algorithms working on the same data elements to collaborate and mutually improve their results (Sublime et al., 2015). Since this algorithm showed good performances for horizontal collaboration applications, our goal was to modify it for transfer learning purposes. Doing so would require to get rid of the constraint that with this Framework all algorithms must work on the same data, even if they have access to different feature spaces. Instead, we wanted to have several algorithms working on different datasets in the same feature spaces and looking for similar clusters. Unfortunately, modifying the original Framework and its mathematical model to adapt them to this new context proved to be too difficult. Instead of working on a new Framework for vertical collaboration from scratch, we modified the original framework by using the properties of unsupervised neural networks based on vector quantization, such as the Self-Organizing Maps (SOM : Kohonen, 2001) or the GTM (Bishop et al., 1998).

The principle of these algorithms is that when initialized properly, and when used on datasets that have similar data distributions and are in the same feature spaces, they produce very similar topographic maps where the prototypes are roughly identical from one dataset to another. The maps and their equivalent prototypes can then be seen as a split dataset to which it is possible to apply our previous collaborative Framework without any modification. Therefore, using the structure of these unsupervised neural networks, it is possible to solve a vertical collaboration problem using an horizontal collaboration framework.

Our idea here is to apply the previously proposed collaborative framework to the second step of the GTM algorithm, i.e. the clustering of the prototypes using the EM algorithm. To do so, we use the map prototypes vectors as input datasets for our collaborative model. Under the hypothesis that all topographic maps have the same number of prototypes and underwent the same initialization, if we suppose that the different datasets have similar distributions, and knowing that we use the batch version of the GTM algorithm, the prototypes computed by different GTM algorithms can be seen as a dataset the attributes of which have been split between the different GTM algorithm instances. Therefore, since each prototype has a unique equivalent in each other topographic map, we can apply the collaborative framework for Heterogeneous algorithms.

Let us consider a group of GTM algorithms $C = \{c_1, ..., c_J\}$, which we independently apply to our dataset (observations) $X = \{x_1, ..., x_N\}, x_i \in \mathbb{R}^d$ resulting in the solution vectors $S = \{S^1, S^2, ...S^J\}$, where S^i is the solution vector provided by a given clustering algorithm

c_i searching for K_i clusters. A solution vector contains for each data element the label of the cluster it belongs to. $s_n^i \in [1..K_i]$ is the id of the cluster that algorithm c_i associates to the n^{th} element of X (i.e. x_n). We also note $\boldsymbol{\theta} = \{\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, ..., \boldsymbol{\theta}^J\}$ the parameters of the different algorithms (for example the mean-values and co-variances of the clusters).

Let $\Psi^{i\to j}$ be the *Probabilistic Correspondence Matrix* (PCM) mapping the clusters from a map c_i to the clusters of a map c_j. Likewise $\Psi_{a,b}^{i\to j}$ is the probability of having a data element being put in the cluster b of clustering algorithm c_j if it is in the cluster a of algorithm c_i.

Algorithm 1: Vertical Collaborative Clustering using GTM : V2C-GTM

```
Data transformation
forall the datasets $X^i$ do
    Apply the regular GTM algorithm on the data $X^i$.
    Run a first instance of the EM algorithm on the prototypes $\mathbf{W^i}$
end
Retrieve the prototypes $\mathbf{W^i}$ and their clustering labels $S^i$
Local step :
forall the clustering algorithms do
    Apply the regular EM algorithm on the prototypes matrix $\mathbf{W}$.
    $\to$ Learn the local parameters $\Theta$
end
Compute all $\Psi^{i\to j}$ matrices
Collaborative step :
while the system global entropy is not stable do
    forall the clustering algorithms $c_i$ do
        forall the $w_q \in \mathbf{W^i}$ do
            Find $s_q^i$ that maximize $P(w_q|s_q^i, \theta_{s_q}^i) \times \prod_{j\neq i} \Psi_{s_q}^{j\to i}$.
        end
    end
    Update the solution vectors $S$
    Update the local parameters $\Theta$
    Update all $\Psi^{i\to j}$ matrices
end
```

Let's now suppose that we are running these several GTM algorithms on different datasets that have the same features and for which we can assume the same cluster distributions can be found. If we use the same initialization for the prototypes of the topographic maps as described before, then we will have the prototype equivalents on the different maps. In this context, using the map prototypes $\mathbf{W}$ and their temporary cluster labels S from the local EM algorithm, we can apply a collaborative step to the EM algorithm. By doing so, the whole framework would be equivalent to a transfer learning process between the different datasets using vertical collaboration. Based on the collaborative version of the EM algorithm, the transfer learning algorithm with Generative Topographic Maps using Collaborative Clustering is described in Algorithm 1. Figure 1 is an illustration of the kind of result we can expect from this Framework used with topographic maps.

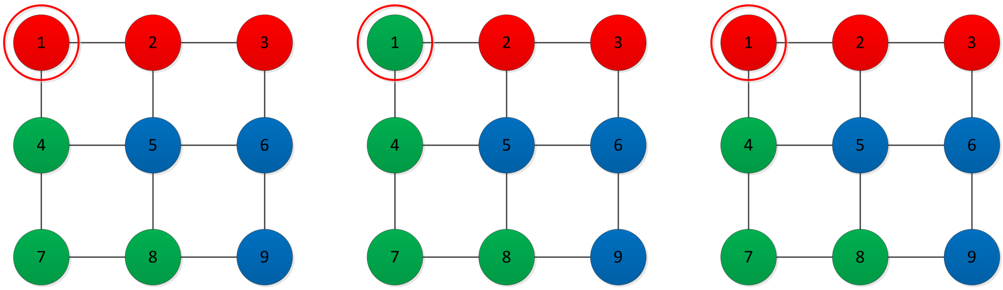

FIG. 1 – *Example of 3 collaborating topographic maps. Since they had the same initialization and are used on data that are assumed to have similar distributions, the neurons are equivalent from one map to another. This simple example shows a conflict on the cluster associated to the first neuron. Using our collaborative method, the first neuron will most likely be switched to the red cluster in the second map. With bigger maps, more algorithms and more clusters, conflicts will be more difficult to resolve than in this simple example.*

3 Experiments

To evaluate the proposed Collaborative Clustering approach, we applied our algorithm on several datasets of different sizes and complexity. We chose the following : Waveform, Wisconsin Diagnostic Breast Cancer (wdbc), Madelon and Spambase. The experimental protocol was the following : All datasets were randomly shuffled and split into 5 subsets with roughly equivalent data distributions in order to have the topographic maps collaborating between the different subsets. As criteria to validate our approach we consider the external purity index of the map which is equal to the average purity of all the cells of the map. If we know a class label for each element, the purity of a cell is the maximum percentage of elements represented by this cell and sharing the same label. A good GTM map should have a high purity index. First, we ran the local step, to obtain a GTM map for every subset. The size of all the used maps were fixed to 12×12 for the SpamBase and Waveform datasets and 4×4 for the wdbc and Madelon datasets. Then we started the collaborative step using our proposed collaborative framework with the goal of improving each local GTM by exchanging based on the maps found for the other subsets. We evaluated the maps purity of the final cluster, before and after collaboration. We compared our algorithm to the vertical version of the collaborative clustering using prototype-based techniques (GTM_{Col}) introduced in Ghassany et al. (2012).

In Table 1, we show the comparative results of the average gain of purity measured before and after collaboration. As one can see, while both methods give mild performances at improving the purity of a GTM map for our algorithm and a SOM map for the GTM_{Col} method, our algorithm is always positive on average for all datasets and our global results are also slightly better. It is easy to see that the proposed V2C-GTM method outperforms other methods by increasing every time the accuracy index after the collaboration step. We note here that our proposed V2C-GTM approach can use several distant information from several collaborators without fixing any collaboration parameters and usually the accuracy gain is positive. The two other methods can't.

These results are quite interesting because unlike the GTM_{Col} method that was specifi-

TAB. 1 – *Comparison of the average gain of purity before and after collaboration*

Dataset	Purity		
	V2C-GTM	GTM_{Col}	SOM_{Col}
SpamBase	+1.43%	-2.31%	-2.4%
WDBC	+0.416%	-2.45%	±0.32%
Madelon	+1.15%	+2.85%	+2.1%
Waveform	+0.11%	+0.07%	±2.6%

cally thought and developed with the idea of using it with semi-organized maps or generative topographic maps, the collaborative framework that we use was thought to be as generic as possible and not particularly adapted to the GTM algorithm. The conclusion we can draw from these results is that the probabilistic approach used by our framework is usually more effective than the derivative approach used in the other method.

4 Conclusion

In this article, we have proposed an original collaborative learning method based on collaborative clustering principles and applied to the Generative Topographic Mapping (GTM) algorithm. Our framework consists in applying the GTM algorithm on different datasets where similar clusters can be found (same feature spaces and similar data distributions). Our proposed method makes it possible to exchange information between different instances of the GTM algorithm with the goal of a faster convergence and better tuning of the topographic maps parameters. Our experimental results have shown our framework to be very effective at improving the final clustering of the maps involved in the collaborative process at least based on external indexes such as the maps purity, thus fulfilling its intended purpose.

One attractive perspective for our work would be to find a way to remove both constraints that either the observed data or the feature spaces have to be identical in order to use either horizontal or vertical collaboration. Getting rid of both constraints would enable transfer learning between datasets that are very different but have similar clusters structures.

Références

Bishop, C. M., M. Svensen, et C. K. I. Williams (1998). Gtm : The generative topographic mapping. *Neural Computation 10*, 215–234.

Depaire, B., R. Falcon, K. Vanhoof, et G. Wets (2011). PSO Driven Collaborative Clustering : a Clustering Algorithm for Ubiquitous Environments. *Intelligent Data Analysis 15*, 49–68.

Ghassany, M., N. Grozavu, et Y. Bennani (2012). Collaborative clustering using prototype-based techniques. *International Journal of Computational Intelligence and Applications 11*(3).

Grozavu, N. et Y. Bennani (2010). Topological collaborative clustering. *Australian Journal of Intelligent Information Processing Systems 12*(3).

Jain, A. K., M. N. Murty, et P. J. Flynn (1999). Data clustering : a review. *ACM Computing Surveys 31*(3), 264–323.

Kohonen, T. (2001). *Self-organizing Maps*. Springer Berlin.

Pedrycz, W. (2002). Collaborative fuzzy clustering. *Pattern Recognition Letters 23*(14), 1675–1686.

Pedrycz, W. (2005). *Knowledge-Based Clustering*. John Wiley & Sons, Inc.

Pedrycz, W. et K. Hirota (2008). A consensus-driven fuzzy clustering. *Pattern Recognition Letters 29*(9), 1333–1343.

Sublime, J., N. Grozavu, Y. Bennani, et A. Cornuéjols (2015). Collaborative clustering with heterogeneous algorithms. In *2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-18, 2015*.

Zarinbal, M., M. F. Zarandi, et I. Turksen (2015). Relative entropy collaborative fuzzy clustering method. *Pattern Recognition 48*(3), 933 – 940.

Summary

Collaborative clustering is a recent field of Machine Learning that shows similarities with both ensemble learning and transfer learning. In this article, we propose a method where we combine a collaborative framework with the topological structure of the Generative Topographic Mapping (GTM) algorithm and take advantage of it to transfer information between collaborating algorithms working on different datasets featuring similar distributions. The proposed approach has been validated on several datasets, and the experimental results have shown very promising performances.