Incremental Self-Organizing Maps for Collaborative Clustering

Denis Maurel[1,2], Jérémie Sublime[1], and Sylvain Lefebvre[1]

[1] LISITE, ISEP
28 rue Notre Dame des Champs 75006 Paris FRANCE
firstname.lastname@isep.fr
[2] CEDRIC, CNAM
292 rue Saint-Martin 75003 Paris FRANCE

Abstract. Collaborative clustering aims at revealing the common structures of data distributed on different sites using local clustering methods such as Self-Organizing Maps (SOM). To face the ever growing quantity of data available, incremental clustering methods are needed. This paper presents an algorithm to perform incremental SOM-based collaborative clustering without topological modifications of the map. The experiments conducted on several datasets demonstrate the validity of the method and present the influence of the batch size on the learning.

Keywords: collaborative clustering, incremental clustering, self organizing maps

1 Introduction

In this article, we study the clustering of several distributed datasets (called views), also known as collaborative clustering (CC). An incremental method for this problem, in which data are continuously added to each dataset through time, would help to solve some real life difficulties : it would make it possible to perform incremental data analysis with strong constraints on data exchange because of data confidentiality. Also, the distribution of data would allow to share the computational costs entailed by clustering problems. Unlike online methods, incremental algorithms are allowed to store data when they arrive, making it possible to work with batches instead of just singletons.

The elaboration of such a method presents several challenges. CC relies on prototype based clustering, thus reducing the number of available clustering algorithms. The second one lies in the adaptation of the collaborative update rules depending on the chosen clustering method. These adaptations have been done for algorithms such as Self-Organizing Maps (SOM) [5] and Fuzzy C-Means (FCM) [10, 8]. In this article we try to address theses challenges by focusing on the case of SOM.

This paper presents an incremental SOM-based CC method without topological modification of the SOM and which is robust to a possible data distribution evolution. The key component of this approach lies in the modification of the temperature function of the SOM which does not depend on time anymore. To the best of our knowledge, this kind of method has not been presented yet.

The rest of this paper is organized as follows: a brief overview of the collaborative and incremental clustering fields is presented in Section 2, then Section 3 presents a brief overview of the classical SOM method. Our approach on incremental SOM and its application to CC is presented in Section 4, followed by the experimental results presented in Section 5. Conclusion and future works are offered in Section 6.

2 Related Works

The aim of horizontal CC is to find a method to cluster the same samples described by different features in each view by making the different sites communicate with each other, without any sample being exchanged. This is achieved by the use of prototypes, which are vectors representing the information contained in the dataset. Collaborative clustering can be separated in two phases: the local phase, during which each clustering algorithm will work locally to find the best descriptors (prototypes) to describe their data, and the collaboration phase, during which each site will exchange the prototypes obtained in the first phase to share what has locally been learned.

An overview of CC can be found in [2]. It presents the main specificities of the field along with its main challenges. SOM-based CC has previously been studied in [7, 4, 11]. However, these proposed methods do not work with the incremental constraint that is studied here. Another clustering algorithm similar to SOM and called Generative Topographic Mapping (GTM) [1], has also been used in order to improve the results already available in SOM-based CC [12, 6]. However here again, the constraint of data streaming is not taken into account. This constraint may be found in research works solely dedicated to incremental clustering algorithms like in [14] for Fuzzy C-Means or in [3, 9] for SOM. In this latter case, the method provided by the authors is based on a topological modification of the already existing SOM. These references bring to light that even if methods exist for each part of the problem taken separately, there is currently no method for CC in an incremental context.

3 Classical SOM

SOM have originally been designed to perform unsupervised learning using a static database. We consider the following model composed by a map W of neurons $\omega_{j\in\{1..|W|\}}$which have mutual influences on each other defined by a neighboring function K. Given two neurons i and j, their mutual influence K_{ij} can be defined by:

$$K_{ij} = \exp\left(-\frac{d_1^2(i,j)}{\lambda(t)}\right) \tag{1}$$

where $\lambda(t)$ is defined as the temperature function modeling the range of the neighborhood affected by a neuron and d_1 being the Manhattan distance between two nodes, which corresponds to the number of edges separating the two nodes in the map. The temperature function is typically defined as:

$$\lambda(t) = \lambda_{max} \left(\frac{\lambda_{min}}{\lambda_{max}} \right)^{min(\frac{t}{t_{max}},1)} \tag{2}$$

where λ_{max} and λ_{min} are two fixed parameters which control the way the map evolves all along the learning process, and with t being the number of iterations already performed. The learning is considered finished when the sum of the distances between all the samples and all the neurons weighted by the kernel function is minimized:

$$R(\chi, W) = \sum_{i=1}^{N} \sum_{j=1}^{|W|} K_{j,\chi(x_i)} \|x_i - \omega_j\|^2 \tag{3}$$

with χ being the function returning the element of W with the smallest distance to x_i, also known as the Best Matching Unit (BMU) and with $X = \{x_i | i \in 1..N\}$ being the dataset used in the experiments, and N the total number of samples. The differentiation of this criterion implies the following update rule:

$$\omega_j^{(t+1)} = \omega_j^{(t)} + \epsilon(t) \sum_{i=1}^{N} K_{j,\chi(x_i)} (x_i - \omega_j^{(t)}) \tag{4}$$

with $\epsilon(t)$ is the time dependent learning step of the model.

4 Incremental SOM-based Collaborative Clustering

4.1 Incremental and Collaborative Clustering ?

A problem encountered with the incremental version of SOM-based CC is that existing incremental SOM are all based on topological modifications of the map [9, 3], and this kind of modifications are not permitted by the CC update rules. Indeed, the CC paradigm supposes that each dataset describes its data by the same number of prototypes to allow comparisons between several views and to keep topological mapping between each pair of views. In our case, the prototypes correspond to the neurons of the SOM, and as such without modification of the algorithm, the topology of each SOM has to be fixed during the initialization of the algorithm. Another problem encountered in general with incremental clustering is the possibility for the algorithms to answer changes in the data distribution through time. If the data distribution evolves, one has to be sure that the prototypes will follow the distribution of the most recent batches.

4.2 Incremental SOM

In our incremental version of SOM, we consider that the data are arriving all along the experiment. Therefore we assume that at each moment the model only knows the batch B of the N_{batch} last samples that have appeared during the learning. Our method here presents a variation of the original temperature function which aims at avoiding the dependence between the temperature and the time. This is motivated by the incremental aspect of the subject, for which it is not possible to define a time limit t_{max} at which the algorithm will end. In order to make the SOM responsive to the arrival of new data, a new function $\widetilde{\lambda}$ is defined by:

$$\widetilde{\lambda}(B, W) = \frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} \|x_i - \chi(x_i)\|_2 \tag{5}$$

With $B \subset X$ the batch currently used and with $|B| = N_{batch}$. This $\widetilde{\lambda}$ function is then capped between λ_{min} and λ_{max} in order to avoid extreme modifications of the SOM. This definition of the temperature function allows the SOM to be more responsive to novelty encountered in the batch. If the elements of a batch are far from the current neurons, the whole map would need to be adjusted to the new distribution of the sample, and this case is empirically achieved for high values of $\widetilde{\lambda}$. On the opposite, if the samples are near the current centroids, the map only needs some adjustments in order to better match the sample distribution. This case is achieved for low values of $\widetilde{\lambda}$. To clarify notations, the neighboring function which is defined by $\widetilde{\lambda}$ will be named $\widetilde{K}$.

4.3 Adaptation to Collaborative Clustering

In this paper, we only consider the case of horizontal CC as defined in [5]. For the rest of this paper, we consider datasets $\{X[i] | i \in 1..P\}$ containing the same set of objects described in different spaces, with P models (in our case SOM) being trained to represent each view separately. To clarify notation, $W^{m \in \{1..P\}}$ names the m-th model created using the m-th dataset. The point of CC is to make those models collaborate in order to reveal common structures among them. The main hypothesis that is made here is if an observation from the i-th dataset belongs to the j-th neuron of the i-th model, then the same observation in the i'-th model will also belong to its j-th neuron or to its neighborhood. In other words, *equivalent neurons from different maps should capture the same observations* [5].

In order to adapt the original criterion to the incremental version of CC, we use an approximation of this criterion using the kernel function $\widetilde{K}$ defined in Section 4.2 and where the distance are summed over the current batch instead of over the whole dataset:

$$\widetilde{R}^m(\chi, \omega) = \widetilde{R}_{Local}(W) + \widetilde{R}_{Collab}(W) \tag{6}$$

$$\widetilde{R}_{Local}(W) = \alpha_m \sum_{i=1}^{N_{batch}} \sum_{j=1}^{|W|} \widetilde{K}^m_{j,\chi(x_i)} \|x_i^k - \omega_j^k\|^2 \tag{7}$$

$$\widetilde{R}_{Collab} = \sum_{m'=1, m' \neq m}^{P} \beta_m^{m'} \sum_{i=1}^{N_{batch}} \sum_{j=1}^{|W|} (\widetilde{K}^m_{j,\chi(x_i)} - \widetilde{K}^{m'}_{j,\chi(x_i)})^2 \|x_i^m - \omega_j^m\|^2 \tag{8}$$

Algorithm 1 Incremental horizontal CC

```
Set the collaboration matrix $\{\alpha_{i,j}\}$
loop
  if new sample appears then
    Update batch as a FIFO stack of samples
    1. Local Step
    $\forall m \in 1..P$, Update prototypes of $W^m$ with one pass of incremental SOM
    2. Collaboration Step
    for $m \in 1..P$ do
      for $\omega \in W^m$ do
        $\omega = \omega + \frac{\sum_{i=1}^{N_{batch}} \widetilde{K}^m_{j,\chi(x_i^m)}(x_i^m - \omega) + \sum_{m'=1,m'\neq m}^{P} \sum_{i=1}^{N} \alpha_m L_{ij}(x_i^m - \omega)}{\sum_{i=1}^{N_{batch}} \widetilde{K}^m_{j,\chi(x_i^m)} + \sum_{m'=1,m'\neq m}^{P} \sum_{i=1}^{N} \alpha_m L_{ij}}$
        with $L_{ij} = (\widetilde{K}^m_{j,\chi x_i} - \widetilde{K}^{m'}_{j,\chi x_i})^2$
      end for
    end for
  end if
end loop
```

With α and β being defined as the collaboration coefficients, which in our cases are fixed by the user at the beginning of the experiments. Each α and β defines the comparative weights of the local and collaboratives terms in relation to each other.

A summary of the incremental horizontal CC can be found in Alg. 1.

It is interesting to note that a lot of computation time may be avoided by performing the local step only during the first few steps of the algorithm. The first local steps help to improve the final quality of the clustering in terms of mean neurons to samples distance, whereas additional steps do not change much the final results.

5 Experimental Results

5.1 Datasets and quality measures

To evaluate the method presented in this paper, we have tested them on four different datasets found on the UCI website [13]: Spam Base, Waveform, Wisconsin Diagnostic Breast Cancer (WDBC) and Isolet.

During our experiments, each of those datasets has been normalized and divided in 3 views each containing a third of the original variables. We suppose here that one has enough information on each variable to allow its normalization at the time it appears, for example by knowing its bounds. Each view will stand for an individual "site" which will collaborate with its peers.

The quality measures used during those experiments are the quantization error and the purity index commonly used to analyze SOM.

The quantization error can be defined in our case by the following expression:

$$qe = \frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} \|x_i - \omega_{\chi(x_i)}\|^2 \tag{9}$$

The purity of a neuron is defined by the proportion of the most represented class, and the purity of a map is equal to the average purity of its neurons.

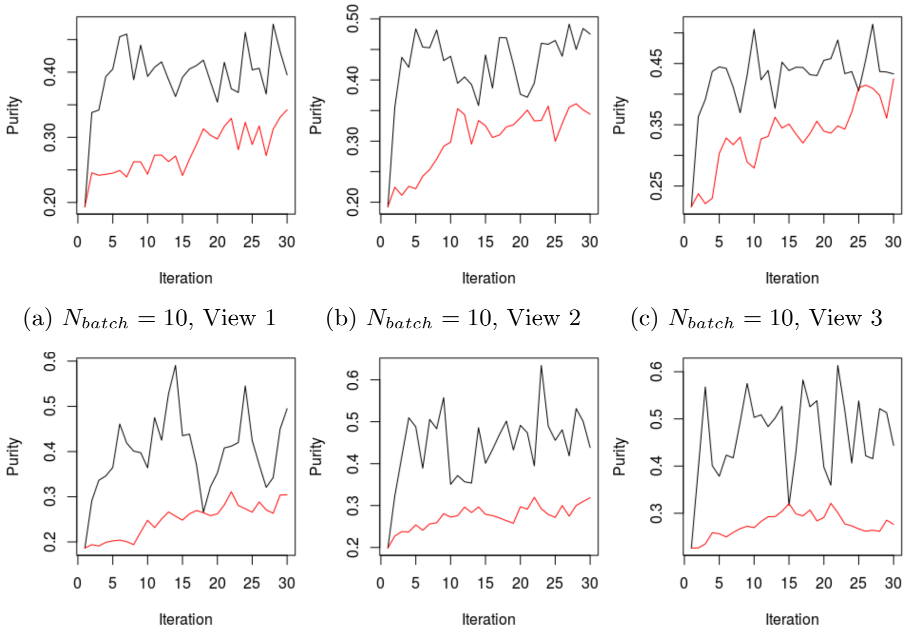
5.2 Experiments

In this section we present the results obtained on the four datasets. For these experiments, a 10x10 SOM has been used, with $\lambda_{min} = 0.3$, $\lambda_{max} = 3$, $\epsilon = 0.5$ (while it is usually time-dependent for classical SOM), $N_{batch} = 10$ and the local step of CC has been performed for the first 10 batches. The models have been trained using only the 30 first batches in order to test the early convergence of the model and because it appears that the results do not change a lot on the long term. The methods have been coded in R v3.2.3.

Table 1: Mean quantization errors on each database. ISOM stands for Incremental SOM and ICC stands for Incremental Collaborative Clustering. Bold numbers are the lowest ones for each column.

	Spam Base			Waveform			WDBC			Isolet		
	1	2	3	1	2	3	1	2	3	1	2	3
ISOM	0.31	**0.18**	0.18	**0.18**	**0.17**	**0.24**	**0.19**	**0.16**	0.20	2.15	2.84	2.85
ICC	**0.26**	0.19	**0.16**	0.23	0.19	0.30	0.19	0.19	**0.16**	**1.27**	**1.38**	**1.37**

The purities of the maps can be seen on Fig. 1a, Fig. 1b and Fig. 1c. For the sake of clarity, only the results for the Isolet dataset are presented here. It appears that CC improves the purity of the maps even if it makes it less stable than the incremental SOM. The terms stable and unstable refers here to the standard deviation of the purities through time, which is higher in the case of CC. This instability could be caused by the batch learning. The results of the learning phase depends on the incoming data: if one specific class is more represented than the others in a batch (which is more prone to happen if the batch is small), the neurons updates for this step will focus on this class specifically, and in the end it might hurt the next phases because of the bias acquired by the model for this specific class. An increase of the instability of the purities proportionally to the decrease of the batch size can be seen by comparing Fig. 1a, 1b and 1c to Fig. 1d, 1e and 1f, where we have set $N_{batch} = 3$. It is possible that the collaborative part of Eq. 6 makes the centroids move from the local solution minimizing Eq. 7: the collaborative SOM makes the centroid move toward a global solution rather than toward a local one. Concerning the quantization errors presented in Table 1, they all stay in an acceptable range considering that the data are scaled before the training. The case of the Isolet dataset is special because there are many more features than in the other datasets, and the database is sparser, point which may lower the performances of each local model depending on the distribution of features. Otherwise, it appears that the errors are always close to each other with a small advantage for the incremental SOM.



(a) $N_{batch} = 10$, View 1
(b) $N_{batch} = 10$, View 2
(c) $N_{batch} = 10$, View 3
(d) $N_{batch} = 3$, View 1
(e) $N_{batch} = 3$, View 2
(f) $N_{batch} = 3$, View 3

Fig. 1: Evolution of the purities for the Isolet database with 2 different N_{batch}. The red lines represent the incremental SOM whereas the black lines represent the collaborative SOM. Each iteration corresponds to the arrival of a new sample

6 Conclusion and Future Works

In this study, we proposed a methodology to perform incremental SOM without topological modifications of the map as well as the application of this method to adapt horizontal CC according to the incremental constraint. Its application to vertical CC is possible but has not been described in this paper. With these methods, the temperature function $\widetilde{\lambda}$, and so the neighboring function $\widetilde{K}$ of the generated maps are no longer time-dependent, and now only depend on incoming data. Knowing that, the map can be adapted to continuously incoming data. The presented methods have been tested on 4 different datasets, and the results show that the version of incremental SOM presented in this paper can be adapted to perform incremental CC. The influence of the parameter N_{batch}, namely its impact on the stability of the learning, has also been investigated.

To pursue this work, we plan to investigate the methods which would allow to adapt topological modifications on SOM in the context of CC. Furthermore, we plan to adapt what has been presented in this research work to the GTM, which are by nature similar to SOM.

References

1. Bishop, C.M., Svensén, M., Williams, C.K.: Gtm: The generative topographic mapping. Neural computation 10(1), 215–234 (1998)
2. Cornuéjols, A., Wemmert, C., Gançarski, P., Bennani, Y.: Collaborative clustering: Why, when, what and how. Information Fusion 39, 81–95 (2018)
3. Deng, D., Kasabov, N.: Esom: An algorithm to evolve self-organizing maps from online data streams. In: Neural Networks, 2000. vol. 6, pp. 3–8. IEEE
4. Filali, A., Jlassi, C., Arous, N.: Som variants for topological horizontal collaboration. In: Advanced Technologies for Signal and Image Processing (ATSIP), 2016 2nd International Conference on. pp. 459–464. IEEE (2016)
5. Ghassany, M., Grozavu, N., Bennani, Y.: Collaborative clustering using prototype-based techniques. International Journal of Computational Intelligence and Applications 11(03), 1250017 (2012)
6. Ghassany, M., Grozavu, N., Bennani, Y.: Collaborative multi-view clustering. In: The 2013 International Joint Conference on. pp. 1–8. IEEE (2013)
7. Grozavu, N., Cabanes, G., Bennani, Y.: Diversity analysis in collaborative clustering. In: 2014 International Joint Conference on. pp. 1754–1761. IEEE (2014)
8. Mitra, S., Banka, H., Pedrycz, W.: Rough–fuzzy collaborative clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 36(4), 795–805 (2006)
9. Papliński, A.P.: Incremental self-organizing map (isom) in categorization of visual objects. In: ICONIP. pp. 125–132. Springer (2012)
10. Pedrycz, W., Rai, P.: Collaborative clustering with the use of fuzzy c-means and its quantification. Fuzzy Sets and Systems 159(18), 2399–2427 (2008)
11. Rastin, P., Cabanes, G., Grozavu, N., Bennani, Y.: Collaborative clustering: How to select the optimal collaborators? In: Computational Intelligence, 2015 IEEE Symposium Series on. pp. 787–794. IEEE (2015)
12. Sublime, J., Grozavu, N., Cabanes, G., Bennani, Y., Cornuéjols, A.: From horizontal to vertical collaborative clustering using generative topographic maps. International Journal of Hybrid Intelligent Systems 12(4), 245–256 (2015)
13. UCI: Machine Learning Repository. https://archive.ics.uci.edu/ml/index.php
14. Wang, Y., Chen, L., Mei, J.P.: Incremental fuzzy clustering with multiple medoids for large data. IEEE Transactions on Fuzzy Systems 22(6), 1557–1568 (2014)