Analysis of the influence of diversity in collaborative and multi-view clustering

Jérémie Sublime*†, Basarab Matei† and Pierre-Alexandre Murena ‡
*LISITE Laboratory, RDI Team - ISEP Paris, France
Email: jeremie.sublime@isep.fr
†Université Paris 13, Sorbonne Paris Cité, LIPN UMR CNRS 7030
99 av. J-B Clément, 93430 Villetaneuse, France
Email: basarab.matei@lipn.univ-paris13.fr
‡ LTCI CNRS, Télécom ParisTech, Université Paris-Saclay
46 rue Barrault, 75013 Paris, France
Email: murena@telecom-paristech.fr

***Abstract*—Multi-source clustering is common data mining task the aim of which is to use several clustering algorithms to analyze different aspects of the same data. Well known applications of multi-source clustering include horizontal collaborative clustering and multi-view clustering, where several algorithms combine their strengths by exchanging information about their finding on local structures with a goal of mutual improvement. However, many of these proposed algorithms and statistical models lack the capability to detect weak collaborations that may prove detrimental to the global clustering process.**

In this article, we propose a weighing optimization method that will help detecting which algorithms should exchange their information based on the diversity between the different algorithms' solutions.

I. Introduction

Data Clustering is a fundamental task in the process of knowledge extraction from databases that aims to discover intrinsic structures in a set of objects by forming clusters that share similar features. This task is more difficult than supervised classification in the sense that in clustering the number of clusters to be found is generally unknown and that it is difficult to rate the quality of a clustering result. Recent data sets are even more challenging for traditional clustering algorithms: They present different types of features for the same data elements, very large data sets, distributed data sets, etc. This increased complexity in an already hard problem makes it difficult for a single clustering algorithm to give competitive results with a high degree of confidence. However, very much like in the real world, such problems can be tackled more easily by splitting the tasks between several algorithms that would work together with the goal of achieving more reliable results. One of the idea behind multi-source clustering is that unsupervised learning being a difficult task, we can't say much about the quality of a solution found by a single algorithm. However, if several algorithms find similar solutions, then there is a good chance that they found a structure that makes sense.

Collaborative clustering methods [1], [2], [3], [4] and multi-view clustering methods [5], [6], [7] are two close families of frameworks that effectively enable several clustering algorithms to work on different views of the same data sets, possibly with redundant features. While they were originally thought for different purposes, horizontal collaborative clustering and multi-view clustering are in fact equivalent in most cases. The main differences between the two are the following:

- Multi-view clustering usually applies to several instances of the same algorithm working on different views of the same data, while horizontal collaborative clustering can allow different types of algorithms to work together [8], [4].
- Collaborative clustering original application was distributed data sets in cloud environments [1], [9] and had originally the constraint that all collaborators should search for the same number of clusters. Its applications were later extended to multi-view clustering, muli-scale clustering and transfer learning [3], [10]. On the other hand, multi-view clustering comes from a statistical background of several algorithms analyzing different views of the same data with different attributes [11], [12].

In this article, we will assume that multi-view clustering can be seen as a sub-case of horizontal collaborative clustering and therefore that the optimal conditions to achieve good results are the same for both families of algorithms.

Collaborative clustering usually follows a 2-step process where first all algorithm work on their own to build initial solution, and then they exchange their information to improve each other results. These two steps are sometimes followed by an ensemble learning step which aims at reaching a consensus with the final results after collaboration, see Figure 1. We will not address the consensus step in this work.

Within this context, the aim of this article is to propose a weighting method that will determine how the different algorithms should influence each other in either framework with the goal of maximizing the quality of the results. To do so, we use the fact that most of these methods rely on optimizing a log-likelihood the form of which is given in Equation (1)

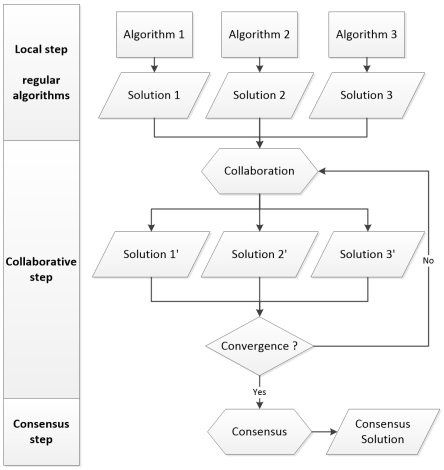

Fig. 1. Example of a collaboration or multi-view scheme

[5], [13], [2], where J is the number of algorithms in the system, $\mathcal{L}(X^i, S^i)$ is the local log-likelihood of the i^{th} algorithm based on the local information, $\Delta(S^i, S^j)$ a difference function between to solution, and $\tau_{j,i}$ a weight determining the influence of algorithm j over algorithm i. Depending on the collaborative algorithm, common $\Delta(S^i, S^j)$ functions include the Kullback-Leibler divergence, various *a posteriori* entropy measures between the solutions and diversity indexes.

$$L(X, \boldsymbol{S}) = \sum_{i=1}^{J} \left(\mathcal{L}(X^i, S^i) - \sum_{j \neq i} \tau_{j,i} \cdot \Delta(S^i, S^j) \right) \quad (1)$$

The model that we introduce in this article relies on weighting the influence of each algorithm (the $\tau_{j,i}$) using a stability and diversity-based criterion obtained by optimizing the global likelihood function of the collaborative system under the Karush-Kuhn-Tucker conditions. We then interpret the result of this optimization to draw a broader conclusion on the role of cluster diversity in the context of collaborative clustering and the optimal conditions for good collaborations.

The remainder of this article is organized as follows: In section 2 we introduce the formalism, model and algorithm that we use for our collaborative framework. In section 3, we present our optimization method for the collaborative term, the resulting optimal weights, and the interpretation that we can make from these results. Section 4 features some experimental results. Finally, this article ends with a conclusion and perspective on future works.

II. A Framework for horizontal collaboration

A. Formalism

In multi-source clustering we consider a group of algorithms that are working on the same data elements, albeit possibly with a different view point. To this end, let us consider $X = \{x_1, ..., x_N\}, x_n \in \mathbb{R}^d$ a data set containing N elements, each of them with d real number features. We have J clustering algorithms $\mathcal{A} = \{\mathcal{A}^1, ..., \mathcal{A}^J\}$ working on this data set. Each clustering algorithm will have its own parameters Θ^j to describe either the clusters or its model, and will produce its own solution S^i made of K_i clusters, based on the features of the data set it has access to $X^i \subseteq X$. The solutions S^i are computed for the data X using the parameters Θ^i. In the case of hard clustering S^i is a vector of size N, for fuzzy clustering it is a matrix of size $N \times K_i$.

The partitions produced by the algorithms can be either fuzzy partitions or hard ones. The solutions S^i output by the algorithms are therefore two-dimensional matrices of size $N \times K_i$ where each element $s_n^i(c)$ expresses the responsibility given by algorithm $\mathcal{A}^i$ to a cluster c for the data element x_n. In soft clustering we have $s_n^i(c) \in (0, 1)$, while in hard partitioning $s_n^i(c) \in \{0, 1\}$ and $s_n^i(c) = 1$ for only one cluster c for each data element x_n. In this last case, we will use $s_n^i = c$ to express the fact that data element x_n has been (entirely) allocated to cluster c by algorithm $\mathcal{A}^i$ (i.e. $s_n^i(c) = 1$). Thus we have : $\forall i$, $\mathcal{A}^i = \{X^i, \Theta^i, S^i, K_i\}$. In order to simplify the notation, we will most often use X and x_n to describe the data set and the data indifferently for all algorithms, but the differences in viewpoints should still be kept in mind.

One general setting in which to cast the participating clustering algorithms is the probabilistic model-based framework. Many clustering techniques can indeed be depicted in this model, e.g. fuzzy C-Means, Gaussian mixtures models (GMM), mixtures of Bernouilli distributions, etc.

For example, in the case where $\mathcal{A}^i$ would follow a GMM with K_i clusters, we would have $\Theta^i = \{\theta_1^i, ..., \theta_{K_i}^i\}, \theta_k^i = \{\pi_k^i, \mu_k^i, \Sigma_k^i\}$ where μ_k^i is the mean value of the k^{th} cluster, Σ_k^i its covariance matrix and π_k^i the mixing probability.

However, the Θ^i are not limited to parameters describing densities of probability and can also contain the configuration parameters of the different algorithms.

B. Algorithm

In this article, and within the context of horizontal collaboration, we are dealing with clustering algorithms that may be very different in nature and can possibly work over different feature spaces of the same data set. Given these condition, it is impossible for these algorithms to collaborate by exchanging their prototypes or parameters Θ^j. Earlier works have shown that the only solution for heterogeneous algorithms to collaborate must be based on their solution vectors [8], [14]. Conveniently, with the data having the same id for all the algorithms, it is possible to compare their partitions and to make such communication between them happen.

As described in the introduction, the main collaborative framework consists in a local step followed by a collaborative step. This second step consists in exchanging the solutions of the different algorithms to compare and improve their respective partitions of the data.

During the iterative collaborative step, for a given algorithm $\mathcal{A}^i$ and a data x_n, this can be translated into Equation (2), where $s_n^i(c)|_{coll}$ is the responsibility given locally to cluster c by algorithm $\mathcal{A}^i$ for the data element x_n after taking

into account the solutions of the other algorithms; $s_n^i(c)$ denotes the responsibility given locally to cluster $c \in [1..K_i]$ by algorithm $\mathcal{A}^i$ depending on its parameters Θ^i; λ_i is a parameter that controls the degree to which each algorithm accept information from the other collaborators; $\mathcal{Q}$ is the ensemble of all possibles combinations of clusters that can be entertained by the different algorithms for a given data x_n; $\boldsymbol{q} = \{q_1, ..., q_J\}, q_j \in [1..K_j]$ one of this possible combination of clusters; and $g^i(\boldsymbol{q}, c)$ is a combination function chosen to assess the likelihood of having $q_i = c$ given the other clusters in the combination $boldsymbolq$. In Equation (2), for any given data x_n, we consider all the possible combinations $\boldsymbol{q} \in \mathcal{Q}$ so that $q_i = c$.

$$s_n^i(c)|_{coll} = (1-\lambda_i) \cdot s_n^i(c) + \lambda_i \sum_{\boldsymbol{q} \in \mathcal{Q} | q_i = c} \left(g^i(\boldsymbol{q}, c) \cdot \prod_{\substack{j=1 \\ j \neq i}}^{J} s_n^j(q_j) \right) \quad (2)$$

Equation (2) can be interpreted as follows: The first term $(1-\lambda_i) \cdot s_n^i(c)$ is a *local term* weighted by the parameter λ_i. It gives the likelihood for the element x_n to be linked to a cluster c based only on the local parameters of the algorithm $\mathcal{A}^i$. For probabilistic algorithms, we have: $s_n^i(c) = p(s_n^i = c | x_n, \theta_c^i)$. Thus, we want to highlight that $s_n^i(\cdot)$ may have different values after each iteration of the algorithm as Θ^i is optimized. $s_n^i(\cdot)$ must consequently be recomputed during each iteration: During the first iteration of the collaborative step, this is the same $s_n^i(\cdot)$ than at the end of the local step, but after that it must be recomputed using the updated local parameters.

The second term of Equation (2) is called a *collaborative term* and evaluates the likelihood for the element x_n to be linked to a cluster c based on the other algorithms partitions and their choice of cluster for the same data x_n, weighted by λ_i. To do so, in the case of fuzzy clustering this likelihood must be evaluated for all possibles combinations of clusters where $q_i = c$ -with $g^i(\boldsymbol{q}, c)$- and is weighted based on the likelihood of this combination to occur for the data x_n assuming that all algorithms are independent from each other - hence the $\prod s_n^j(q_j)$-. In this context, $g^i(\boldsymbol{q}, c)$ is a combination function that rates the likelihood of a combination $\boldsymbol{q}$ to contain $q_i = c$ for algorithm $\mathcal{A}^i$. The combination function should be viewed as an approximation of the *a posteriori* distribution involved in the computation of the likelihood of the element x_n. Examples of possible combination functions $g^i(\cdot)$ will be given in the next section where we will see that the computation of g^i is done using the a posteriori partitions of the algorithms and that the idea behind optimizing collaborative terms containing these functions is to reduce the entropy between then different partitions.

This collaborative term can be interpreted as follows: for each cluster combination $\boldsymbol{q} \in \mathcal{Q}$, $g^i(\boldsymbol{q}, c)$ assesses the likelihood of having $q_i = c$ with this combination $\boldsymbol{q}$. Then $g(\cdot)$ is weighted by the likelihood of such combination with the other collaborators $\mathcal{A}^j \neq \mathcal{A}^i$ for the data x_n. The sum over all the $\boldsymbol{q} \in \mathcal{Q}$ makes it possible to take into account the fuzziness of the solutions by summing up the likelihood of having $q_i = c$ for all possible combinations.

For hard clustering or when $\mathcal{Q}$ contains too many combinations (K^J on average) for Equation (2) to be computed in a reasonable time for all the data, a semi-hard version is given in Equation (3) where $\boldsymbol{q}^{max_n}$ is the most likely combination of clusters for x_n so that $\boldsymbol{q}^{max_n} \in \mathcal{Q}$, $q_i = c$ and $\forall j \neq i, q_j = argmax_{q_i} s_n^j(q_i)$. Since Equation (3) considers only the most likely combination of cluster for each element x_n instead of all possible combinations, it can be seen as an approximation of Equation (2) that is much easier to compute. With this approximation the product $\prod_{j \neq i}^{J} s_n^j(q_j)$ from Equation (2) is different from 0 only when $\boldsymbol{q} = \boldsymbol{q}^{max_n}$, and is therefore the equation simplifies.

Note that when factorizing λ_i in Equation (2) or (3) we find the original form shown in Equation (1) and we can figure out the expression for the diversity function in this system.

$$s_n^i(c)|_{coll} = (1-\lambda_i) \cdot s_n^i(c) + \lambda_i \cdot g^i(\boldsymbol{q}^{max_n}, c) \quad (3)$$

The combination function $g(\cdot)$, as an approximation of the underlying *a posteriori* distribution, has specific properties:

- $g^i(\boldsymbol{q}, c)$ needs to increase strictly when the consensus between the different algorithms grows on the likelihood of having $q_i = c$ for a given combination q.
- $g^i(\boldsymbol{q}, c)$ needs to be normalized so that for any cluster combination $\boldsymbol{q}$ that occurs at least once, we have: $\sum_{c \in [1..K_i]} g^i(\boldsymbol{q}, c) = 1$.
- When the algorithms have the exact same partitions and $c = argmax_{q_i} s_n^i(q_i)$, then: $g^i(\boldsymbol{q}^{max_n}, c) = 1$.

These properties ensure that Equations (2) and (3) are well defined. Since the goal of collaborative clustering is to mutually improve the results for all algorithms, the best way to achieve this goal is to optimize Equation (2) over all the data for all the algorithms. This is equivalent to find $\mathbf{S}$ that maximizes the likelihood function shown in Equation (6), where $\mathbf{S} = \{S^1, ..., S^J\}$ contains all the clustering partitions and where $s_{ni}^{max} = argmax_c(s_n^i(c)|_{coll})$ is the cluster that has the maximum score based on Equation (2) or (3).

An elegant way to maximize Equation (6), and thus to increase the likelihood of the solutions, is to use a parallelized version of the EM algorithm [15].

The general Framework of our collaborative algorithm is shown in Algorithm 1. This algorithm includes the local step where each algorithm works on its own, and the collaborative step with the meta-EM algorithm. The collaborative part of the algorithm is iterative with the optimization process continuing as long as the system global entropy [16], [17], [8] from Equation (4) is not stable.

$$\mathcal{H} = \sum_{i,j} \frac{-1}{K_i \ln(K_j)} \sum_{l=1}^{K_i} \sum_{m=1}^{K_j} \frac{|c_l^i \cap c_m^j|}{|c_l^i|} \ln \left(\frac{|c_l^i \cap c_m^j|}{|c_l^i|} \right) \quad (4)$$

In Equation (4), the notation c_l^i refers to the l^{th} cluster of algorithm $\mathcal{A}^i$, $|c_l^i|$ is the number of data in this cluster, and $|c_l^i \cap c_m^j|$ is the number of data linked to the l^{th} cluster of $\mathcal{A}^i$ and the m^{th} cluster of $\mathcal{A}^j$ at the same time.

$$L(\mathbf{S}) = \sum_{i=1}^{J}\sum_{n=1}^{N}\left((1-\lambda_i)\cdot s_n^i(s_{ni}^{max}) + \lambda_i \sum_{\boldsymbol{q}\in\mathcal{Q}|q_i=s_{ni}^{max}} g^i(\boldsymbol{q}, s_{ni}^{max})\cdot\prod_{\substack{j=1\\j\neq i}}^{J} s_n^j(q_j)\right) \tag{6}$$

$$\mathcal{C}(\mathbf{S}) = \sum_{i=1}^{J}\sum_{n=1}^{N}\lambda_i \sum_{\boldsymbol{q}\in\mathcal{Q}|q_i=s_{ni}^{max}}\left(\frac{1}{Z(i,\boldsymbol{q})}\sum_{j\neq i}\tau_{j,i}\frac{|s_{ni}^{max}\cap q_j|}{|q_j|}\right)\cdot\prod_{\substack{j=1\\j\neq i}}^{J} s_n^j(q_j) \tag{7}$$

Algorithm 1: Collaborative Clustering Guided by Diversity: General Framework

Local step:
forall *clustering algorithms* $\mathcal{A}^i$ **do**
  Apply $\mathcal{A}^i$ on the data X^i.
  $\rightarrow$ Learn the local parameters Θ^i
  $\rightarrow$ Compute $s_n^i(c)$ for all data point $\mathbf{x}_n$ and all cluster c considered by $\mathcal{A}^i$
end
Collaborative step:
while *the system's global entropy* $\mathcal{H}$ *is not stable* **do**
  Meta E-Step:
  forall *clustering algorithms* $\mathcal{A}^i$ **do**
    forall $\mathbf{x}_n^i \in \mathbf{X}^i$ **do**
      Assess $s_n^i(c)|_{coll}$ using Equation (2) or (3)
    end
  end
  Meta M-Step:
  forall *clustering algorithms* $\mathcal{A}^i$ **do**
    Update the local parameters Θ^i -if any, and when it is relevant- by using a maximum likelihood estimation:

$$\Theta^i = \underset{\Theta}{\text{Argmax}}\{\log p(X^i|\Theta)\} \tag{5}$$

  end
end

III. Optimizing the collaboration

A. Introducing the problem

The algorithm introduced in the previous section has convergences properties identical to these of the original EM algorithm [15], [18], with $L(\mathbf{S})$ (Equation (6)) converging strictly toward a stationary point. Furthermore, using a linear combination function $g(\cdots)$ such as the one in Equations (8) or (15) for which all values can be pre-computed, the overall complexity of the algorithm is $O\left(J\times(argmax_i\left[Cpx(\mathcal{A}^i(N))\right]+K\times N)\right)$ where $Cpx(\mathcal{A}^i(N))$ is the algorithmic complexity of collaborator $\mathcal{A}^i$. The collaboration term is therefore linear and does not affect the overall algorithm complexity in most cases. However, these properties say nothing on about the following points:

- The global convergence of the likelihood function doesn't not warrant that all algorithms will converge locally. Oscillations and partitions swaps are possible locally with a global likelihood being stable.
- The fact that the algorithm converges quickly does not mean that the results will be good when using quality indexes. The risk of negative collaboration -where poorly performing algorithms drag down all the others- remains quite likely.

To solve this issue, in this section, we study how the choice of a good combination function $g(\cdot)$ and its further optimization can lead to a higher value of the likelihood function and reduce the risk of negative collaboration by further optimizing weight factors between the algorithms.

$$g_+^i(\boldsymbol{q}, c) = \frac{1}{Z}\sum_{j\neq i}\tau_{j,i}\frac{|c\cap q_j|}{|q_j|} \tag{8}$$

To do so, we propose to use the combination function $g(\cdot)$ described in Equation (8) that allows to weight the different algorithms. This function sums the pairwise likelihood of having $q_i = c$ based on the *a posteriori* intersections of the clusters. Each likelihood is then weighted by the term $\tau_{j,i}$ that describes the degree to which algorithm $\mathcal{A}^j$ can influence algorithm $\mathcal{A}^i$. This function is equivalent to a weighted vote where each algorithm gives a degree of agreement between its cluster q_j and the local algorithm cluster $q_i = c$. The term Z is a normalization constant so that $\sum_{c=1}^{K_i} g_+^i(\boldsymbol{q}, c) = 1$.

$$Z(i,\boldsymbol{q}) = \sum_{c=1}^{K_i}\sum_{j\neq i}\tau_{j,i}\frac{|c\cap q_j|}{|q_j|}$$

We also remind that $|q_j|$ is the number of elements that are in the cluster q^j of algorithm $\mathcal{A}^j$, and that $|c\cap q_j|$ is the number of elements that are in both the cluster c of algorithm $\mathcal{A}^i$ and the cluster q^j of algorithm $\mathcal{A}^j$.

Given this Equation, the weights $\tau_{j,i}$ should obviously be chosen to maximize the likelihood function L which in turn should ensure better results. Using Equations (6) and (8), this is equivalent to find the $\tau_{j,i}$ that maximize the collaborative term for all data and all algorithms and thus to maximizing Equation (7).

By changing the position of the sum $\sum_{j\neq i}\tau_{j,i}$, Equation (7) simplifies into Equation (9) where $\beta_{j,i}$ contains the most information from the collaborative term and can be interpreted as

a non-normalized criterion assessing the degree of agreement of algorithm $\mathcal{A}^j$ with algorithm $\mathcal{A}^i$. $\beta_{j,i}$ is an asymmetrical information based similarity measure between two different clustering solutions.

$$\mathcal{C}(\mathbf{S}) = \sum_{i=1}^{J} \sum_{j \neq i} \tau_{j,i} \cdot \beta_{j,i} \tag{9}$$

B. Optimization with the Karush-Kuhn-Tucker conditions

We now want to find the $\tau_{j,i}$ that maximizes Equation (9). To do so, we use the Karush-Kuhn-Tucker conditions (KKT) [19] assuming that the weights $\tau_{j,i}$ are positive and subject to the constraint of parameter p given in Equation (10).

$$\forall i \quad \sum_{j \neq i}^{J} (\tau_{j,i})^p = 1, \qquad p \in \mathbb{N}^* \tag{10}$$

The results of the optimization under the Karush-Kuhn-Tucker conditions are shown bellow for different values of p in Equations (11), (12) and (13). The detailed calculi are available in subsection IV.E thereafter.

- If $p = 1, \forall j \neq i$:

$$\tau_{j,i} = \begin{cases} \frac{1}{Card(\beta_{j,i} = max_j(\beta_{j,i}))} & \text{if } \beta_{j,i} = max_j(\beta_{j,i}) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

- For $p > 1$:

$$\forall j \neq i, \tau_{j,i} = \frac{|\beta_{j,i}|^{\frac{1}{p-1}}}{(\sum_{k \neq i}^{J} |\beta_{k,i}|^{\frac{p}{p-1}})^{\frac{1}{p}}} \tag{12}$$

- When $p \to \infty$:

$$\forall (i,j), \tau_{j,i} = Cte \tag{13}$$

The summary of these results is the following: In the context of collaborative clustering, the results should be better if each individual algorithm gives higher weights to algorithms that have solutions similar to the local one (high $\beta_{j,i}$ value for a given $\mathcal{A}^i$).

If we go into the details, we see that the degree to which one algorithm should collaborate with other collaborators that have dissimilar solutions depends on p in Equation (10). For $p = 1$ (Equation (11)), each algorithm would only collaborate with the algorithm that has the most similar solution. If several algorithms have the same most similar solution, they would be given the same weight. When using a higher degree of normalization (Equation (12)), the algorithms with the most similar solutions would still be favored to optimize the likelihood of the global collaborative framework, but algorithms the solutions of which have a lesser degree of similarity would still be taken into consideration locally. In fact as p gets higher, the solutions from dissimilar algorithms would have a heavier and heavier weight, and at some point they would matter just as much as any other solution. In this later case, when the value of p is high enough, this would be equivalent to give the same weight to all the algorithms (Equation (13)).

C. Extension to other likelihood functions

In this subsection, we argue that the results of the previous subsection are applicable to any function that uses weights in the context of horizontal collaboration. To prove our point, we first take the example of the simplified collaboration function with hard or semi-fuzzy clusters. Then we use a different weighted combination function $g(\cdot)$ from one of our earlier work. In both cases, we show that the best weights still rely on giving more importance on solution that are similar to the local partition.

In the case of the simplified collaboration with hard or semi-fuzzy clusters using Equation (3) and the same combination function $g_+(\cdot)$ from Equation (8). In this case, we can find the exact same results after the optimization but with a simplified $\beta_{j,i}$, see Equation (14).

$$\tilde{\beta_{j,i}} = \lambda_i \sum_{n=1}^{N} \frac{1}{Z(i,n)} \frac{|s_{ni}^{max} \cap s_{nj}^{max}|}{|s_{nj}^{max}|} \tag{14}$$

The same properties can also be found with other combination functions. For example, in one of our earlier work [8], we use a product for the combination function instead of a sum. The weighted version of this combination function as shown in Equation (8) would lead to the same optimal weights as in Equations (11), (12) and (13), with a slightly $\beta_{j,i}$ that clearly still is a similarity index. This $\beta_{j,i}$ is shown in Equation (16).

$$g_*^i(\boldsymbol{q}, c) = \frac{1}{Z} \prod_{j \neq i} \left(\frac{|c \cap q_j|}{|q_j|} \right)^{\tau_{j,i}} \tag{15}$$

$$\beta_{j,i}^* = \sum_{n=1}^{N} \ln \left(\frac{|s_{ni}^{max} \cap s_{nj}^{max}|}{|s_{nj}^{max}|} \right) - \ln Z(i,n) \tag{16}$$

D. Interpretation of the results

These results are interesting because they go against the common idea that collaboration works best between collaborators having an average diversity [20], [21]. Indeed, common sense would want us to think that a low diversity means not much room for improvement since everyone agrees, and a high diversity not enough common ground to reach an agreement, thus making average diversity the best case scenario.

However it is our opinion that this interpretation carries the bias of supervised learning. If we think about the goal of collaboration in the context of unsupervised learning, these mathematical results make sense: We are in a situation where each algorithm does an exploratory task and has no supervised index to rely on to guess quality of its solution. Therefore, when several algorithms find solutions that are similar, it is quite likely that they have actually found a structure in the data. As a consequence collaborating with algorithms that have solutions similar to the local partitioning is a convenient way to avoid the risk of negative collaboration. There are actually good reasons not to collaborate with an algorithm whose results are too different from the local partition: Such collaborator may be in a feature space where the clusters to be found are completely different even for the same objects. The

dissimilarity of a solution with all others may simply mean that this solution is a poor one.

These results can also be linked to recent works on clustering stability [22], [23]. A clustering is said to be stable if the partition remains similar when the data set or the clustering process are perturbed. In the context of collaborative clustering, the perturbations would be that (1) we observe the same data in different feature spaces, and (2) we use different algorithms. With our proposed weighting methods, the algorithms with the strongest influence will be these with solutions most often similar to the other algorithms' solutions. It matches with the definition of stability: Such solutions that highlight common structures and clusters through several feature spaces with different algorithms are the most stables.

As a conclusion to this theoretical section and based on the previous results, we make the following proposition:

Proposition 1: In the context of horizontal collaboration between several heterogeneous algorithms, the most efficient way for a given algorithm $\mathcal{A}^i$ to collaborate is to favor exchanges with other collaborators $\mathcal{A}^j$ that have similar and stable solutions.

If this proposition may seem somewhat counter intuitive, we are not aware of any existing mathematical counter-proof for collaborative clustering.

E. KKT calculus

In this subsection, we show the calculus to solve the optimization with the Karush-Kuhn-Tucker conditions: given the $\beta_{j,i} \geq 0$ and $p \in \mathbb{N}^*$, we are trying to find the matrix $T = \{\tau_{j,i}\}_{J \times J}$ solving the following optimization problem:

$$\begin{aligned} \underset{T}{\text{minimize}} \quad & -\sum_{i=1}^{J}\sum_{j\neq i} \tau_{j,i} \cdot \beta_{j,i} \\ \text{subject to} \quad & \sum_{j\neq i}^{J} (\tau_{j,i})^p = 1, \qquad \forall i, \\ & \tau_{j,i} \geq 0 \qquad \forall (i,j). \end{aligned} \tag{17}$$

From the previous system, by using Lagrange multipliers, we get the following KKT conditions:

$$\forall (i,j), i \neq j \begin{cases} (1) & \tau_{j,i} \geq 0 \\ (2) & \sum_{j\neq i}^{J} (\tau_{j,i})^p = 1 \\ (3) & \lambda_{j,i} \geq 0 \\ (4) & \tau_{j,i} \cdot \lambda_{j,i} = 0 \\ (5) & -\beta_{j,i} - \lambda_{j,i} + \nu_i \cdot \left(p \cdot (\tau_{j,i})^{p-1}\right) = 0 \end{cases}$$

For now, we will ignore the case $p = 1$ to which we will come back later. Let's begin by considering the case where $\lambda_{j,i} \neq 0$ in (4). Then, we would have $\tau_{j,i} = 0$ and with (5): $\beta_{j,i} = -\lambda_{j,i} \leq 0$. Since the $\beta_{j,i}$ have been defined as non-negative, this case is not possible, therefore we will only consider the case $\tau_{j,i} \neq 0$ and $\lambda_{j,i} = 0$. Then, with (5), we have:

$$\tau_{j,i} = \left(\frac{\beta_{j,i}}{p \cdot \nu_i}\right)^{\frac{1}{p-1}} \tag{18}$$

From Equation (18) and (2), we have:

$$1 = (p \cdot \nu_i)^{\frac{-p}{p-1}} \sum_{j\neq i} (\beta_{j,i})^{\frac{p}{p-1}} = (\nu_i)^{\frac{-p}{p-1}} \sum_{j\neq i} \left(\frac{\beta_{j,i}}{p}\right)^{\frac{p}{p-1}} \tag{19}$$

Then we can write:

$$\nu_i = \left(\frac{1}{\sum_{j\neq i}\left(\frac{\beta_{j,i}}{p}\right)^{\frac{p}{p-1}}}\right)^{-\frac{p-1}{p}} = \frac{1}{p}\left(\sum_{j\neq i}(\beta_{j,i})^{\frac{p}{p-1}}\right)^{\frac{p-1}{p}} \tag{20}$$

Then by injecting the expression of ν_i into Equation (18), $\forall (i,j), i \neq j, p > 1$ we have:

$$\tau_{j,i} = \frac{(\beta_{j,i})^{\frac{1}{p-1}}}{(\sum_{k\neq i}^{J}(\beta_{k,i})^{\frac{p}{p-1}})^{\frac{1}{p}}} \tag{21}$$

And thus we have proved the result of Equation (12).

We will now come back to the case where $p = 1$. In this case, the system obtained from the KKT conditions is a bit different:

$$\forall (i,j), i \neq j \begin{cases} (1) & \tau_{j,i} \geq 0 \\ (2) & \sum_{j\neq i}^{J} (\tau_{j,i}) = 1 \\ (3) & \lambda_{j,i} \geq 0 \\ (4) & \tau_{j,i} \cdot \lambda_{j,i} = 0 \\ (5) & -\beta_{j,i} - \lambda_{j,i} + \nu_i = 0 \end{cases} \tag{22}$$

With (3) and (5), we have:

$$\lambda_{j,i} = \nu_i - \beta_{j,i} \geq 0 \tag{23}$$

Let's suppose that there exists k so that $\tau_{k,i} > 0$. Then with (4), we have: $\lambda_{k,i} = 0$. And with Equation (23), we have: $\nu_i = \beta_{k,i}$. Then, using this information we can say that:

$$\forall j \neq k \begin{cases} \tau_{j,i} \neq 0 \implies \beta_{j,i} = \beta_{k,i} \implies \lambda_{j,i} = 0 \\ \tau_{j,i} = 0 \implies \lambda_{j,i} = \beta_{k,i} - \beta_{j,i} \geq 0 \end{cases} \tag{24}$$

From the second line of Equation (24), we can conclude the following:

$$\tau_{j,i} \neq 0 \implies \beta_{j,i} = \max_k \beta_{k,i} \tag{25}$$

Then, using conditions (4) and (5), we have:

$$\tau_{j,i}(\nu_i - \beta_{j,i}) = 0 \tag{26}$$

Summing Equation (26) over j and use (2), we obtain:

$$\nu_i = \sum_{j \neq i}^{J} \tau_{j,i} \cdot \beta_{j,i} \tag{27}$$

For Equation (27) to be correct while respecting the constraints given in Equations (24) and (25), the only solution is:

$$\forall j \neq i, \tau_{j,i} = \begin{cases} \frac{1}{Card(\beta_{j,i} = max_k(\beta_{k,i}))} & \text{if } \beta_{j,i} = max_k(\beta_{k,i}) \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

We have therefore proved the result shown in Equation (11).

IV. Experimental Results

A. Analysis of the weight evolution

In a first experiment, we sought to assess the behavior of our proposed weighting method with two goals: 1) Checking that the weights worked as intended with heavier weights given to algorithms with a lower diversity. 2) Assessing that the dynamic weights would stabilize at some point and thus would not hinder the convergence process of the collaborative framework.

To do so, we used the Waveform data set from the UCI website:

- *Waveform data set*: This data set consists of 5,000 instances divided into 3 classes. The original base included 40 variables, 19 are all noise attributes with mean 0 and variance 1. Each class is generated from a combination of 2 of 3 "base" waves.

Our experiment was the following: We ran 5 regular EM algorithms (GMM) in parallel over different versions of the data set: Two with almost no noisy variables, two with a moderate number of noisy variables, and one with a lot of noisy variables. At the end of this initial step we had results that were more or less good and close to each others depending on the number of noisy variables. We then started the collaborative step from these results and ran our proposed method with five EM algorithms collaborating together. During each iteration and for each algorithm, we observed the absolute average evolution of the weights $\tau_{j,i}$ they were giving to the other collaborators. For this experiment, we used $p = 2$ (See Equation (10)) and $\lambda = 0.5$.

In Figure 2 we show the absolute average weight evolution per iteration for the five EM algorithms. Each curve represent one of the collaborator.

Here is our interpretation of this graphics: First as one can see, the weight slowly stops evolving over time. As displayed on the diagram this is not a strict convergence, but a convergence on average. This result makes sense since the weights are based on a similarity measure between the solutions and because in our algorithm the system global entropy also converges on average. Another interesting remark is that we can see that the algorithms that had the most noisy variable (starting the highest) had their weights changing a lot during the two first iterations but converged fast overall.

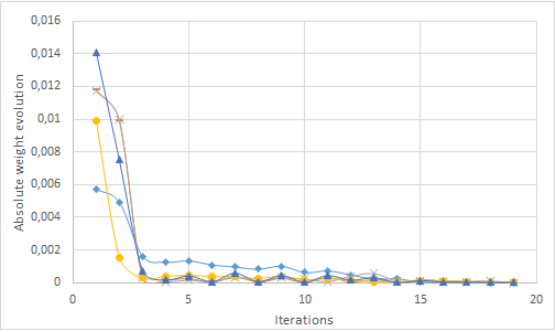

Fig. 2. Absolute average weight evolution per iteration for five EM algorithms using the Waveform data set

On the other hand, the weights for the collaborator that had no noise evolved less in the beginning but took more time to stabilize as the other solutions became slowly more similar. The internal behavior of our weight was the one expected from the KKT optimization with bigger weights given to more similar solutions.

B. Qualitative results

In our second experiment, we applied our proposed framework to several data sets from the UCI website -see Table I-, and assessed 2 indexes before and after collaboration. To do so, we ran a few simulations on several split data sets and checked the values of the Silhouette index [24] and the Davies-Bouldin index [25] at the end of the local step and the end of the collaborative step. Both indexes are clustering indexes that evaluate the quality of a clustering based the distance from each data to the nearest cluster centroid for the Silhouette index and on the cluster separation and compactness for the Davies-Bouldin index.

TABLE I
Data sets characteristics

Data Set	Instances	Attributes	Clusters
Wine	178	13	3
WDBC	569	30	2
Waveform	5.000	19	3
EColi	336	7	8
Image Segmentation	2,310	19	7

In our experimental protocol, for each data sets, we created 5 subsets by removing some of the attributes randomly. Each subset was then assigned to an EM algorithm using the gaussian mixture model. We then ran our collaborative framework as described in the previous section, with the parameters $p = 2$ and $\lambda = 0.5$. The results of this experiments are shown in Table II where the average result over a dozen simulations is shown in the main cells, as well as the best and worst results over all simulation that are shown between brackets.

As one can see, our proposed frameworks has overall positive results for two different clustering indexes. While it is true that the improvement after collaboration is not huge, our proposed method has the advantage of having very few cases of negative collaborations (results getting worst after

TABLE II
AVERAGE IMPROVEMENT AFTER COLLABORATION

Data Set	Silhouette Index	DB-Index
Wine	+1% $\begin{pmatrix} +3\% \\ -2\% \end{pmatrix}$	+1% $\begin{pmatrix} +2\% \\ -1\% \end{pmatrix}$
WDBC	+1% $\begin{pmatrix} +2\% \\ -1\% \end{pmatrix}$	+1% $\begin{pmatrix} +3\% \\ +1\% \end{pmatrix}$
Waveform	+4% $\begin{pmatrix} +15\% \\ +1\% \end{pmatrix}$	+1% $\begin{pmatrix} +2\% \\ -1\% \end{pmatrix}$
EColi	+9% $\begin{pmatrix} +34\% \\ -2\% \end{pmatrix}$	+4% $\begin{pmatrix} +14\% \\ +1\% \end{pmatrix}$
Image Segmentation	+4% $\begin{pmatrix} +12\% \\ 0\% \end{pmatrix}$	+7% $\begin{pmatrix} +15\% \\ -3\% \end{pmatrix}$

collaboration) when compared with several of our previously proposed methods that had no weighting system [8]. These results highlight that our weighting system has the positive aspect of reducing the cases of negative collaboration, but the inconvenient that it may also lead to results that on average are less impressive.

V. CONCLUSION

In this article, we have proposed and optimized a collaborative model applicable to the case of collaborative and multi-view clustering. Using this model we have demonstrated interesting properties that can be generalized to several frameworks, and in particular we have shown how diversity can be used to improve the results of unsupervised collaborative learning. The conclusion from the theoretical part of our article is that a lower diversity is a good criterion to choose collaborators because it tends to favor stable clustering solutions, and stability is a good unsupervised quality criterion to find the intrinsic structures in a data set.

Our proposed collaborative framework as well as our weighting model based on diversity have shown to be compatible with other collaborative models, and gave fair results in the experimental part that show a reduced risk of negative collaboration.

Possible extensions for this work would include a more complete study on the stability properties of collaborative frameworks as well as a possible generalization to other cases of collaborative frameworks such as vertical collaboration frameworks.

ACKNOWLEDGMENTS

The authors would like to thank Professor Younès Bennani and Professor Antoine Cornuéjols who have both been the initiators of this work on collaborative clustering.

REFERENCES

[1] W. Pedrycz, "Collaborative fuzzy clustering," *Pattern Recognition Letters*, vol. 23, no. 14, pp. 1675–1686, 2002.

[2] N. Grozavu and Y. Bennani, "Topological collaborative clustering," *Australian Journal of Intelligent Information Processing Systems*, vol. 12, no. 3, 2010.

[3] M. Ghassany, N. Grozavu, and Y. Bennani, "Collaborative clustering using prototype-based techniques." *International Journal of Computational Intelligence and Applications*, vol. 11, no. 3, 2012.

[4] G. Forestier, C. Wemmert, and P. Gancarski, "Collaborative multi-strategical classification for object-oriented image analysis," in *Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications in conjunction with IbPRIA*, Jun. 2007, pp. 80–90.

[5] S. Bickel and T. Scheffer, "Estimation of mixture models using co-em," in *Machine Learning: ECML 2005, 16th European Conference on Machine Learning, 2005, Proceedings*, ser. Lecture Notes in Computer Science, vol. 3720. Springer, 2005, pp. 35–46.

[6] C. Wemmert and P. Gancarski, "A multi-view voting method to combine unsupervised classifications," *Artificial Intelligence and Applications, Malaga, Spain,*, pp. 447 – 452, 2002.

[7] J. Sublemontier, "Unsupervised collaborative boosting of clustering: An unifying framework for multi-view clustering, multiple consensus clusterings and alternative clustering," in *The 2013 International Joint Conference on Neural Networks, IJCNN 2013*. IEEE, 2013, pp. 1–8.

[8] J. Sublime, N. Grozavu, Y. Bennani, and A. Cornuéjols, "Collaborative clustering with heterogeneous algorithms," in *2015 International Joint Conference on Neural Networks, IJCNN 2015*, 2015.

[9] B. Depaire, R. Falcon, K. Vanhoof, and G. Wets, "PSO Driven Collaborative Clustering: a Clustering Algorithm for Ubiquitous Environments," *Intelligent Data Analysis*, vol. 15, pp. 49–68, January 2011.

[10] J. Sublime, N. Grozavu, G. Cabanes, Y. Bennani, and A. Cornuéjols, "From horizontal to vertical collaborative clustering using generative topographic maps," *International Journal of Hybrid Intelligent Systems*, vol. 12, no. 4, 2016.

[11] A. Zimek and J. Vreeken, "The blind men and the elephant: on meeting the problem of multiple truths in data from clustering and pattern mining perspectives," *Machine Learning*, vol. 98, no. 1-2, pp. 121–155, 2015.

[12] S. Bickel and T. Scheffer, "Multi-view clustering," in *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004)*. IEEE Computer Society, 2004, pp. 19–26.

[13] B. Y. Grozavu N., "Topological collaborative clustering," *in LNCS Springer of ICONIP'10 : 17th International Conference on Neural Information Processing*, 2010.

[14] G. Forestier, P. Gancarski, and C. Wemmert, "Collaborative clustering with background knowledge." *Data & Knowledge Engineering*, vol. 69, no. 2, pp. 211–228, 2010.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[16] X.-N. Wang, J.-M. Wei, H. Jin, G. Yu, and H.-W. Zhang, "Probabilistic confusion entropy for evaluating classifiers," *Entropy*, vol. 15, no. 11, pp. 4969–4992, 2013.

[17] J.-M. Wei, X.-J. Yuan, Q.-H. Hu, and S.-Q. Wang, "A novel measure for evaluating classifiers." *Expert System Applications*, vol. 37, no. 5, pp. 3799–3809, 2010.

[18] C. F. J. Wu, "On the Convergence Properties of the EM Algorithm," *The annals of statistics*, vol. 11, no. 1, pp. 95–103, 1983.

[19] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Proceedings of 2nd Berkeley Symposium*, B. U. of California Press, Ed., 1951, pp. 481–492.

[20] N. Grozavu, G. Cabanes, and Y. Bennani, "Diversity analysis in collaborative clustering," in *2014 International Joint Conference on Neural Networks, IJCNN 2014, Beijing, China, July 6-11, 2014*, 2014, pp. 1754–1761.

[21] P. Rastin, G. Cabanes, N. Grozavu, and Y. Bennani, "Collaborative clustering: How to select the optimal collaborators?" in *IEEE Symposium Series on Computational Intelligence, SSCI 2015*. IEEE, 2015, pp. 787–794.

[22] S. Ben-David, U. von Luxburg, and D. Pl, "A sober look at clustering stability," in *Learning Theory*, ser. Lecture Notes in Computer Science, G. Lugosi and H. Simon, Eds. Springer Berlin Heidelberg, 2006, vol. 4005, pp. 5–19.

[23] U. von Luxburg, "Clustering stability: An overview," *Foundations and Trends in Machine Learning*, vol. 2, no. 3, pp. 235–274, Mar. 2010.

[24] R. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics.*, vol. 20, pp. 53–65, 1987.

[25] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 224–227, Feb. 1979.