Rethinking Collaborative Clustering: A Practical and Theoretical Study within the Realm of Multi-View Clustering

Pierre-Alexandre Murena, Jérémie Sublime, Basarab Matei

Rethinking Collaborative Clustering: A Practical and Theoretical Study within the Realm of Multi-View Clustering

Pierre-Alexandre Murena[1], Jérémie Sublime[2,3], and Basarab Matei[3]

[1] Helsinki Institute for Information Technology HIIT
Department of Computer Science, Aalto University, Helsinki, Finland
pierre-alexandre.murena@aalto.fi
[2] ISEP, School of Engineering
10 rue de Vanves, 92130 Issy-Les-Moulineaux, France
jsublime@isep.fr
[3] LIPN - CNRS UMR 7030, LaMSN - Sorbonne Paris North University
93210 St Denis, France
[sublime][matei]@lipn.univ-paris13.fr

Abstract. With distributed and multi-view data being more and more ubiquitous, the last 20 years have seen a surge in the development of new multi-view methods. In unsupervised learning, these are usually classified under the paradigm of multi-view clustering: A broad family of clustering algorithms that tackle data from multiple sources with various goals and constraints. Methods known as collaborative clustering algorithms are also a part of this family. Whereas other multi-view algorithms produce a unique consensus solution based on the properties of the local views, collaborative clustering algorithms aim to adapt the local algorithms so that they can exchange information and improve their local solutions during the multi-view phase, but still produce their own distinct local solutions.

In this chapter, we study the connections that collaborative clustering shares with both multi-view clustering and unsupervised ensemble learning. We do so by addressing both practical and theoretical aspects: First we address the formal definition of what is collaborative clustering as well as its practical applications. By doing so, we demonstrate that pretty much everything called collaborative clustering in the literature is either a specific case of multi-view clustering, or misnamed unsupervised ensemble learning. Then, we address the properties of collaborative clustering methods, and in particular we adapt the notion of clustering stability and propose a bound for collaborative clustering methods. Finally, we discuss how some of the properties of collaborative clustering studied in this chapter can be adapted to broader contexts of multi-view clustering and unsupervised ensemble learning.

Keywords: collaborative clustering · multi-view clustering · stability

1 Introduction

Clustering techniques play a central role in various part of data analysis and are key to finding important clues concerning the structure of a data sets. In fact, clustering is often considered to be the most commonly used tool for unsupervised exploratory data analysis. However due to an explosion both in the number of frequently occurring multi-view data (be it with organic views or with "artificial views" created by different feature extraction algorithms), and also the number and diversity of available clustering methods to tackle them, there has been a surge in the number of clustering methods that are either multi-view, multi-algorithm, or both.

While regular clustering itself presents its own challenges (the most common of which is to find which methods are "best" for a given task), these new paradigms, involving multiples views and sometimes multiple clustering algorithms, make the problem even more complex. Yet, despite an extensive literature on multi-view clustering, unsupervised ensemble learning and collaborative clustering, very little is known about the theoretical fundations of clustering methods belonging to these families of algorithms. Furthermore, while it is easy to tell the difference between unsupervised ensemble learning and multi-view clustering, the third family of algorithm –namely collaborative clustering [33,23]– which is also the most recent of the three is a lot more problematic in the sense that this notion is ill-defined in the literature and that it shares many similarities with both multi-view clustering and ensemble learning, both in terms of practical applications, but also when it comes to the algorithms used.

To address these issues with a special focus on collaborative clustering, in this chapter we propose the following main contributions:

- First, we propose formalization of the three notions of collaborative clustering, multi-view clustering and unsupervised ensemble learning. And from it, we clearly define what are the foundations of collaborative clustering.
- Second, we show that collaborative clustering is very much related and overlapping with multi-view clustering and ensemble learning. We also show that ultimately multi-view clustering and collaborative clustering are equivalent to regular clustering.
- We define the notion of pure collaborative clustering algorithms, a notion that guaranties the definitive aspect of the results produced by such algorithms.
- We introduce two key notions, namely novelty and consistency, that can be used as quality metric for both collaborative clustering and unsupervised ensemble learning.
- Then, we lay the groundwork for a theory of multi-view and collaborative clustering approaches, by extending the notion of clustering stability originally proposed by Ben-David et al. [2]. We propose three complementary characterizations of stable multi-view clustering algorithms, involving in particular the stability of the local algorithms and some properties of the collaboration.

We tackle these issues all the while trying to keep a theoretical setting as generic as possible.

Chapter organisation. This chapter is organized as follows: In Section 2 we introduce the state of the art and current situation of collaborative clustering. Section 3 introduces the main notations and concepts that will be used in this chapter, and it formalizes some definitions of collaborative and multi-view clustering. Based on these formal definitions, it introduces the interconnections between all these tasks. Section 4 presents the main contributions of our chapter, presentation and the study of several theoretical aspects and properties of collaborative clustering. Section 5 lists various open questions subsequent to the theoretical framework we presented. We conclude the chapter with a brief discussion on how the formal classification of these fields could help future research.

2 Collaborative clustering: State of the art of a polymorphic notion with very diverse applications

Before we start to describe the different variations of collaborative clustering, we first present,in Table 1 below, the full spectrum of methods often falling under the umbrella of collaborative clustering in the literature. We detail each subcategory of methods according to their other denominations in the literature, its characteristics and its inputs. Please note that the term "different algorithms" may include cases where the same algorithm is used with different number of cluster, or different parameters. This classification is our personal view of the field, and is in no way fixed. Nevertheless, we needed it to make clear what we were referring to when we mention these different notions throughout the chapter.

2.1 Evolution of the notion of collaborative clustering

Collaborative clustering is a term that was first coined by Pedrycz [33] to describe a clustering framework whose aim is to find common structures in several data subsets. It essentially involves an active way of jointly forming or refining clusters through exchanges and information communication between the different data sites with the goal of reconciling as many differences as possible [35]. In its original design collaborative clustering was not aimed at a specific application and was essentially targeted at fuzzy clustering [32,37] and rough set clustering [30] applications.

As the original idea gained in visibility, the notion of collaborative clustering was re-used by several research teams who thought of various possible applications. This led to a diversification of what can be considered collaborative clustering, but also to the question of its place as a tool or as a field compared with already established problems such as multi-view clustering [3,48] and unsupervised ensemble learning [38].

Method family	Characteristics	Input	References
Multi-view clustering	Organic views of the same objects under different features	Data only	[48]
Horizontal collaboration	Same object, different features	Data, local partition, local algorithms	[23,47,21]
Multi-algorithm collaboration	Same objects, same features, different algorithms	Data, local partition, local algorithms	[44,31,42,39]
Vertical collaboration	Different objects, same features	Data, local partitions, local algorithms	[23,21,40]
Partition collaboration	Same objects, different features	Data local partitions	[16,17,14]
Ensemble learning	Same objects, same features, different algorithms	Local partitions only, no data, no algorithm	[15,38,20]

Table 1: The spectrum of notions sometimes falling under the term collaborative clustering.

A first attempt at formalizing collaborative clustering and categorizing its different uses was made by Grozavu and Bennani [23]. In their work the authors describe collaborative clustering as a two-steps process where partitions are first created locally in each data site, and then refined during a collaborative step which involves information communication between the different sites with a goal of mutual improvement. The full process is illustrated in Figure 1. In the same paper, the authors also distinguish two types of collaborative clustering: *horizontal collaboration* (previously mentioned but not fully formalized in [33] and [34]) which involves the same data with attributes spread over different sites, and *vertical collaboration* where different data with the sames attributes are spread across different sites. The algorithms developed following these ideas are heavily inspired by the original work of Pedrycz et al. and rely on the same principles applied to self-organizing maps [28] and generative topographic mapping [4] instead of the fuzzy C-Means algorithm [10].

It is worth mentioning that the term *horizontal collaborative clustering* had already been used prior to their work to describe a version of collaborative fuzzy C-Means applied to what can reasonably be considered a multi-view clustering application [47].

Following the same idea as *horizontal collaboration*, a large number of methods [7,12,21,22,24,42] have since been developed with applications that fall under the umbrella of what is traditionally known as multi-view clustering. Sometimes the multiple views will be organic to the data, and sometimes it will be something more artificial like different features or views artificially created by feature extracting algorithms. The common point of these horizontal collaborative methods

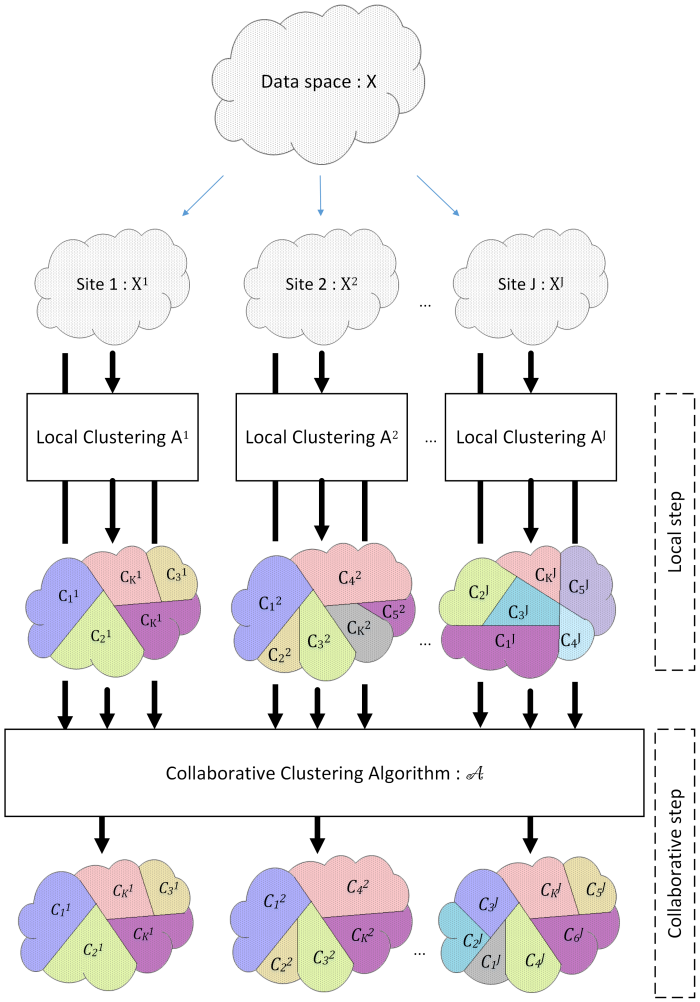

Fig. 1: Graphical definition of collaborative clustering as it is defined by Grozavu and Bennani [23]: local clustering algorithms produce clustering partitions locally during the local step. During the collaborative step, these local partitions are passed alongside the original data and local algorithms to produce improved partitions in each site after the collaboration process. This graphical definition works for both horizontal and vertical collaboration.

is that they perform the clustering of the same data spread across several sites. Differences exist however as some of them aim at a single consensus result, while others highlights that the specificity of collaborative clustering is that a consensus should not be the main goal [21]. In other cases, collaborative clustering is directly referenced as being multi-view clustering [22,25,43]. And sometimes

the authors use collaborative clustering for multi-view application, but prefer to opt for a neutral name [31]. While horizontal collaborative clustering and multi-view clustering appear to be the same thing, a few authors point out that on the one hand most multi-view clustering methods have access to all the views, while on the other hand collaborative clustering with its local algorithms and exchanges of information offer more possibilities for data anonymization and the control of privacy [26,49,47]. Furthermore, it is worth mentioning that collaborative clustering in its so called *horizontal from* encompasses both real multi-view applications [23,47,21], and also cases of multi-algorithms collaboration where several algorithms tackle the same data without any views [44,31,42,39]. In the second case, this is in a way similar to boosting techniques but for unsupervised methods.

Finally, another common recent use of collaborative clustering for multi-view application is its application to the clustering of data sets spread across networks under various constraints [45,41,11]. As with the privacy issues, collaborative clustering with its local methods and information exchanges offers more possibilities than classical multi-view clustering framework for this type of applications.

The second form described by Grozavu and Bennani, *vertical collaboration*, appears to be less common in the literature [21,40]: In this case we consider different samples of the same initial database spread across several sites. It is likely that this term is less common in the literature simply because it matches the definition of *federated learning*, which ironically is sometimes coined under *collaborative learning*. However, unlike vertical collaborative clustering which so far has only been tested on mostly outdated algorithms (K-Means, GTM, and SOM), federated learning is currently researched for deep learning and using block chain technology [5,1,36,9]. Indeed, in the case where we consider different data distributed across several sites and with nearly identical distribution, vertical collaborative clustering can be seen as a form of unsupervised federated learning. On the other hands, if the distributions are too different, this becomes transfer learning for which the current collaborative clustering methods are ill-adapted.

Lastly, we can mention hybrid collaborative clustering [13], a mix between horizontal and vertical collaborative clustering with little to no practical application.

2.2 Remarkable branches of collaborative clustering and applications that blur the lines between ensemble learning and multi-view clustering

While we have presented an almost chronological evolution of the notion of collaborative clustering, it is worth mentioning that not all algorithms coined as "collaborative clustering" fall under these definitions. There is indeed a whole spectrum of collaborative clustering algorithm ranging from the mostly multi-view applications that we have seen, to collaborations between algorithms working on the same data subsets, and even some methods discarding the original

data and algorithms altogether to have a "collaboration" only between partitions, thus drifting towards what seems to be ensemble learning.

We can for instance mention group of methods and algorithms described by their authors as collaborative clustering, but that differ slightly from both the original idea by Pedrycz and the notion of *horizontal collaboration* coined by Grozavu and Bennani. In [46,15,20,16,17], the authors propose various iterations of the SAMARAH method [19]. Like in Grozavu and Bennani [23], they define collaborative clustering as a two-steps process where results are first produced by local algorithms, and are then refined. For some applications, several algorithms are applied to the same source data [15,20], and in other multiple views or sources for the same data are considered [16,17,14].In their case, they don't use collaborative clustering for multi-view learning but to merge the results of several and potentially different clustering algorithms applied to the exact same data and attributes. Furthermore, unlike in previous collaborative methods described previously in this step of the art, the algorithms are completely removed from the collaborative step and only the local partitions are kept to search for a consensus. As one can see, this type of collaborative clustering is identical to what is known as unsupervised ensemble learning [38]. It is worth mentioning that the strength of this approach is that is it compatible with any clustering algorithm, and this is due to the removal of the algorithms from the collaborative step.

The collision between collaborative clustering and ensemble learning was further increased by the third attempt at formalizing collaborative clustering by a group researchers from several teams working on the subject [8]. Furthermore, this approach of giving less importance to the local algorithm and to focus more on the partitions appears to be a growing trend too in collaborative clustering for multi-view applications as more and more authors appear to favor it in recently produced collaborative clustering algorithms [31,18].

As one can see from the state of the art and from Table 1, collaborative clustering is a polymorphic notion whose main applications in the literature range from multi-view clustering to ensemble learning. Yet, it is also obvious that many of the methods under the name *collaborative clustering* share common points that are unique to them. One of the goals of this chapter is to address the overlap and confusion that may exist between the 3 notions and we will do so in section 3 by formally defining what we consider to be the properties specific to each type of method.

The second goal of this chapter, detailed in Section 4, is to introduce a formal understanding of collaborative clustering, which appears to be missing in all papers from the state of the art that we have mentioned previously: With dozens of methods described, none of them so far has studied the theoretical properties of collaborative clustering such as the question of its stability, the question of the consistency between the original local partitions and the collaborative result(s), and potential guarantees that it will produce novel solutions compared with the local ones, a property important for both multi-view and ensemble learning applications. We will formalize and address these notions for

collaborative clustering in general, and then we will discuss to what degree they can be extended to multi-view clustering and unsupervised ensemble learning.

3 Distinguishing regular clustering, collaborative clustering, multi-view clustering and unsupervised ensemble learning

3.1 Notations

In the remainder of this work, we will use the notations presented in this subsection.

3.1.1 Regular clustering. Let us consider that all clustering methods – regular, collaborative, multi-view or otherwise– will be applied a data space $\mathbb{X}$ endowed with a probability measure P. If $\mathbb{X}$ is a metric space, let ℓ be its metric. In the following, let $S = \{x_1, \cdots, x_m\}$ be a sample of size m drawn i.i.d. from $(\mathbb{X}, P, \Sigma)$, where Σ is the set of finite partitions of $\mathcal{X} \subseteq \mathbb{X}$.

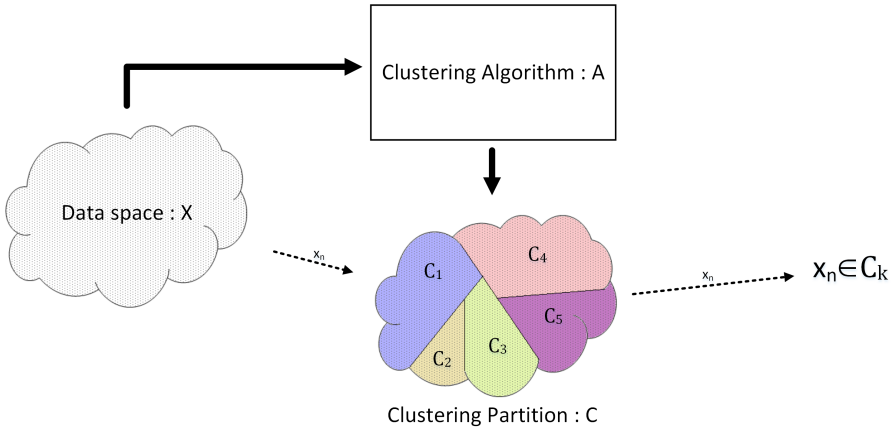

Fig. 2: Graphical definition of the notions of clustering algorithm, clustering partition, and cluster in the case of *regular* clustering

In this work, we emphasize the difference between a clustering or partition, and the clustering algorithm that produces this partition: In regular clustering, a *clustering* C of a subset $X \subseteq \mathbb{X}$ is a function $C : X \to \mathbb{N}$ which to any of said subset X associates a solution vector in the form of matching clusters $S = C(X)$. Individual *clusters* are then defined by: $C_i = C^{-1}(\{i\}) = \{x \in X; C(x) = i\}$. The *clustering algorithm* A is the function which produces the clustering partition, i.e. a function that computes a clustering of X for any finite sample $\mathcal{S} \subseteq X$, so that $A : X \mapsto C$.

These definitions are graphically explained in Figure 2. The proposed definition differs from a more standard view of clustering, in that they aim to produce clustering partition for the whole space and not only for the dataset of interest. Note however that this specific case can be retrieved easily from the definitions, by defining the total space to correspond to the dataset. This trivial case is weaker, in the sense that the theoretical analysis proposed below does not apply to it.

Example 1. Consider data representing individuals, represented by two features, the height and the weight. Here, the *data space* $\mathbb{X}$ is $\mathbb{X} = \mathbb{R}^2$. Consider a population distribution P over $\mathbb{X}$, and a sample of m individuals drawn from distribution P. The K-means algorithm is a *clustering algorithm* which, given the sample, produces the *partition* defined as a Voronoi diagram associated to some optimal seeds, the *means* computed by the algorithm.

3.1.2 Reminders on risk optimization schemes. A large class of clustering algorithms choose the clustering by optimizing some risk function. The large class of center based algorithms falls into this category, and spectral clustering can also be interpreted in this way. Risk optimization schemes are an important clustering notion discussed by Ben David et al. [2]. We will also use them when discussing clustering stability for both regular, multi-view and collaborative clustering. This subsection reviews some of the basics needed to understand our work.

Definition 1. ***(Risk optimization scheme)*** *A risk optimization scheme is defined by a quadruple $(\mathbb{X}, \Sigma, \mathcal{P}, \mathcal{R})$, where $\mathbb{X}$ is some domain set, Σ is a set of legal clusterings of $\mathbb{X}$, and $\mathcal{P}$ is a set of probability distributions over $\mathbb{X}$, and $\mathcal{R} : \mathcal{P} \times \Sigma \to [0, \infty)$ is an objective function (or risk) that the clustering algorithm aims to minimize. We denote $opt(\mathcal{P}) := inf_{C \in \Sigma} \mathcal{R}(\mathcal{P}, C)$. For a sample $\mathcal{X} \subseteq \mathbb{X}$, we call $\mathcal{R}(\mathcal{P}_X, C)$ the empirical risk of C, where $\mathcal{P}_X$ is the uniform probability distribution over X. A clustering algorithm A is called R-minimizing, if $\mathcal{R}(\mathcal{P}_X, A(X)) = opt(\mathcal{P}_X)$, for any sample X.*

Example 2. Generic examples regarding risk optimization schemes usually use center-based clustering algorithms such as K-means and K-medians, and any K-medoid based algorithm fuzzy or not. Those algorithms pick a set of k center points $c_1, \ldots, c_k$ and then assign each data point in the metric space to the closest center point. Such a clustering is a k-cell Voronoi diagram over (X, ℓ), ℓ being the metric on the space X. To choose the centers, the K-Means algorithm minimizes the following risk function:

$$R(P, C) = \mathop{\mathbb{E}}_{x \sim P} \left[\min_{1 \leq i \leq k} (\ell(x, c_i))^2 | \mathrm{Vor}(c_1, c_2, \ldots, c_k) \right]$$

while the K-medians algorithm minimizes:

$$R(P, C) = \mathop{\mathbb{E}}_{x \sim P} \left[\min_{1 \leq i \leq k} (\ell(x, c_i)) | \mathrm{Vor}(c_1, c_2, \ldots, c_k) \right]$$

where $\mathrm{Vor}(c_1, c_2, \ldots, c_k)$ is the minimization diagram of the k functions $(\ell(x, c_i))^2$ respectively $\ell(x, c_i)$, $1 \leq i \leq k$.

Usually, risk-minimizing algorithms are meant to converge to the true risk as the sample sizes grow to infinity, which is formalized by the notion of *risk convergence*.

Definition 2. ***(Risk convergence)*** *Let A be an R-minimizing clustering algorithm.*

We say that A is risk converging, if for every $\epsilon > 0$ and every $\delta \in (0, 1)$ there is m_0 such that for all $m > m_0$, $\quad \Pr_{S \sim P^m}[R(P, A(S)) < opt(P) + \epsilon] > 1 - \delta$ for any probability distribution $P \in \mathcal{P}_X$.

For example, Ben-David et al. [2] have shown that, on bounded subset of $\mathbb{R}^d$ with Euclidean metric, both K-means and K-medians minimize risk from samples.

Note that this definition represents, from measure theory point of view, the almost everywhere convergence.

3.1.3 Notations in the multi-view context. The problem of interest in this chapter involves clustering algorithms that can be applied to several data sites. This setting includes all applications in Table 1. Therefore, we consider a data space $\mathbb{X}$ which is decomposed into the product $\mathbb{X} = \mathbb{X}^1 \times \cdots \times \mathbb{X}^J$ of J spaces $\mathbb{X}^j$, that may or may not overlap depending on the application. We will call the spaces $\mathbb{X}^j$ *view spaces* or simply *views*. The interdependence between the views is not solely contained in the definition of the different views $\mathbb{X}^j$, but also in the probability distribution P over the whole space $\mathbb{X}$.

For the remainder of this chapter, we will use strict notation conventions. Upper indexes will usually refer to the view or data site index, and lower indexes to individual data or clusters in specific views. For instance, $\mathcal{C}_k^j$ would be the k-th cluster of data site j, x_n^j the n-th data element of site j, etc. For simplicity purposes, we will sometimes use the notation $O^{1:J}$ to designate the tuple $O^1, \ldots, O^J$, where O can be any object distributed among the J views (including algorithms, partitions or data).

Example 3. Consider the data described in Example 1. Consider that now data are available in two different sites, corresponding to two view spaces (i.e. $J = 2$). In the first site, both height and weight are observed ($\mathbb{X}^1 = \mathbb{R}^2$), while only the height is observed in the second site ($\mathbb{X}^2 = \mathbb{R}$). In this multi-view description, the data space $\mathbb{X}$ is then defined as $\mathbb{X} = \mathbb{X}^1 \times \mathbb{X}^2 = \mathbb{R}^2 \times \mathbb{R}$. The total distribution on $\mathbb{X}$ must satisfy the equality of the height between $\mathbb{X}^1$ and $\mathbb{X}^2$ (if $x \sim P$, then feature 0 of x^1 is equal to x^2).

3.2 Definitions, context, and practical setting

We now formalize the different tasks presented in Table 1 and show, based on their definitions, how interconnected they are.

3.2.1 Multi-view clustering partition. From the definitions above, we define the notion of multi-view clustering partition as follows:

Definition 3. *(**Multi-view clustering partition**) A multi-view partition is defined as a combination of local clustering in the following sense: A multi-view clustering $\mathcal{C}$ of the subset $X \subseteq \mathbb{X}$ is a function $\mathcal{C} : X \to \mathbb{N}^q$, where $q \in \{1, J\}$ is called the* index of the partition *and indicates whether the goal is to reach a single consensus solution ($q = 1$) or to keep independent clustering in each view ($q = J$).*

As one can see, the very broad definition of a multi-view partition given above covers all cases of collaborative and multi-view clustering, with the different objectives of reaching a consensus between the views ($q = 1$) and refining the partitions produced for each view ($q = J$). Both cases are important depending on the context. When the views describe features of same objects but the goal is to have groups of similar objects, then a consensus is needed: In Example 3, it would be the case if the goal is to group individuals with similar morphological traits. Conversely, refining the results of the views is important when the goal is not to propose a unique group for each object, but one group per view: In Example 3, the joint information of height and weight can provide refined information about the height distribution, but the clustering of heights must still provide a description of the height characteristics only.

A very important observation here is that a multi-view clustering partition can be interpreted as a regular clustering partition. This is clear when $q = 1$, since the definitions of regular and multi-view partitions match completely. When $q = J$, this result is based on the observation that $\mathbb{N}$ and $\mathbb{N}^J$ are equipotent (i.e. there exists a bijection $\nu : \mathbb{N}^J \to \mathbb{N}$, for instance the Cantor pairing function). For instance, saying that a point $x = (x^1, \ldots, x^J) \in \mathbb{X}$ is associated to clusters $(c^1, \ldots, c^J)$ in the multi-view setting, is equivalent to considering that x is associated to cluster $\nu(c^1, \ldots, c^J)$ in a regular clustering of $\mathbb{X}$.

Example 4. Consider $\mathbb{X} = \mathbb{X}^1 \times \mathbb{X}^2$. In a case where both spaces are partitioned into 2 clusters (namely $\mathcal{C}_0^1$ and $\mathcal{C}_1^1$ for $\mathbb{X}^1$ and $\mathcal{C}_0^2$ and $\mathcal{C}_1^2$ for $\mathbb{X}^2$), this can be represented as a partition of $\mathbb{X}$ into four clusters: $\mathcal{C}_0 = \mathcal{C}_0^1 \times \mathcal{C}_0^2$, $\mathcal{C}_1 = \mathcal{C}_0^1 \times \mathcal{C}_1^2$, $\mathcal{C}_2 = \mathcal{C}_1^1 \times \mathcal{C}_0^2$ and $\mathcal{C}_3 = \mathcal{C}_1^1 \times \mathcal{C}_1^2$.

Far from being anecdotal, this observation shows that any multi-view clustering sums up to a regular clustering. This result will be exploited further to extend the main property of stability to collaborative clustering (Theorem 1). The converse is not true though, since any partition of the total space $\mathbb{X} = \mathbb{X}^1 \times \ldots \times \mathbb{X}^J$ does not correspond to a multi-view partition. In order to correspond to a valid multi-view partition, the global partition needs to satisfy another additional property:

Proposition 1. *A global clustering partition $\mathcal{C}$ on $\mathbb{X} = \mathbb{X}^1 \times \ldots \times \mathbb{X}^J$ corresponds to a local multi-view partition $(\mathcal{C}^1, \ldots, \mathcal{C}^J)$ on the views $\mathbb{X}^1, \ldots, \mathbb{X}^J$ if and only if for all $j \in \{0, \ldots, J\}$ and for all $x^j \in \mathbb{X}^j$, all clusters of $\mathcal{C}$ containing a point x' with $x'^j = x^j$ have for projection over $\mathbb{X}^j$ the set $\mathcal{C}^j(x^j)$.*

Proof. Suppose first that the global partition $\mathcal{C}$ corresponds to the local multi-view partition $(\mathcal{C}^1, \dots, \mathcal{C}^J)$, i.e. there exists a bijection $\nu : \mathbb{N}^J \to \mathbb{N}$ such that, for all $c^1, \dots, c^J \leq 0$, $\mathcal{C}_{\nu(c^1,\dots,c^J)} = \mathcal{C}^1_{c^1} \times \dots \times \mathcal{C}^J_{c^J}$. Consider a view j and a point $x^j \in \mathbb{X}^j$. By construction, the clusters in $\mathcal{C}$ containing points with x^j as their j-th component are the clusters of the form $\mathcal{C}_{\nu(c^0,\dots,\mathcal{C}^j(x^j),\dots,c^J)}$ which have all $\mathcal{C}^j(x^j)$ as their projection on $\mathbb{X}^j$.

Suppose now the converse, and let us show that $\mathcal{C}$ corresponds to the multi-view partition $(\mathcal{C}^1, \dots, \mathcal{C}^J)$. From the hypothesis, we see that the projection of each cluster $\mathcal{C}_i$ onto $\mathbb{X}^j$ is the union of some clusters from $\mathcal{C}^j$. However, the clusters being disjoint, the projection of $\mathcal{C}_i$ onto $\mathbb{X}^j$ being equal to one cluster implies that the union contains one single element. Therefore, each cluster $\mathcal{C}_i$ is the Cartesian product of clusters in local views: $\mathcal{C}_i = \mathcal{C}^1_{c^1_i} \times \dots \times \mathcal{C}^J_{c^J_i}$ for some c^j_i. We must show that the function defined by $\nu(c^1_i, \dots, c^J_i) = i$ is bijective. It is direct to show that $\nu(n_1, \dots, n_J)$ is well defined for n_j lower than the number of clusters on $\mathcal{C}^j$ (this can be shown by considering $x = (x^1, \dots, x^J)$ with $x^j \in \mathcal{C}^j_{n_j}$). Then, $\nu(c^1_i, \dots, c^J_i) = \nu(c^1_i, \dots, c^J_i)$ implies $(c^1_i, \dots, c^J_i) = (c^1_i, \dots, c^J_i)$ since the clusters are distinct, which concludes the proof.

3.2.2 Multi-view and Collaborative Clustering Algorithms. Based on the notions introduced before, we can now formalize the various notions exposed in Table 1, in particular the notions of collaborative clustering algorithm, multi-view clustering algorithm and unsupervised ensemble learning.

Defining these notions requires understanding the main differences between them. Assessing *multi-view clustering* problems is direct: a multi-view clustering problem simply partitions the data based on observations of samples from the views $\mathbb{X}^j$ (Figure 3). For collaborative clustering algorithms, we will focus on algorithms that have at least the following property: the collaboration process should include local clustering algorithms exchanging information and must not be limited to only exchanging local partitions. We feel like this definition is the broadest we can have as it includes most algorithms developed by Pedrycz et al., as well as all algorithms falling under the definition of vertical and horizontal collaboration as defined by Grozavu and Bennani [23], and thus only excludes so called collaborative methods that are in fact unsupervised ensemble learning as they deal only with partitions fusion (see Figure 4).

Definition 4 (Multi-view and collaborative clustering algorithms). *Consider a total space $\mathbb{X} = \mathbb{X}^1 \times \dots \times \mathbb{X}^J$ and a subspace $X \subseteq \mathbb{X}$. Let S^j be a sample of X^j. Then a* ***collaborative clustering algorithm*** *is a mapping*

$$\mathcal{A}^{col} : (S^1, \dots S^J, A^1, \dots, A^J, C^1 \dots, C^J) \mapsto \mathcal{C} \tag{1}$$

and a ***multi-view clustering algorithm*** *is a mapping*

$$\mathcal{A}^{MV} : (S^1, \dots, S^J) \to \mathcal{C} \tag{2}$$

where A^j designates a clustering algorithm over $\mathbb{X}^j$ and C^j is a partition of X^j.

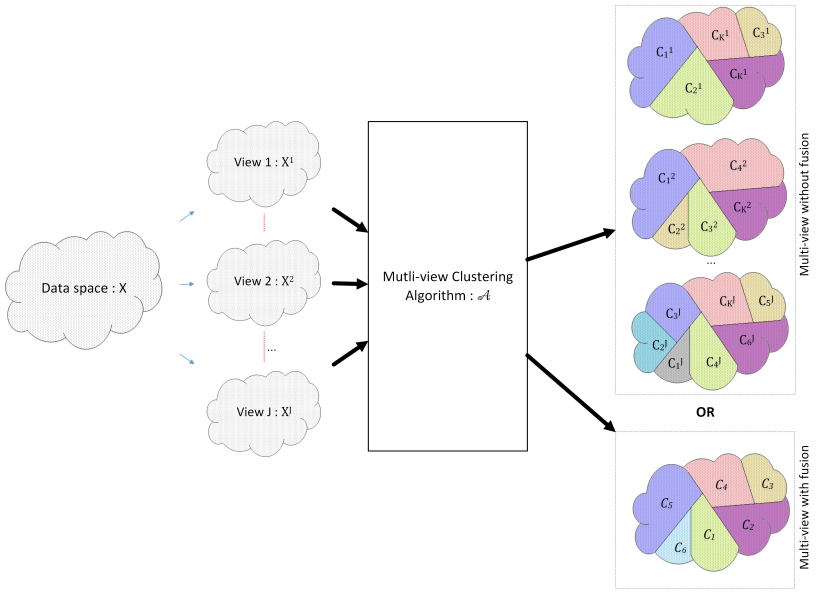

Fig. 3: Graphical definition of multi-view clustering: In this figure we display two possible cases, namely multi-view clustering without fusion, and multi-view clustering leading to a single consensus partition.

For collaborative clustering algorithms, we denote the local algorithms as index: $\mathcal{A}^{col}_{\langle A^1,\dots,A^J\rangle}(S^1,\dots,S^J,C^1,\dots,C^J)$.

These two definitions describe successfully the multi-view clustering, horizontal collaboration, multi-algorithm collaboration, vertical collaboration and partition collaboration families described in Table 1. We observe that many existing collaborative clustering methods suppose the application of a same clustering algorithm to the different views, which corresponds in essence to having $A^1,\dots,A^J$ belonging to a same class of algorithms[4]. Please note that Equation 1 is compatible with both horizontal and vertical collaborative clustering as it makes no assumptions about whether the full data space is cut into sub-sites alongside the features or the data themselves.

It is noticeable that the collaborative clustering algorithms are given local algorithms $A^1,\dots,A^J$ as input, and could be rewritten as functions of the form: $\mathcal{A}^{col} : (A^1,\dots,A^J) \mapsto \mathcal{C}$.

The role of local algorithms as inputs in collaborative clustering is twofold : They have an influence over the collaboration between the views. Intuitively, the decision of altering an optimal local partition to incorporate information from other views must be constrained by the biases of the local algorithm. This strategy is explicit in the multi-algorithm collaboration setting [31,39,42,44], where

[4] Formally, it would be incorrect to state that $A^1 = \dots = A^J$, since the algorithms A^j are defined relatively to different spaces $\mathbb{X}^j$ and are therefore of different natures.

the impact of a local algorithm intervenes as a penalization of a risk minimization objective by the information held in the produced partitions [31,44], as the core of a risk minimization scheme with a penalty for divergences between views [42], or as a bias in the selection of data for learning the local partitions from one step to another [39]. Noticeably, many collaborative clustering algorithms are thought to apply to one single class of local clustering algorithms, such as C-Means, Self-Organizing Maps or Generative Topographic Maps [12,23,35,21]. In this case, the nature of the local algorithms is directly exploited for the collaboration. In addition, following the idea of a 2-step process introduced by [23], where the algorithm is divided into the generation of local partitions and the refinement of these partitions, the local algorithms A^j are naturally involved in the first step. The support of these algorithms (i.e. the space of parameters on which they are properly defined)is then constrained to satisfy $C^j = A^j(S^j)$ for all j. It follows directly from Definition 4 that such collaborative clustering algorithms, once the local clustering algorithms are fixed, are strictly equivalent to multi-view clustering algorithms.

Proposition 2. *Let $\mathcal{A}^{col}$ be a collaborative clustering algorithm the support of which is restricted to satisfy $C^j = A^j(S^j)$. Given $A^1, \dots, A^J$, J fixed local clustering algorithms, the function*

$$S^1, \dots, S^J \mapsto \mathcal{A}^{col}_{\langle A^1, \dots, A^J \rangle} \left(S^1, \dots, S^J, C^1(X^1), \dots, C^J(X^J) \right)$$

is a multi-view clustering algorithm.

This proposition is a direct application of Definition 4. A consequence of this result is that, if we follow the definition of collaborative clustering as given by Grozavu and Bennani [23] where partitions should not be merged (See Figure 1), we have that collaborative clustering algorithms are a specific case of multi-view clustering algorithms and they produce multi-view partition $\mathcal{C}$ of the subset $\mathcal{X} \subseteq \mathbb{X}$ whose mapping follows the form $\mathcal{C} : \mathcal{X} \to \mathbb{N}^J$, given J algorithms collaborating together.

A very straightforward property of collaborative clustering algorithms which, when used in practice, applies quite naturally the constraint $C^j = A^j(S^j)$, is that the produced solution cannot be altered by a second application of the collaboration. We call that property the *purity of the collaboration*:

Definition 5 (Pure collaborative clustering algorithm). *Let $\mathcal{A}^{col}$ be a collaborative clustering algorithm that outputs a multi-view partition of index J. $\mathcal{A}^{col}$ is said to be pure, if and only if*

$$\mathcal{A}^{col}\left(S^{1:J}, A^{1:J}, \mathcal{A}^{col}\big(S^{1:J}, A^{1:J}, C^{1:J}\big) \right) = \mathcal{A}^{col}\big(S^{1:J}, A^{1:J}, C^{1:J}\big). \tag{3}$$

The previous definition entails that a collaborative algorithm is pure if and only if re-applying it to its own ouput will not change the resulting partitions. As such purity is a desirable property for collaborative algorithms since it ensures the definitive aspect of the results for algorithms that have this property.

3.2.3 Unsupervised Ensemble Learning. The main difference between the aforementioned collaborative and multi-view clustering algorithms, and the unsupervised ensemble learning ones, is that the latter operate at the level of partitions only (Figure 4) and are therefore slightly difference in essence.

Definition 6 (Unsupervised ensemble learning). *Consider a total space* $\mathbb{X} = \mathbb{X}^1 \times \ldots \times \mathbb{X}^J$ *and a subspace* $X \subseteq \mathbb{X}$*. For all* j*, let* C^j *be a partition of* X^j*. An unsupervised ensemble learning algorithm is defined as a mapping* $\mathcal{A}^{ens} : C^1 \times \ldots \times C^J \to C^j$*, where the* C^j *is a partition of a given view* X^j*.*

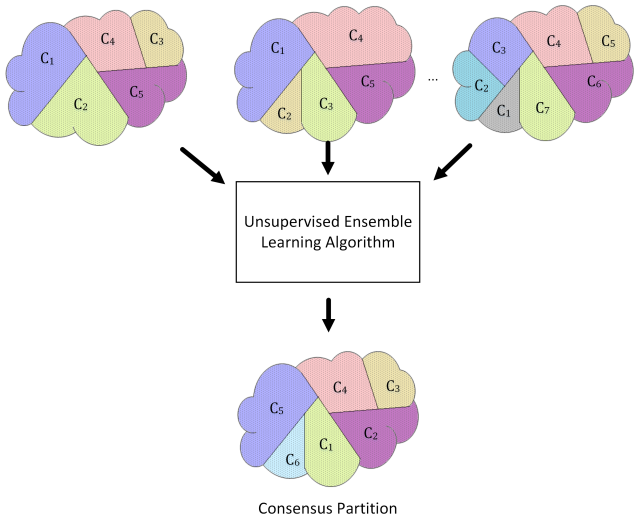

Fig. 4: Graphical definition of unsupervised ensemble learning: Notice that the data themselves are never involved in the process.

Although this definition is chosen to be as general as possible, most applications focus on the simplest case where all views are equal ($\mathbb{X}^1 = \ldots = \mathbb{X}^J$ and $X^1 = \ldots = X^J$) with the implicit assumption that the data are the same in all views (which, formally, corresponds to an assumption on the probability distribution P, such as in Example 3).

This very different nature makes it impossible to relate unsupervised ensemble learning algorithms to the other families, such as done for instance for multi-view and collaborative frameworks (Proposition 2). However, we can observe the following relations between the three families of algorithms:

Proposition 3. *Let* $\mathcal{X} = \mathbb{X}^1 \times \ldots \times \mathbb{X}^J$ *be a total space and* $A^1, \ldots, A^J$ *local clustering algorithms on the views* $\mathbb{X}^j$*. Consider a multi-view algorithm* $\mathcal{A}^{MV}$ *on* $\mathbb{X}$*. Let* $\mathcal{A}^{ens,1}, \ldots, \mathcal{A}^{ens,J}$ *be* J *unsupervised ensemble learning algorithms,*

where $\mathcal{A}^{ens,j}$ produces a partition in $\mathbb{X}^j$. We define

$$\mathcal{A}^{ens}(C^1,\dots,C^J) = (\mathcal{A}^{ens,1}(C^1,\dots,C^J),\dots,\mathcal{A}^{ens,J}(C^1,\dots,C^J))$$

Then the following statements are correct:

1. *The function $S^1,\dots,S^J,A^1,\dots,A^J,C^1,\dots,C^J \mapsto \mathcal{A}^{ens}(C^1,\dots,C^J)$ is a collaborative clustering algorithm.*
2. *If $\mathcal{A}^{MV}$ produces a multi-view partition of index J, then the function $\mathcal{A}^{ens} \circ \mathcal{A}^{MV}$ is a multi-view clustering algorithm of index J.*
3. *$\mathcal{A}^{MV}$ can be decomposed as the combination $\mathcal{A}^{MV} = \mathcal{A}^{ens,\prime} \circ \mathcal{A}^{MV,loc}$ of local regular clustering algorithms: $\mathcal{A}^{MV,loc} : (S^1,\dots,S^J) \mapsto (A^1(S^1),\dots,A^J(S^J))$ and of unsupervised ensemble learning algorithms $\mathcal{A}^{ens,\prime}$ (defined in a similar manner as $\mathcal{A}^{ens}$).*

These statements are direct consequences of the definitions. Point (3) formalizes the decomposition of multi-view algorithms into two steps: applying local algorithms to each view to produce a partition, and applying an unsupervised ensemble learning to exchange the information between the views. The combination of this point and of Proposition 2 formalizes the idea of Grozavu and Bennani [23] of a two-step process (applying local algorithms, then refine). Regarding the notations, the multi-view algorithm $\mathcal{A}^{MV,loc}$ will be called, in the following section, *concatenation* of local algorithms and will be denoted by $\bigoplus_j A^j$. It will play a central role in the study of theoretical properties of a collaboration.

3.3 Summary: Four Interleaving Notions

In this section, we have introduced four interleaving notions: regular clustering, multi-view clustering, collaborative clustering and unsupervised ensemble learning. The definitions we proposed are extremely general, and in particular do not incorporate some classical (sometimes implicit) properties associated to these notion, for instance the independence to the order of arguments (a permutation of S^i and S^j in the arguments of $\mathcal{A}^{MV}$ simply yields a permutation of C^i and C^j in the output). Actually, these properties are not essential to the very nature of these notions and it would be reasonable to think of applications where these do not hold.

A fundamental result we showed is that multi-view clustering and regular clustering are in essence similar, in the sense that multi-view clustering (be it with or without fusion) generates a clustering of the total space and, conversely any regular clustering satisfying some constraints can be understood as a multi-view clustering (Proposition 1). These constraints can be understood as some regularization of the produced clustering. This result however is not meant to lower the importance of multi-view clustering as an independent domain: on contrary, implementing these constraints is a challenge *in se* and is the core motivation of a whole field. It is however of primary importance for a theoretical

study of multi-view clustering (and by extension, of collaborative clustering), since it entails that the same tools and results apply to it.

Such as introduced in Definition 4, multi-view and collaborative clustering algorithms differ mostly on the nature of their input arguments. Collaborative clustering algorithms are more general, since they consider the local algorithms and initial partitions in addition to the data points. As a direct result, it is clear that multi-view methods are specific cases of collaborative clustering, but, conversely, the collaborative clustering algorithms inspired by the works of Grozavu and Bennani [23] can be reduced to multi-view algorithms. Indeed, such algorithms use the local algorithms only to constrain the local views.

Unsupervised ensemble learning algorithms are not clustering algorithms in the sense of Section 3.1.1, since they do not take data points as input. However, they are strongly involved in collaborative and multi-view clustering: It has been discussed that a multi-view algorithm can be decomposed into a regular clustering algorithm and an unsupervised ensemble learning. This decomposition, which may be only theoretical and does not necessarily reflect how the algorithms work *in practice*, amounts to considering the output of a multi-view algorithm as a correction of local partitions based on the information provided by the other views. In the next section, a measure of the influence of this ensemble algorithm will be used to define the *novelty* and *consistency* of a collaborative/multi-view algorithm.

4 Properties of Collaborative Clustering: Stability, Novelty and Consistency

In this section, we introduce various properties which could be expected from collaborative algorithms. These notions and formal definitions are inspired by the notion of clustering *stability* introduced in the original work of Ben David et al. [2]. We will see how stability can be extended to collaborative clustering and how the stability of a collaborative clustering algorithm depends inherently from a novel notion, called *consistency*.

All these notions focus more precisely on the influence of the local clustering algorithms $A^1, \ldots, A^J$ and of the input data $S^1, \ldots, S^J$, onto the produced partitions. In particular, we will consider the initial partitions $C^1, \ldots, C^J$ as fixed, for instance as $C^j = A^j(S^j)$. We will use the notation $\mathcal{A}_{\langle A^1,\ldots,A^J\rangle}$ (shorten in $\mathcal{A}$ when the local context is explicit) to designate such a collaborative clustering algorithm based on local algorithms $A^1, \ldots, A^J$. Notice that the defined function $\mathcal{A}_{\langle A^1,\ldots,A^J\rangle}$ corresponds to a multi-view algorithm, when the local algorithms are fixed.

4.1 Reminders on Clustering Stability

Stability is a key notion in regular clustering that assesses the ability of a clustering algorithm to find a consistent partitionning of the data space on different

subsamples [2,29,6]. From its definition it is a neutral quality index that evaluates the noise robustness of clustering algorithms.

The stability of a clustering algorithm is defined as the ability to produce always the same partition, given enough data from a fixed distribution. In order to formalize this idea, it is important to be able to measure how to partitions differ. This is done with the notion of *clustering distance*:

Definition 7. *(**Clustering distance**) Let $\mathcal{P}$ be a family of probability distributions over some domain $\mathbb{X}$. Let Σ be a family of clusterings of $\mathbb{X}$. A clustering distance is a function $d : \mathcal{P} \times \Sigma \times \Sigma \to [0, 1]$ that for any $P \in \mathcal{P}$ and any clusterings C, C', C'' satisfies:*

1. $d_P(C, C) = 0$
2. $d_P(C, C') = d_P(C', C)$ *(symmetry)*
3. $d_P(C, C'') \leq d_P(C, C') + d_P(C', C'')$ *(triangle inequality)*

Please note that clustering distances as we have defined them are not required to satisfy $d_P(C, C') = 0 \Rightarrow C = C'$, which is not true with most clustering distances that are commonly used.

Example 5. A typical example of a clustering distance (introduced for instance by Ben-David et al. [2]) is the Hamming distance:

$$d_P^H(C, C') = \mathop{\mathbb{P}}_{\substack{x \sim P \\ y \sim P}} \Big[\big(C(x) = C(y)\big) \oplus \big(C'(x) = C'(y)\big) \Big] \tag{4}$$

where $\oplus$ denotes the logical XOR operation. The Hamming distance measures how much the two partitions group together the same pairs of points. It can be easily checked that d_P^H satisfies the properties of a clustering distance. It is also clear that two partitions C and C' can be different and yet have a 0 distance. For instance, if the space is continuous and C and C' differ only on one point, we still have $d_P^H(C, C')$.

As we have mentioned, clustering stability measures how a perturbation in the data affects the result of a clustering algorithm. Using the proposed definition of a clustering algorithm, the stability of algorithm A can then be formalized as the distance between the produced partitions $A(X_1)$ and $A(X_2)$ for X_1 and X_2 sampled from the same distribution P:

Definition 8. *(**Stability of a clustering algorithm**) Let P be a probability distribution over $\mathcal{X}$. Let d be a clustering distance. Let A be a clustering algorithm (a regular one). The stability of the algorithm A for the sample of size m with respect to the probability distribution P is:*

$$stab(A, P, m) = \mathop{\mathbb{E}}_{\substack{X_1 \sim P^m \\ X_2 \sim P^m}} [d_P(A(X_1), A(X_2))] \tag{5}$$

From there, the stability of algorithm A with respect to the probability distribution P is:

$$stab(A, P) = \limsup_{m \to \infty} stab(A, P, m) \tag{6}$$

We say that a regular clustering algorithm A is stable for P, if $stab(A, P) = 0$.

A very strong property of clustering stability, demonstrated by Ben David et al. [2], states that a risk minimizing clustering algorithm (see Definition 1) satisfying a specific property of unicity of the optimal produced partition (such as defined below), is stable.

Definition 9. ***(Unique minimizer)*** *We fix a risk minimization scheme* $(\mathbb{X}, \Sigma, \mathcal{P}, \mathcal{R})$. *Let d be a clustering distance. We say that a probability distribution P has unique minimizer* C^* *if:*

$$(\forall \eta > 0)(\exists \epsilon > 0)(R(P, C) < opt(P) + \epsilon) \Longrightarrow d_P(C^*, C) < \eta).$$

More generally, we say a probability distribution P has n distinct minimizers, if there exists $C_1^*, C_2^*, \ldots, C_n^*$ *such that* $d_P(C_i^*, C_j^*) > 0$ *for all* $i \neq j$, *and*

$$(\forall \eta > 0)(\exists \epsilon > 0)(R(P, C) < opt(P) + \epsilon \Longrightarrow (\exists 1 \leq i \leq n) \quad d_P(C_i^*, C) < \eta).$$

Note that there is a technical subtlety here: the definition does not require that there is only a single clustering with the minimal cost, but rather that for any two optima $C_1^*, C_2^*, d_P(C_1^*, C_2^*) = 0$, which does *not* imply that $C_1^* = C_2^*$. Technically, we can overcome this difference by forming equivalence classes of clusterings, saying that two clusterings are equivalent if their clustering distance is zero. Similarly, n distinct optima correspond n such equivalence classes of optimal clusterings.

4.2 From Regular to Collaborative Clustering: Stability, Novelty and Consistency

As discussed previously, multi-view clustering, and in particular collaborative clustering, can be interpreted as a specific constrained form of clustering. Following this idea, we show now how the general theoretical notions presented for regular clustering can be formulated for collaborative clustering.

We remind that regular clustering and multi-view partitions are theoretically equivalent because $\mathbb{N}^J$ and $\mathbb{N}$ are equipotent. In the following, we will denote by $\nu : \mathbb{N}^J \to \mathbb{N}$ a bijective application mapping $\mathbb{N}^J$ to $\mathbb{N}$. With this application, the mapping $\nu \circ \mathcal{C}$ is a clustering of $X \subseteq \mathbb{X}$.

4.2.1 Multi-view Clustering Distance and Stability.

Any analysis of the theoretical properties of collaborative clustering requires us to firstly define the relevant clustering distance used to measure the discrepancy between the produced clusters on the total space $\mathbb{X} = \mathbb{X}^1 \times \ldots \mathbb{X}^J$. Even though in theory any distance satisfying the conditions of Definition 7 would be applicable, it

seems also interesting to consider distances more adapted to the specificity of the space decomposition. In the following proposition, we show that a simple linear combination of local distances is a valid distance for the total space.

Proposition 4 (Canonical multi-view clustering distance). *Let $\mathbb{X} = \mathbb{X}^1 \times \cdots \times \mathbb{X}^J$ be a domain, and the d^j clustering distance on $\mathbb{X}^j$. We define the function $d : \mathcal{P} \times \mathcal{S} \times \mathcal{S} \to [0, 1]$ such that $d_P(\mathcal{C}_1, \mathcal{C}_2) = \frac{1}{J} \sum_{j=1}^{J} d^j_{P_j}(\mathcal{C}_1^j, \mathcal{C}_2^j)$. Then d defines a clustering distance on $\mathcal{X}$. We call it the* ***canonical multi-view clustering distance****.*

Proof. The clustering distance properties follow directly from the linearity in terms of d^j and from the properties of the local clustering distances.

In this definition, we chose the coefficients of the linear combination to be uniformly equal to $1/J$, with the will to give the same importance to all views, but we would like to emphasize that none of the following results would be altered by choosing non-uniform weights.

Given a distance on the total space $\mathbb{X} = \mathbb{X}^1 \times \ldots \times \mathbb{X}^J$, the definition of clustering stability above (Definition 8) can be applied directly to the case of collaborative and multi-view clustering.

When the used distance is the canonical multi-view clustering distance, the stability of a collaborative or multi-view algorithm on the total space has a simple interpretation. Let $\mathcal{A}$ be the total algorithm and $\mathcal{A}^j : S^1, \ldots, S^J \mapsto \big(\mathcal{A}(S^1, \ldots, S^J)\big)^j$. Algorithm $\mathcal{A}^j$ considers the projection of the multi-view partition produced by $\mathcal{A}$ onto the subspace $\mathbb{X}^j$. Note that, in the case of collaborative clustering, this algorithm $\mathcal{A}^j$ is distinct from the local algorithm A^j. The following characterization of multi-view stability comes directly from the definitions:

Proposition 5. *Multi-view algorithm $\mathcal{A}$ is stable for the canonical multi-view clustering distance if and only if, for all j, the projections $\mathcal{A}^j$ are stable.*

Such a characterization of multi-view stability, despite being intuitive, is actually a consequence of the choice of the canonical distance. For a general clustering multi-view clustering distance, there is no guarantee that this result remains correct. Satisfying Proposition 5 can be an important property that a reasonable multi-view clustering distance should satisfy.

Example 6. Given $\mathbb{X} = \mathbb{X}^1 \times \ldots \mathbb{X}^J$ and local clustering distances d^j, the function $d_P(C_1, C_2) = d_{P^1}(C_1^1, C_2^1)$ defines a proper multi-view clustering distance. With this distance, the equivalence in Proposition 5 is not always satisfied. For instance, any algorithm which would be stable on view 1 but unstable on at least one other view would still be globally stable with respect to the chosen multi-view distance.

4.2.2 Novelty and Consistency.

Using Definition 4 and under the assumptions of Proposition 2, we have seen that collaborative clustering algorithms

(with fixed local algorithms) are a specific case of multi-view clustering algorithms that aim to refine locally found partitions throughout the collaboration process.

It is worth mentioning that in general, the expected goal of collaborative clustering is that the projection of the clustering obtained by the collaborative algorithm onto one of the views j should be distinct from the original clustering results obtained by the local algorithm $\mathcal{A}^j$ for the same view. This corresponds to the effect of the unsupervised ensemble learning step as discussed in Section 3.3 and Proposition 3.3. In other words, if $\mathcal{C} = \mathcal{A}(X)$, then in general we have that $\mathcal{C}^j \neq A^j(X^j)$. It is this property that makes the collaboration interesting and more valuable than a simple concatenation of the local results.

Definition 10. *(**Concatenation of local clustering algorithms**) The concatenation of local clustering algorithms A^1 to A^J, denoted by $\bigoplus_{j=1}^{J} A^j$ is defined as follows: If $\mathcal{C}$ is the global clustering induced by $\mathcal{A} = \bigoplus_{j=1}^{J} A^j$ on a dataset X, then:*

$$\forall x \in \mathbb{X}, \forall j \in \{1, \ldots, J\}, \qquad \mathcal{C}^j(x^j) = \left(A^j(X^j)\right)(x^j) \tag{7}$$

This defines the concatenation of local clustering algorithms as a collaborative algorithm that "does nothing" (i.e. in which there is no exchange of information between the various views), and produces the exact same results as the ones obtain by the local algorithms A^j. Such a degenerate algorithm had been already used in Proposition 3 under the name of $\mathcal{A}^{MV,loc}$.

We now introduce the notion of *novelty*, the property of any collaborative clustering algorithm to do more than just concatenating the local solutions. This represents the ability of a collaborative algorithm to produce solutions that could not have been found locally.

Definition 11. *(**Collaborative clustering novelty**) Let P be probability distribution over $\mathcal{X}$. The novelty of the algorithm $\mathcal{A}$ for the sample size m with respect to the probability distribution P is*

$$nov(\mathcal{A}_{\langle A^1,\ldots,A^J\rangle}, P, m) = \mathop{\mathbb{P}}_{X\sim P^m}\left[\mathcal{A}_{\langle A^1,\ldots,A^J\rangle}(X) \neq \bigoplus_{j=1}^{J} A^j(X^j)\right] \tag{8}$$

where $\mathcal{A}(X)$ is the collaborative or multi-view clustering and $\bigoplus_{j=1}^{J} A^j(X^j)$ is the concatenation of all local clusterings.

Then, the novelty of algorithm $\mathcal{A}$ with respect to the probability distribution P is

$$nov(\mathcal{A}, P) = \limsup_{m\to\infty}\ nov(\mathcal{A}, P, m) \tag{9}$$

$\mathcal{A}$ satisfies the novelty property for distribution P if $nov(\mathcal{A}, P) > 0$

Yet, while novelty is often described as a desirable property, in collaborative clustering (and in unsupervised ensemble learning as we will see later), there is

also a need that the results found a the global level after the collaborative step remain *consistent* with the local data when projected onto the local views. This leads us to the notion of consistency:

Definition 12. ***(Collaborative clustering consistency)*** *Let P be probability distribution over $\mathcal{X}$. Let d be a clustering distance. Let $\mathcal{A}$ be a collaborative clustering algorithm. The consistency of the algorithm $\mathcal{A}$ for the sample size m with respect to the probability distribution P is*

$$cons(\mathcal{A}_{\langle A^1,\dots,A^J\rangle}, P, m) = \mathop{\mathbb{E}}_{X\sim P^m}\left[d_P\left(\mathcal{A}_{\langle A^1,\dots,A^J\rangle}(X), \bigoplus_{j=1}^{J} A^j(X^j)\right)\right] \quad (10)$$

The consistency of algorithm $\mathcal{A}$ with respect to the probability distribution P is

$$cons(\mathcal{A}, P) = \limsup_{m\to\infty} cons(\mathcal{A}, P, m) \quad (11)$$

Please note that this definition of consistency for collaborative clustering has no link with the consistency of regular clustering algorithms as it was defined by Kleinberg [27]. Intuitively, our consistency measures the distance of the global clustering produced by the collaboration to the clustering produced by concatenation of local algorithms.

Two remarks can be made about novelty and consistency: The first one is that obviously these notions are very specific to the case of collaborative clustering and unsupervised ensemble learning (as we will see after), as it is obvious that without intermediary local clustering partitions, these notions simply do not exist. The second remark is that there exists a noticeable link between consistency and novelty, since novelty is actually a particular case of consistency based on the clustering distance defined as follows:

$$\forall \mathcal{C}, \mathcal{C}', \quad d_P^{\mathbb{I}}(\mathcal{C}, \mathcal{C}') = \mathbb{I}(\mathcal{C} \neq \mathcal{C}') \quad (12)$$

It can be verified easily that the function $d_P^{\mathbb{I}}$ is clustering distance.

However, although novelty is a specific case of consistency for a given distance, consistency and novelty are not equivalent in general, for an arbitrary clustering distance. This means that consistent algorithms are not necessarily concatenations. This is mainly due to the fact that clustering distances do not satisfy $d_P(\mathcal{C}, \mathcal{C}') = 0 \Rightarrow \mathcal{C} = \mathcal{C}'$. The converse is true however: $nov(\mathcal{A}, P) = 0$ implies $cons(\mathcal{A}, P) = 0$, whatever clustering distance is used to compute the consistency.

Example 7. Using Hamming distance as a local clustering distance and the canonical multi-view distance as d_P, we have seen previously that local partitions which would differ on a set of P-measure zero (in particular a finite set) would have a zero distance while being distinct. Consider then a trivial collaborative algorithm which changes the cluster of one single point in all the local partitions. Such an algorithm would be consistent, and yet the produced partitions would be distinct.

Finally, it is worth mentioning that while producing novel solutions is generally considered a desirable property for any multi-view or collaborative methods, this might not always be the case: We can for instance imagine a scenario where local solutions are already optimal but different and where novelty might mean sub-optimal solutions everywhere. Another more standard scenario would be a local view (or several local views) having too strong an influence and forcing other views to change otherwise fine but too different local solutions.

4.3 Stability of collaborative clustering

Now that we have defined the notion of stability for regular clustering, as well as key notions from collaborative and multi-view clustering, we have two goals : The first one is to derive a notion of stability for collaborative clustering algorithms. And second, we want to know how this notion of collaborative clustering stability can be linked to the notions of novelty (Definition 11) and consistency (Definition 12), and more importantly to the stability of the local algorithms since collaborative clustering has the particularity of using sets of regular clustering algorithms whose stability is already clearly defined (See Definition 8).

4.3.1 Stability of Risk-Minimizing Collaborative Clustering Algorithms. Theorem 1 below shows a direct adaptation of Ben-David's key theorem on clustering stability (Theorem 10 in [2]) to collaborative clustering.

Theorem 1. *If P has a unique minimizer $\mathcal{C}^*$ for risk $\mathcal{R}$, then any $\mathcal{R}$-minimizing collaborative clustering algorithm which is risk converging is stable on P.*

Proof. Let $\mathcal{A}$ be a collaborative clustering algorithm on $\mathbb{X} = \mathbb{X}^1 \times \cdots \times \mathbb{X}^J$. Let us also consider a bijection $\nu : \mathbb{N}^J \to \mathbb{N}$. Then, based on collaborative algorithm $\mathcal{A}$, one can build a clustering algorithm $\tilde{\mathcal{A}}$ such that the clustering $\tilde{\mathcal{C}}$ induced by a sample S for $\tilde{\mathcal{A}}$ is such that $\tilde{\mathcal{C}} = \nu \circ (\mathcal{A}(S))$. For simplicity purposes, we will denote this algorithm $\tilde{\mathcal{A}} = \nu \circ \mathcal{A}$. We call d_P the global clustering distance and $\tilde{d}_P$ its associated local distance such that $\tilde{d}_P(\tilde{\mathcal{C}}_1, \tilde{\mathcal{C}}_2) = d_p(\nu^{-1} \circ \tilde{\mathcal{C}}_1, \nu^{-1} \circ \tilde{\mathcal{C}}_2)$.

Using these two clustering distances in the definition of stability, the following lemma is straightforward:

Lemma 1. *If $\tilde{\mathcal{A}} = \nu \circ \mathcal{A}$ is stable (for a distance $\tilde{d}_P$), then $\mathcal{A}$ is stable (for the distance d_P).*

If $\mathcal{A}$ is $\mathcal{R}$-minimizing, then $\tilde{\mathcal{A}}$ is $\tilde{\mathcal{R}}$-minimizing with $\tilde{\mathcal{R}}(P, \tilde{\mathcal{C}}) = \mathcal{R}(P, \nu^1 \circ \tilde{\mathcal{C}})$. It is direct that $Opt_{\mathcal{R}}(\tilde{P}) = Opt_{\mathcal{R}}(P)$ and that $\tilde{\mathcal{A}}$ is risk-converging. It is also direct that P has a unique minimizer $\tilde{\mathcal{C}}^*$ associated to $\tilde{\mathcal{R}}$.

Combining all the previous results together, it follows that $\tilde{\mathcal{A}}$ is $\tilde{\mathcal{R}}$-minimizing and risk converging. Since P has a unique minimizer for $\tilde{\mathcal{R}}$, then using [2] we have that $\tilde{\mathcal{A}}$ is stable. Lemma 1 guarantees the result.

Theorem 1 above implies that collaborative clustering algorithms can be treated exactly the same way as standard clustering algorithms when it comes to stability analysis. As we have seen previously, since $\mathbb{N}^J$ and $\mathbb{N}$ are equipotent, a collaborative clustering can be interpreted as a clustering of $X \subseteq \mathbb{X}$, and therefore there is a direct adaptation of stability from regular clustering to collaborative clustering when using the clustering distance from Proposition 4.

The result of Theorem 1 is extremely general and does *not* depend on the choice of a specific clustering distance: it shows the stability (*relative to a fixed distance* d_P) of risk-minimizing collaborative algorithms with a unique minimizer for distance d_P. The question remains open to know whether state-of-the-art collaborative clustering algorithms are stable with respect to a reasonable clustering distance (for instance the canonical distance).

4.3.2 Stability and Consistency. Stability is a notion introduced in regular clustering to describe how an algorithm is affected by slight changes in the data. We have introduced consistency as a measure of how strongly the collaboration affects the local decisions. This notion is inherent to multi-view and collaborative techniques. We will now show that, even though these two notions are intrinsically of different natures, they are strongly connected.

A first result on collaborative clustering stability can be shown about the concatenation of clustering algorithms. Proposition 6 below states that a concatenation of local algorithms is stable provided that the local algorithms are stable.

Proposition 6. *Suppose that the local algorithms A^j are stable for distance $d^j_{P_j}$. Then the concatenation of local algorithms $\mathcal{A} = \bigoplus_{j=1}^J A^j$ is stable for the canonical distance.*

Proof. Let X_1 and X_2 be two samples drawn from distribution P. Then we have :

$$d_P(\mathcal{A}(X_1), \mathcal{A}(X_2)) = \frac{1}{J}\sum_{j=1}^{J} d^j_{P_j}\left((\mathcal{A}(X_1))^j, (\mathcal{A}(X_2))^j\right) \tag{13}$$

$$= \frac{1}{J}\sum_{j=1}^{J} d^j_{P_j}\left(A^j(X_1^j), A^j(X_2^j)\right) \tag{14}$$

Because of the linearity of the expected value, it comes that:

$$stab(\mathcal{A}, P, m) = \frac{1}{J}\sum_{j=1}^{J} stab(A^j, P^j, m) \tag{15}$$

Hence the stability of $\mathcal{A}$.

This result is rather intuitive, since the concatenation corresponds to a collaborative algorithm that does nothing. From this point of view, it is expected that the unmodified results of stable local algorithms will remain stable. This result is a consequence of the choice of the canonical distance and may be invalid for other distances. We note here that conserving the stability of concatenation is a desirable property for the choice of a clustering distance on the total space.

More interestingly, using the notion of consistency, the same result can be applied to get a more generic result (still valid only for the canonical distance):

Theorem 2. *Let $\mathcal{A}_{\langle A^1,\cdots,A^J\rangle}$ be a collaborative clustering algorithm. Then the stability of $\mathcal{A}$ relatively to the canonical distance is upper-bounded as follows:*

$$stab(\mathcal{A}, P) \leq cons(\mathcal{A}, P) + \frac{1}{J}\sum_{j=1}^{J} stab(A^j, P^j) \tag{16}$$

Proof. Let X_1 and X_2 be two samples drawn from distribution P. Since the canonical distance satisfies the triangular inequality, we have:

$$d_P(\mathcal{A}(X_1), \mathcal{A}(X_2)) \leq d_P\left(\mathcal{A}(X_1), \left(\bigoplus_{j=1}^{J} A^j\right)(X_1)\right) \tag{17}$$

$$+ d_P\left(\left(\bigoplus_{j=1}^{J} A^j\right)(X_1), \left(\bigoplus_{j=1}^{J} A^j\right)(X_2)\right) \tag{18}$$

$$+ d_P\left(\left(\bigoplus_{j=1}^{J} A^j\right)(X_2), \mathcal{A}(X_2)\right) \tag{19}$$

Then, by taking the expected value of this expression, we obtain:

$$stab(\mathcal{A}, P, m) \leq 2 \times \mathbb{E}_{X\sim P^m}\left[d_P\left(\mathcal{A}(X), \left(\bigoplus_{j=1}^{J} A^j\right)(X)\right)\right] \tag{20}$$

$$+ \mathbb{E}_{X_1,X_2\sim P^m}\left[d_P\left(\left(\bigoplus_{j=1}^{J} A^j\right)(X_1), \left(\bigoplus_{j=1}^{J} A^j\right)(X_2)\right)\right] \tag{21}$$

which is the result we wanted.

This result has the advantage of being generic since it makes no assumption on the nature of the collaboration process. It also has the direct consequence that any consistent collaborative algorithm working from stable local results is stable for the canonical distance. However, this corollary is quite limited since the consistency assumption is extremely strong and does not apply to most practical cases where the collaborative process is expected to find results that differ from the simple concatenation of the local results from each views.

4.3.3 Stability of contractive collaborative algorithms. We have seen that any multi-view algorithm can be, at least theoretically, decomposed into two steps: a local step where local algorithms compute a first partition, and a collaborative step in which the partitions are refined by exploiting the collaboration. Mathematically, we expressed this property in Proposition 3 by defining $\mathcal{A}^{ens,j}$ as an unsupervised ensemble learning algorithm producing a partition of $\mathbb{X}^j$, and setting $\mathcal{A}^{ens}(C^1,\dots,C^J) = (\mathcal{A}^{ens,1}(C^1,\dots,C^J),\dots,\mathcal{A}^{ens,J}(C^1,\dots,C^J))$. With these notations, any collaborative clustering algorithm $\mathcal{A}_{\langle A^1,\dots,A^J\rangle}$ can be expressed as $\mathcal{A}_{\langle A^1,\dots,A^J\rangle} = \mathcal{A}^{ens} \circ \left(\bigoplus_{j=1}^{J} A^j\right)$. We will show that a desirable property, for this generalized ensemble learning algorithm to guarantee the global stability, is to be Lipschitz continuous with respect to clustering distance d_P (which implies, in this context, being contractive[5]).

Theorem 3. *Let $\mathcal{A}_{\langle A^1,\dots,A^J\rangle}$ be a collaborative clustering algorithm, which decomposes into $\mathcal{A}^{ens} \circ \left(\bigoplus_{j=1}^{J} A^j\right)$. Suppose that $\mathcal{A}^{ens}$ is Lipschitz continuous for the canonical distance d_P, in the sense that there exists $K \in (0,1]$ such that for all $\mathcal{C},\mathcal{C}'$, $d_P(\mathcal{A}^{ens}(\mathcal{C}),\mathcal{A}^{ens}(\mathcal{C}')) \leq K d_P(\mathcal{C},\mathcal{C}')$. Then if all $A^1,\dots,A^J$ are stable, $\mathcal{A}_{\langle A^1,\dots,A^J\rangle}$ is also stable.*

Proof. Let us consider two samples X_1 and X_2 drawn from the distribution P. From there, we have:

$$d_P\big(\mathcal{A}(X_1) - \mathcal{A}(X_2)\big) = d_P\left(\mathcal{A}^{ens} \circ \left(\bigoplus_{j=1}^{J} A^j\right)(X_1) - \mathcal{A}^{ens} \circ \left(\bigoplus_{j=1}^{J} A^j\right)(X_2)\right) \tag{22}$$

Since $\mathcal{A}^{ens}$ is a Lipschitz contraction function, there exists a real constant $0 < K \leq 1$ such that:

$$d_P\big(\mathcal{A}(X_1) - \mathcal{A}(X_2)\big) \leq K\ d_P\left(\left(\bigoplus_{j=1}^{J} A^j\right)(X_1) - \left(\bigoplus_{j=1}^{J} A^j\right)(X_2)\right) \tag{23}$$

Since the A^j are stable for all j, from Equation (23) and Proposition 6 we directly infer the stability of $\mathcal{A}$.

This proposition is interesting but raises the question of what would be required in practice and from an algorithm point of view for a collaborative algorithm to be a Lipschitz continuous function. On the other hand, when looking at Equation (23) and considering what it means for a collaborative algorithm to be a contraction mapping ($K \leq 1$), we see that the partitioning of the two samples X_1 and X_2 drawn from the distribution P should be closer after the collaborative process. It turns out that this is exactly what is expected from a

[5] The fact that the Lipschitz constant K must be lower than 1 is due to the convention that the clustering distances are defined between 0 and 1.

collaborative algorithm. Therefore, being a contraction mapping seems to be a necessary property that any collaborative algorithm should have. However, the mathematical analysis ends here, and it would be up to algorithms developers to demonstrate that their collaborative methods indeed have this property for all possible scenarii.

From there, once again we have a proposition that looks promising, as it may validate that all 'well-designed' collaborative algorithms are stable given that the local algorithms are stable too. However, we can't be sure that it applies to any existing collaborative algorithm, as such a demonstration has never been done for any existing method. Furthermore, most of the existing implementations of collaborative clustering algorithms rely on local algorithms that are known to be unstable: K-Means, FC-Means, EM for the GMM, SOM and GTM. Thus, these methods are already excluded from the scope of this proposition.

5 Open Questions

In the history of learning theory, clustering has always remained marginal compared to supervised learning, and in particular to classification. Within the broad domain of clustering, the question of a theoretical analysis of multi-view methods is even less represented. With the formal treatment we proposed in this chapter, we aimed to give good foundations for future theoretical works on multi-view and collaborative clustering, by clarifying the involved concepts and providing first fundamental results. However, it will not escape the reader's attention that there is still a long way to having solid theoretical results. In particular, multiple questions remain open and should be investigated in future works:

Choice of a multi-view distance. The multi-view distance is the core notion conditioning the definition of stability. Because of the unsupervised nature of clustering, there is no objective way to qualify the quality of a produced partition, and in particular stability can be defined only with regards to a chosen distance. Therefore, choosing which distance to use is essential for stability to reflect interesting properties of the algorithms.

We have introduced in this chapter the *canonical multi-view clustering distance*, a simple linear combination of local clustering distances. This choice is obviously the most straightforward way to define a multi-view clustering distance that exhibits some intuitive properties. Actually, we have shown that it leads to fundamental results, in particular Theorems 2 and 3. But this multi-view clustering distance may not be entirely satisfactory, since it ignores, in its definition, a core issue of multi-view problems: the interdependence between views. By taking marginal distributions in the local space, the canonical multi-view distance essentially ignores that views could be correlated. For instance, if two views are identical, an independent stability in each of these two views is not sufficient.

It is clear from this remark that other multi-view clustering distances should be investigated and that their theoretical properties should be analyzed. It is less clear though how to build such distances. We first notice that the currently used

definition is just an extension from the definition of clustering distance for regular clustering. This definition does not constrain the distances d_P on the probability distribution P, which would be reasonable to have for multi-view distances. In addition, we observed that some intuitive results are established only for this specific clustering distance, for instance Proposition 5 (global stability if and only if stability on all views) and Proposition 6 (concatenation of stable algorithms is stable). We think that these properties should be added to the characterization of reasonable multi-view clustering distances.

Unicity of the minimizer for risk-minimizing multi-view algorithms. The stability theorem demonstrated by Ben-David et al. [2] is a fundamental theoretical result regarding clustering. By noticing that multi-view clustering can be seen as similar to regular clustering, we proposed with Theorem 1 a variant of this theorem for the multi-view and collaborative cases. This theorem is the most general we presented in this chapter, since it is not restricted by a specific choice of a clustering distance, however it is, at the same time, the least informative: indeed, we did not find evidences that standard collaborative clustering techniques satisfy the conditions of the theorem.

Multiple works in collaborative clustering have used an objective function of the form:

$$R(P, \mathcal{C}) = \sum_{j=1}^{J} \left(R^j(P^j, \mathcal{C}^j) + \sum_{i \neq j} \Delta(P, \mathcal{C}^i, \mathcal{C}^j) \right) \tag{24}$$

which corresponds to a trade-off between staying close to a local optimum and minimizing the differences between the views. It is not direct whether such a risk has a unique minimizer. Even when the local risk functions R^j have all a unique minimizer under the marginal probabilities, the divergence term $\Delta(P, \mathcal{C}^i, \mathcal{C}^j)$ brings in some perturbations and could affect the unicity of a minimizer. Intuitively here, it appears that when the Δ term is negligible, the existence of unique minimizers locally should guarantee the existence of a unique minimizer for the global risk. This result is very much in line with Theorem 2, since the multi-view stability is here relative to the local stability, but also to minimal perturbations introduced by the collaboration (i.e. consistency).

As a complement to the stability theorem, Ben-David et al. also proved that, in case there is no unique minimizers and some symmetry in the risk, the clustering algorithm is necessarily unstable. This result has not been presented in this chapter but could be of particular interest for the case of collaborative clustering. Indeed, in this context, it is not rare that two corrections of a partition are completely equivalent, which would lead to unstability. Characterizing this effect, in particular for risks of the form presented above, seems like a promising direction.

Stabilization of a collaboration. The results we presented in this chapter revolve around one main question: does the collaboration maintain the stability of the results? We could see that this is not clear, and that other factors can

enter into account. Consistency has been introduced as a measure of the novel information contained into the collaboration, compared to a simple concatenation of the local partitions. If the consistency is maxed, nothing guarantees that some perturbing information has not been exchanged as well, which may cause unstability.

However, the converse question is still open: can a collaboration of unstable algorithms be stable? We have seen in Example 6 that a trivial choice of a multi-view clustering distance can lead to a stable collaboration if at least one of the local algorithms is stable, no matter if the other are or are not stable. Another trivial example would regard constant algorithms, which by definition will be stable for any input local algorithm. These two examples are trivial, either because of a degenerate choice of a multi-view clustering distance , or of a collaborative algorithm. They show however that the question has no simple answer and requires further investigation.

The question of the stabilization of a collaboration can be seen from two opposite angles: (1) If some local algorithms are unstable, is it possible to stabilize them with a reasonable collaboration? and (2) In a stable collaboration of (a potentially high number of) stable local algorithms, can changing only one local algorithm affect the stability of the collaboration? These two questions have strong practical implications.

Decomposition of the algorithms. At multiple points in the chapter, we have seen that a collaborative clustering algorithm can be decomposed into two steps: a concatenation of the local algorithms, followed by some unsupervised ensemble learning to make the collaboration. Although some algorithms directly implement this decomposition, many others do not and for them it becomes difficult to use results based on it, including Theorem 3. It may be then important to know whether some properties of the second phase (such as Lipschitz-continuity) can be inferred when the corresponding ensemble learning algorithm is not given explicitly.

Stability of consensus In our analysis so far, we have considered only the most consensual definition of collaborative clustering where we have several algorithms working on multiple sites to first produce a local solution and then collaborate to improve each local solution without searching for a consensus partition of space. This corresponds to producing a multi-view partition of index J. However, as we have discussed in the state of the art section, clustering frameworks under the name collaborative clustering are a broad spectrum ranging from fully multi-view clustering to unsupervised ensemble learning (See Table 1). An adaptation of the theoretical notions presented in Section 4 to algorithms with consensus may present interesting peculiarities.

We note that notions such as stability, consistency or novelty do not make sense for unsupervised ensemble learning, which does not take data as input. We remind that these three measures are relative to various behaviours of the clustering algorithm when data are drawn from a specified distribution.

For other multi-view methods with consensus, apart from the inherent difficulty of defining the task (on which space is the produced partition defined?), we note that there is no natural "neutral" algorithm, i.e. an algorithm having no effect (such as the concatenation in Section 4). A possibility could be to rely on the majority vote operator.

6 Conclusion

In this chapter, we made an attempt to rethink collaborative clustering in comparison with the better known fields of multi-view clustering and ensemble learning. This formalization was needed to understand the interconnections between these various fields and to initiate a proper study of the theoretical properties of collaborative clustering. For that purpose, we extended key clustering notions such as clustering *stability* to the context of collaborative clustering, and we identified the additional key notions of *novelty* and *consistency* that are important for typical collaborative clustering applications.

Convinced of the importance to firstly defining a problem correctly before being able to solve it, we formalized the different branches of collaborative clustering, which have evolved during the last decade without being properly classified. This formal look into these algorithms made it clear that multi-view and collaborative clustering methods can all be seen as matching the definition proposed by Grozavu and Bennani [23]: collaborative methods should have an intermediary step with local results computed with local algorithms and should not aim for a consensus. Next, we demonstrated that collaborative algorithms matching this definition can be treated as multi-view clustering algorithms.

The theoretical study we proposed for collaborative and multi-view methods offers a clean basis for further investigations onto the theoretical properties of multi-view methods. Some challenges and open questions have been presented in Section 5 and we wish that our work may help to better consider the properties of existing and future collaborative clustering methods. And we also hope that our attempt at a formally defining the different branches of collaborative clustering will lead to a better integration of the somehow different family of collaborative clustering algorithms inside the multi-view and ensemble learning communities.

References

1. Arivazhagan, M.G., Aggarwal, V., Singh, A.K., Choudhary, S.: Federated learning with personalization layers (2019)
2. Ben-David, S., Von Luxburg, U., Pál, D.: A sober look at clustering stability. In: International Conference on Computational Learning Theory. pp. 5–19. Springer (2006)
3. Bickel, S., Scheffer, T.: Multi-view clustering. In: Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), 1-4 November 2004, Brighton, UK. pp. 19–26. IEEE Computer Society (2004). https://doi.org/10.1109/ICDM.2004.10095, `https://doi.org/10.1109/ICDM.2004.10095`

4. Bishop, C.M., Svensén, M., Williams, C.K.I.: GTM: the generative topographic mapping. Neural Comput. **10**(1), 215–234 (1998). https://doi.org/10.1162/089976698300017953, https://doi.org/10.1162/089976698300017953
5. Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, B., Overveldt, T.V., Petrou, D., Ramage, D., Roselander, J.: Towards federated learning at scale: System design. In: Talwalkar, A., Smith, V., Zaharia, M. (eds.) Proceedings of Machine Learning and Systems 2019, MLSys 2019, Stanford, CA, USA, March 31 - April 2, 2019. mlsys.org (2019), https://proceedings.mlsys.org/book/271.pdf
6. Carlsson, G.E., Mémoli, F.: Characterization, stability and convergence of hierarchical clustering methods. J. Mach. Learn. Res. **11**, 1425–1470 (2010), http://portal.acm.org/citation.cfm?id=1859898
7. Coletta, L.F.S., Vendramin, L., Hruschka, E.R., Campello, R.J.G.B., Pedrycz, W.: Collaborative fuzzy clustering algorithms: Some refinements and design guidelines. IEEE Trans. Fuzzy Syst. **20**(3), 444–462 (2012). https://doi.org/10.1109/TFUZZ.2011.2175400, https://doi.org/10.1109/TFUZZ.2011.2175400
8. Cornuéjols, A., Wemmert, C., Gançarski, P., Bennani, Y.: Collaborative clustering: Why, when, what and how. Inf. Fusion **39**, 81–95 (2018). https://doi.org/10.1016/j.inffus.2017.04.008, https://doi.org/10.1016/j.inffus.2017.04.008
9. Diao, E., Ding, J., Tarokh, V.: Heterofl: Computation and communication efficient federated learning for heterogeneous clients. CoRR **abs/2010.01264** (2020), https://arxiv.org/abs/2010.01264
10. Dunn, J.C.: A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. Journal of Cybernetics **3**(3), 32–57 (1973). https://doi.org/10.1080/01969727308546046, https://doi.org/10.1080/01969727308546046
11. Falih, I., Grozavu, N., Kanawati, R., Bennani, Y., Matei, B.: Collaborative multiview attributed networks mining. In: 2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018. pp. 1–8. IEEE (2018). https://doi.org/10.1109/IJCNN.2018.8489183, https://doi.org/10.1109/IJCNN.2018.8489183
12. Filali, A., Jlassi, C., Arous, N.: SOM variants for topological horizontal collaboration. In: 2nd International Conference on Advanced Technologies for Signal and Image Processing, ATSIP 2016, Monastir, Tunisia, March 21-23, 2016. pp. 459–464. IEEE (2016). https://doi.org/10.1109/ATSIP.2016.7523117, https://doi.org/10.1109/ATSIP.2016.7523117
13. Filali, A., Jlassi, C., Arous, N.: A hybrid collaborative clustering using self-organizing map. In: 14th IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2017, Hammamet, Tunisia, October 30 - Nov. 3, 2017. pp. 709–716. IEEE Computer Society (2017). https://doi.org/10.1109/AICCSA.2017.111, https://doi.org/10.1109/AICCSA.2017.111
14. Forestier, G., Wemmert, C., Gançarski, P.: Multisource images analysis using collaborative clustering. EURASIP J. Adv. Signal Process. **2008** (2008). https://doi.org/10.1155/2008/374095, https://doi.org/10.1155/2008/374095
15. Forestier, G., Wemmert, C., Gançarski, P.: Semi-supervised collaborative clustering with partial background knowledge. In: Workshops Proceedings

of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy. pp. 211–217. IEEE Computer Society (2008). https://doi.org/10.1109/ICDMW.2008.116, https://doi.org/10.1109/ICDMW.2008.116

16. Forestier, G., Wemmert, C., Gançarski, P.: Towards conflict resolution in collaborative clustering. In: 5th IEEE International Conference on Intelligent Systems, IS 2010, 7-9 July 2010, University of Westminster, London, UK. pp. 361–366. IEEE (2010). https://doi.org/10.1109/IS.2010.5548343, https://doi.org/10.1109/IS.2010.5548343
17. Forestier, G., Wemmert, C., Gançarski, P., Inglada, J.: Mining multiple satellite sensor data using collaborative clustering. In: Saygin, Y., Yu, J.X., Kargupta, H., Wang, W., Ranka, S., Yu, P.S., Wu, X. (eds.) ICDM Workshops 2009, IEEE International Conference on Data Mining Workshops, Miami, Florida, USA, 6 December 2009. pp. 501–506. IEEE Computer Society (2009). https://doi.org/10.1109/ICDMW.2009.42, https://doi.org/10.1109/ICDMW.2009.42
18. Foucade, Y., Bennani, Y.: Unsupervised collaborative learning using privileged information. CoRR **abs/2103.13145** (2021), https://arxiv.org/abs/2103.13145
19. Gançarski, P., Salaou, A.: FODOMUST: une plateforme pour la fouille de données multistratégie multitemporelle. In: de Runz, C., Crémilleux, B. (eds.) 16ème Journées Francophones Extraction et Gestion des Connaissances, EGC 2016, 18-22 Janvier 2016, Reims, France. Revue des Nouvelles Technologies de l'Information, vol. E-30, pp. 481–486. Éditions RNTI (2016), http://editions-rnti.fr/?inprocid=1002204
20. Gançarski, P., Wemmert, C.: Collaborative multi-step mono-level multi-strategy classification. Multim. Tools Appl. **35**(1), 1–27 (2007). https://doi.org/10.1007/s11042-007-0115-x, https://doi.org/10.1007/s11042-007-0115-x
21. Ghassany, M., Grozavu, N., Bennani, Y.: Collaborative clustering using prototype-based techniques. Int. J. Comput. Intell. Appl. **11**(3) (2012). https://doi.org/10.1142/S1469026812500174, https://doi.org/10.1142/S1469026812500174
22. Ghassany, M., Grozavu, N., Bennani, Y.: Collaborative multi-view clustering. In: The 2013 International Joint Conference on Neural Networks, IJCNN 2013, Dallas, TX, USA, August 4-9, 2013. pp. 1–8. IEEE (2013). https://doi.org/10.1109/IJCNN.2013.6707037, https://doi.org/10.1109/IJCNN.2013.6707037
23. Grozavu, N., Bennani, Y.: Topological collaborative clustering. Aust. J. Intell. Inf. Process. Syst. **12**(3) (2010), http://cs.anu.edu.au/ojs/index.php/ajiips/article/view/1216
24. Hafdhellaoui, S., Boualleg, Y., Farah, M.: Collaborative clustering approach based on dempster-shafer theory for bag-of-visual-words codebook generation. In: Meurs, M., Rudzicz, F. (eds.) Advances in Artificial Intelligence - 32nd Canadian Conference on Artificial Intelligence, Canadian AI 2019, Kingston, ON, Canada, May 28-31, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11489, pp. 263–273. Springer (2019). https://doi.org/10.1007/978-3-030-18305-9_21, https://doi.org/10.1007/978-3-030-18305-9_21
25. Jiang, Y., Chung, F.L., Wang, S., Deng, Z., Wang, J., Qian, P.: Collaborative fuzzy clustering from multiple weighted views. IEEE Transactions on Cybernetics **45**(4), 688–701 (2015). https://doi.org/10.1109/TCYB.2014.2334595

26. Jiang, Z.L., Guo, N., Jin, Y., Lv, J., Wu, Y., Liu, Z., Fang, J., Yiu, S., Wang, X.: Efficient two-party privacy-preserving collaborative k-means clustering protocol supporting both storage and computation outsourcing. Inf. Sci. **518**, 168–180 (2020). https://doi.org/10.1016/j.ins.2019.12.051, https://doi.org/10.1016/j.ins.2019.12.051
27. Kleinberg, J.M.: An impossibility theorem for clustering. In: Becker, S., Thrun, S., Obermayer, K. (eds.) Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]. pp. 446–453. MIT Press (2002), https://proceedings.neurips.cc/paper/2002/hash/43e4e6a6f341e00671e123714de019a8-Abstract.html
28. Kohonen, T.: The self-organizing map. Neurocomputing **21**(1-3), 1–6 (1998). https://doi.org/10.1016/S0925-2312(98)00030-7, https://doi.org/10.1016/S0925-2312(98)00030-7
29. von Luxburg, U.: Clustering stability: An overview. Found. Trends Mach. Learn. **2**(3), 235–274 (2009). https://doi.org/10.1561/2200000008, https://doi.org/10.1561/2200000008
30. Mitra, S., Banka, H., Pedrycz, W.: Rough-fuzzy collaborative clustering. IEEE Trans. Syst. Man Cybern. Part B **36**(4), 795–805 (2006). https://doi.org/10.1109/TSMCB.2005.863371, https://doi.org/10.1109/TSMCB.2005.863371
31. Murena, P., Sublime, J., Matei, B., Cornuéjols, A.: An information theory based approach to multisource clustering. In: Lang, J. (ed.) Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden. pp. 2581–2587. ijcai.org (2018). https://doi.org/10.24963/ijcai.2018/358, https://doi.org/10.24963/ijcai.2018/358
32. Ngo, L.T., Dang, T.H., Pedrycz, W.: Towards interval-valued fuzzy set-based collaborative fuzzy clustering algorithms. Pattern Recognit. **81**, 404–416 (2018). https://doi.org/10.1016/j.patcog.2018.04.006, https://doi.org/10.1016/j.patcog.2018.04.006
33. Pedrycz, W.: Collaborative fuzzy clustering. Pattern Recognit. Lett. **23**(14), 1675–1686 (2002). https://doi.org/10.1016/S0167-8655(02)00130-7, https://doi.org/10.1016/S0167-8655(02)00130-7
34. Pedrycz, W.: Knowledge-based clustering - from data to information granules. Wiley (2005)
35. Pedrycz, W., Rai, P.: Collaborative clustering with the use of fuzzy c-means and its quantification. Fuzzy Sets Syst. **159**(18), 2399–2427 (2008). https://doi.org/10.1016/j.fss.2007.12.030, https://doi.org/10.1016/j.fss.2007.12.030
36. Pokhrel, S.R.: Federated learning meets blockchain at 6g edge: A drone-assisted networking for disaster response. In: Proceedings of the 2nd ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and Beyond. p. 49–54. DroneCom '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3414045.3415949, https://doi.org/10.1145/3414045.3415949
37. Shen, Y., Pedrycz, W.: Collaborative fuzzy clustering algorithm: Some refinements. Int. J. Approx. Reason. **86**, 41–61 (2017). https://doi.org/10.1016/j.ijar.2017.04.004, https://doi.org/10.1016/j.ijar.2017.04.004

38. Strehl, A., Ghosh, J., Cardie, C.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research **3**, 583–617 (2002)
39. Sublemontier, J.: 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, Dallas, TX, USA, August 4-9, 2013. pp. 1–8. IEEE (2013). https://doi.org/10.1109/IJCNN.2013.6706911, https://doi.org/10.1109/IJCNN.2013.6706911
40. Sublime, J., Grozavu, N., Cabanes, G., Bennani, Y., Cornuéjols, A.: From horizontal to vertical collaborative clustering using generative topographic maps. Int. J. Hybrid Intell. Syst. **12**(4), 245–256 (2015). https://doi.org/10.3233/HIS-160219, https://doi.org/10.3233/HIS-160219
41. Sublime, J., Lefebvre, S.: Collaborative clustering through constrained networks using bandit optimization. In: 2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018. pp. 1–8. IEEE (2018). https://doi.org/10.1109/IJCNN.2018.8489479, https://doi.org/10.1109/IJCNN.2018.8489479
42. Sublime, J., Matei, B., Cabanes, G., Grozavu, N., Bennani, Y., Cornuéjols, A.: Entropy based probabilistic collaborative clustering. Pattern Recognit. **72**, 144–157 (2017). https://doi.org/10.1016/j.patcog.2017.07.014, https://doi.org/10.1016/j.patcog.2017.07.014
43. Sublime, J., Matei, B., Murena, P.: Analysis of the influence of diversity in collaborative and multi-view clustering. In: 2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017. pp. 4126–4133. IEEE (2017). https://doi.org/10.1109/IJCNN.2017.7966377, https://doi.org/10.1109/IJCNN.2017.7966377
44. Sublime, J., Troya-Galvis, A., Puissant, A.: Multi-scale analysis of very high resolution satellite images using unsupervised techniques. Remote. Sens. **9**(5), 495 (2017). https://doi.org/10.3390/rs9050495, https://doi.org/10.3390/rs9050495
45. Vanhaesebrouck, P., Bellet, A., Tommasi, M.: Decentralized collaborative learning of personalized models over networks. In: Singh, A., Zhu, X.J. (eds.) Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA. Proceedings of Machine Learning Research, vol. 54, pp. 509–517. PMLR (2017), http://proceedings.mlr.press/v54/vanhaesebrouck17a.html
46. Wemmert, C., Gançarski, P., Korczak, J.J.: A collaborative approach to combine multiple learning methods. Int. J. Artif. Intell. Tools **9**(1), 59–78 (2000). https://doi.org/10.1142/S0218213000000069, https://doi.org/10.1142/S0218213000000069
47. Yu, F., Tang, J., Cai, R.: Partially horizontal collaborative fuzzy c-means. International Journal of Fuzzy Systems **9**, 198–204 (2007)
48. Zimek, A., Vreeken, J.: The blind men and the elephant: on meeting the problem of multiple truths in data from clustering and pattern mining perspectives. Machine Learning **98**(1-2), 121–155 (2015). https://doi.org/10.1007/s10994-013-5334-y, http://dx.doi.org/10.1007/s10994-013-5334-y
49. Zouinina, S., Grozavu, N., Bennani, Y., Lyhyaoui, A., Rogovschi, N.: Efficient k-anonymization through constrained collaborative clustering. In: IEEE Symposium Series on Computational Intelligence, SSCI 2018, Bangalore, India, November 18-

21, 2018. pp. 405–411. IEEE (2018). https://doi.org/10.1109/SSCI.2018.8628635, https://doi.org/10.1109/SSCI.2018.8628635