Collaborative clustering based on Algorithmic Information Theory

Pierre-Alexandre Murena, Jérémie Sublime, Basarab Matei, Antoine Cornuéjols

Collaborative clustering based on Algorithmic Information Theory

Pierre-Alexandre Murena[1], Jérémie Sublime[2], Basarab Matei[3], et Antoine Cornuéjols[4]

[1]LTCI - Télécom ParisTech - Université Paris-Saclay, 75013, Paris, France
[2]LISITE laboratory - RDI Team, Institut Supérieur d'Électronique de Paris, 92130, Issy-les-Moulineaux, France
[3]Université Paris 13 - Sorbonne Paris Cité, Laboratoire d'Informatique de Paris-Nord - CNRS (UMR 7030), 93430, Villetaneuse, France
[4]UMR MIA-Paris, AgroParisTech, INRA, Université Paris-Saclay, 75005, Paris, France

June 1, 2017

Abstract

Clustering is a compression task which consists in grouping similar objects into clusters. In real-life applications, the system may have access to several views of the same data and have each view process by a specific clustering algorithm: this framework is called collaborative clustering and can benefit from algorithms capable of exchanging information between the different views. In this paper, we consider this type of unsupervised ensemble learning as a compression problem and develop a theoretical framework based on algorithmic theory of information suitable for multi-view clustering and collaborative clustering applications. Using this approach, we propose a new algorithm based on solid theoretical basis, and test it on several real and artificial data sets.

Mots-clef: Collaborative clustering, Minimum Description Length, Kolmogorov complexity.

1 Introduction

Data clustering is a Machine learning task which consists in finding the intrinsic structures of a data set by forming groups of objects that share similar features called clusters. This task is difficult in the sense that unlike in supervised learning, the evaluation of the results and the evaluation of the right number of clusters are generally unknown. Over the past two decades, this task has become even more challenging when the available data sets became more complex with the introduction of multi-view data sets, distributed data, and data set having different scales of structures of interest (e.g. hierarchical clusters). However, very much like in the real world, such problems can be tackled more easily by having several algorithms working together in order to increase both the quality of the results and their reliability.

Within this context, the field *unsupervised ensemble learning* [GSIM09, VR11] encompasses several unsupervised applications in which several unsupervised algorithms are working together with the goal of improving the final result(s). Such applications include Multi-view clustering [ZV15, KI11, BS05], the clustering of distributed data [DFVW11], and collaborative clustering [Ped02, GB10, SGBC15, FWG07]. Depending on the final application or on the context, these tasks may aim at finding a single consensus solution, or simply at globally improving the different solutions found by the algorithms based on information exchanges. Regardless of the final goal (consensus or mutual improvement), these methods are naturally overlapping, and share several common properties:

- Robustness: The ensemble learning process must lead on average to partitions that are better than the local clustering results.
- Consistency : The updated results must be somehow similar to the original local results
- Novelty : Ensemble learning must make it possible to find solutions that would have been otherwise unattainable locally.
- Stability : Results that have a lower sensitivity to noise.

In this article, we are interested in proposing a generic information-based unsupervised ensemble learning framework. Most of the existing frameworks for unsupervised ensemble learning rely either on probabilistic models link to specific algorithms, or on heuristic methods that have strong limitations such as only allowing identical and prototype-based algorithms to work together in an ensemble learning process. In addition, none of these frameworks actually describes the information exchanges between the different algorithms.

To cope with this issue, we propose a generic model based on algorithmic information theory and the principle of minimum description length. Our proposed model is based on a strong theoretical basis using Kolmogorov complexity to describe and reduce the divergences between the partitions of the different algorithms. In this article, we focused on a final application where mutual improvement is the goal, rather than reaching a single consensus solution. Our work is therefore closer to applications in collaborative clustering, but remains applicable to other unsupervised ensemble learning frameworks.

The remainder of this article is organized as follows: In Section 2, we introduce the notions of Minimum Description Length principle and Kolmogorov complexity. In section 3, we describe the principle of collaborative clustering using these notion and we present our proposed method. In section 4, we present complexity computations for two families of clustering models. In section 5, we show some experiments. And finally, in Section 6 we finish this article with some conclusions and perspectives on this work.

2 Minimum Description Length Principle

In this paper, we propose to use the Minimum Description Length (MDL) principle in order to perform collaborative clustering. The MDL principle is an inductive principle introduced by Wallace and Boulton [WB68] and by Rissanen's formal work on induction [Ris78]. The principle states that the best model to select is the model which compresses observations the most: Given a data set and an enumeration of theories to explain data, the best theory is the one that minimizes the sum of the length (in bits) of the description of the theory and of the length (in bits) of the data encoded with the theory.

MDL principle is expressed mathematically with the help of Kolmogorov complexity. Originally introduced as an alternative to probabilities as a description of randomness, Kolmogorov complexity of a string $\mathbf{x}$ is an intrinsic property of the object $\mathbf{x}$ and measures how complex the object is to generate [LV08]. Using a prefix Universal Turing Machine (UTM) $\mathcal{M}$ (hence a UTM producing decodable codes), the complexity of $\mathbf{x}$ related to machine $\mathcal{M}$ is defined as:

$$K_{\mathcal{M}}(\mathbf{x}) = \min_{p \in \mathcal{P}_{\mathcal{M}}} \quad \{l(p); p() = \mathbf{x}\} \tag{1}$$

In equation 1, the term $p \in \mathcal{P}_{\mathcal{M}}$ designates a program (hence a Turing machine) in the set of admissible programs for UTM $\mathcal{M}$ and $p()$ corresponds to the output of program p with no argument specified.

Although the *invariance theorem* extends the definition of complexity to make it machine-independent, we will use the machine-dependent definition for several reasons. First, the ideal Kolmogorov complexity is not calculable because it relies on a double minimization: over all programs of all Universal Turing Machines. Considering a restriction of the research space to a unique machine admitting a simpler set of programs is an admissible way to overcome non-calculability while preserving a meaningful approach. Statistical learning relies on assumptions of the same nature: because of the non-calculability of probabilities and in order to prevent overfitting (ie. to reject distributions which do not obey the commonly admitted aim of generalization), the assumption of choosing a restricted set of hypotheses is well accepted in the machine learning field.

Secondly, this restriction has to be seen as an inductive bias for the learning. As long known by philosophers and demonstrated by the *no-free-lunch theorem* [Wol96], there is no absolute foundation to inductive reasoning and any inductive algorithm is necessarily biased toward some tasks and some solutions. The proposed machine restriction has to be understood as a plausible inductive bias.

A similar definition can be given of *conditional Kolmogorov complexity*: conditional complexity of $\mathbf{x}$ knowing $\mathbf{y}$ is defined as the length of the shortest program on a Turing machine which takes $\mathbf{y}$ as argument and outputs $\mathbf{x}$:

$$K_{\mathcal{M}}(\mathbf{x}|\mathbf{y}) = \min_{p \in \mathcal{P}_{\mathcal{M}}} \quad \{l(p); p(y) = \mathbf{x}\} \tag{2}$$

3 Minimum Description Length Principle for Collaboration

3.1 Problem and Notations

Clustering and compression are by nature two highly related tasks. Clustering is often used as a tool for lossy compression, but the structures discovered by clustering algorithms can also be used to provide a lossless description of data.

In the context of this paper, we consider a set of J views on the same data, denoted $X^1, \ldots, X^J$. A clustering algorithm $\mathcal{A}^J$ is associated with any of the views X^j. The algorithm, given with a parameter θ^j, produces a solution vector S^j. We consider only hard clustering, which means that a point is associated to a single cluster by the algorithm.

In the context of Minimum Description Length, we consider models defined as the parameter and solution of the clustering algorithm: $M^j = \langle \theta^j, S^j \rangle$. The total model used in a collaborative context is the concatenation of all local models: $M = \langle M^1, \ldots, M^J \rangle$. A complete data view X is defined as $X = \langle X^1, \ldots, X^J \rangle$.

Using these notations, the MDL principle states that the chosen parameters and solutions for the clustering minimize the objective $K(M^1, \ldots, M^J) + K(X^1, \ldots, X^J | M^1, \ldots, M^J)$ or, equivalently:

$$K(\theta^1, S^1, \ldots, \theta^J, S^J) + K(X^1, \ldots, X^J | \theta^1, S^1, \ldots, \theta^J, S^J) \tag{3}$$

The purpose of the following sections is to provide a simplified and calculable expression of the two involved complexity terms.

3.2 Data description

We first focus on the term $K(X^1, \ldots, X^J | M^1, \ldots, M^J)$ which corresponds to the way the complete model is used to describe data points. The simplification proposed for this term is straightforward but enlightens important properties of the chosen framework. In particular, we will suppose that no transfer is involved in this part.

The first hypothesis consists in isolating each data representation X^i: the views are supposed to be independently described. However, this hypothesis is different from an actual independence hypothesis: two views can share common or correlated attributes. Description independence is weaker as an assumption than statistical independence because it allows a strong correlation between variables. In particular, two different views may share common attributes or even be identical but the system will keep considering independent computations for them though.

$$K(X^1, \ldots, X^J | M^1, \ldots, M^J) \leq \sum_{i=1}^{J} K(X^i | M^1, \ldots, M^J) \tag{4}$$

This assumption makes sense in a context of collaborative clustering for which a complete view cannot be accessed by the system but is split into components that are managed independently by several collaborating sub-systems.

A second natural hypothesis consists in attributing the whole description of each data representation X^i to the corresponding model M^i, which corresponds to using property $K(X^i | M^1, \ldots, M^J) \leq K(X^i | M^i)$. This property points out that the description length of X^i using one single model is necessarily higher than using all models, because the description does not exploit information contained in other models. Our choice is justified here both by our aim at obtaining a tractable upper-bound and by the idea that collaborative systems first run local terms independently before transfering information between local agents [GB10, SGBC15]. Including this upper-bound into equation 4 leads to:

$$\begin{aligned} &K(X^1, \ldots, X^J | M^1, \ldots, M^J) \\ &\qquad \leq \sum_{i=1}^{J} K(X^i | M^1, \ldots, M^J) \leq \sum_{i=1}^{J} K(X^i | M^i) \end{aligned} \tag{5}$$

The expression obtained in equation 5 has to be considered as a local generation term for the views. No collaboration is involved in this expression: the collaborative component of MDL is held by model description. This hypothesis is particularly restrictive, but consistent with the separation of a local term (describing local fitness for each algorithm and a global term (description interactions between solutions). As mentioned earlier, such a separation is commonly accepted for collaborative clustering. In terms of data description, this choice is straightforward as well: local views can be seen as managed by independent Turing machines provided with local models.

3.3 Model description

The model complexity $K(M) = K(M^1, \ldots, M^J)$ can be expressed using the definitions of the models presented earlier ($M^i = \langle \theta^i, S^i \rangle$). The total model complexity measures the complexity of all solutions together with all parameters. Based on this, the first simplification consists in separating the parameters and

the solutions using chain rule. The chain rule states that for two objects A and B, the complexity $K(A,B)$ is upper-bounded by $K(A)+K(B|A)$ up to a constant. In our context, we obtain:

$$K(M) \leq K(S^1,\ldots,S^J) + K(\theta^1,\ldots,\theta^J|S^1,\ldots,S^J) \quad (6)$$

Applying the same hypothesis as in equation 5, the second term of previous equation can be simplified into:

$$K(\theta^1,\ldots,\theta^J|S^1,\ldots,S^J) \leq \sum_{i=1}^{J} K(\theta^i|S^i) \quad (7)$$

Not much attention will be given to the terms $K(\theta^i|S^i)$ in the following. We can consider these terms as constant. In our context, this term has no real impact as the model parameters are evaluated using the research bias of the corresponding clustering algorithm. Otherwise, it would play a role of *prior* such as in the Bayesian setting.

The solution complexity is obtained by applying the chain rule recursively over all the solutions. Besides, as we do not consider a hierarchy between models (hence an ordering of the solution), we assume that any order can be used and consider an upper-bound in which each solution S^i is described with help of all other solutions (denoted S^{-i}):

$$K(S^1,\ldots,S^J) \leq \sum_{i=1}^{J} K(S^i|S^1,\ldots,S^{i-1}) \leq \sum_{i=1}^{J} K(S^i|S^{-i}) \quad (8)$$

Because a solution S^j is part of the set S^{-i}), a description of S^i provided S^{-i} is necessarily shorter than a description of S^i provided S^j. In mathematical terms, this corresponds to saying that $K(S^i|S^{-i}) \leq K(S^i|S^j)$ for all j (see Corollary 1.4.2 in [Gác88]). Thus, the following upper-bound can be considered for the total solution complexity:

$$K(S^1,\ldots,S^J) \leq \sum_{i=1}^{J} K(S^i|S^{-i}) \leq \frac{1}{J-1}\sum_{i\neq j} K(S^i|S^j) \quad (9)$$

In equation 9, the coefficient $1/(J-1)$ is a consequence of very general bounds. Depending on the problem, the parameters can be refined. In a more general framework (which will not be considered in the scope of this paper), coefficients $\alpha_{i,j}$ could be considered in the solution transfer term [SMM17] .

3.4 Final problem

After all proposed simplifications, the final objective function for collaborative clustering based on MDL principle is:

$$\sum_{i=1}^{J}\left(K(X^i|M^i) + K(\theta^i|S^i) + \frac{1}{J-1}\sum_{j\neq i} K(S^i|S^j)\right) \quad (10)$$

The MDL principle assumes that the optimal models $M^1,\ldots,M^J$ have to minimize this quantity. In equation 10, as in any expression involving complexity given in this paper, we did not precise constant terms. All equalities and inequalities presented are defined up to a constant as stated in the invariance theorem. This constant is not a problem to us though, as we consider only complexity differences and the constant depends only on the Turing machine which is supposed to be fixed in our problem.

4 Local clustering complexity

In this section, we present clustering as a particular case of compression with the help of two families of clustering models: prototype-based models and probabilistic models. We show that this compression leads to an expression of the complexity $K(X^i|M^i)$ and discuss data encoding relative to the models.

4.1 Compression with prototype-based models

Prototype-based methods form a class of methods in which data points are described by their relative position to virtual points called *prototype*. The idea behind prototype-based methods is that the information contained in the absolute position of the prototype has not to be repeated for every associated point. Various prototype-based methods are usually employed in clustering, including K-means [Llo06] and Self-Organizing Maps [KSH01].

A description of a prototype-based model M is offered by the description of its prototypes. We suppose here that the parameter θ for the model M consists simply in the list of the prototypes. A prototype P is a virtual point (ie. P does not necessarily belong to the data set) taken as a reference for the description of close data points. A prototype is described by its absolute coordinates. Considering the independence of prototypes inside a prototype model, we have $K(\theta) = \sum_{P\in M} K(P)$. By definition, we have in particular $K(\theta|S) \leq K(\theta)$.

Given a model, a point is associated to the closest prototype in the model and the relative coordinates to it. This description is to determine an upper-bound of Kolmogorov complexity:

$$K(X|M) \leq \sum_{n=1}^{N} K(X_n|M) \leq \sum_{n=1}^{N} \min_{P \in M} K(X_n|P) \leq \sum_{n=1}^{N} \min_{P \in M} K(X_n - P) \tag{11}$$

where X is a full data matrix, N the number of data in X and X_n is the n-th point in the data set.

4.2 Compression with probabilistic models

Some clustering models are based on probabilistic models. Among them, the Gaussian Mixture Models describes data points as generated by a mixture of normal distributions. In a perspective of hard clustering, a point is associated to the distribution maximizing its conditional distribution.

By construction, probabilities and complexity cannot be directly correlated, since probabilities model the global generation model of a source whereas complexity measures the amount of information contained in a single message independently of the source. A probability distribution can be seen as a compression tool yet: Intuitively, an event of high probability is simple to describe. If x is an element of a given ensemble $\mathcal{X}$ and μ an semi-computable probability distribution over $\mathcal{X}$, we have $K(x) \leq K(\mu) - \log \mu(x)$ and in particular $K(x|\mu) \leq -\log \mu(x)$.

Thus, a probabilistic model can be used for data compression in our framework. In particular, Gaussian Mixture Models can be employed as a clustering algorithm in our framework. An upper-bound of Kolmogorov complexity induced by a probabilistic model is given by:

$$K(X|M) \leq \sum_{n=1}^{N} K(X_n|M) \leq \sum_{n=1}^{N} -\log \mu(X_n) = -\mathcal{L}(X) \tag{12}$$

where $\mathcal{L}$ designates the log-likelihood of a set.

5 Algorithm

In this section, we explain how we optimize the objective function in equation 10. In the case of this article, we consider only the case where the solutions $S^1, \cdots, S^J$ produced by the algorithms are hard partitions, and therefore can be described as vectors.

5.1 Global approach

Following the model of other collaborative and multi-view algorithms, the optimization is done in 2 steps [GB10, SGBC15]:

- A **local step** during which each algorithm $\mathcal{A}^i$ processes its local view X^i and produces a first model $M^i = \langle \theta^i, S^i \rangle$ based only on the local information. These local models are used as initial values.
- A **global step** during which equation (10) is optimized using the MDL principle and Kolmogorov complexity.

The key difficulty of the algorithm lies therefore in the global step, and in particular in the estimation of the complexity $K(S^i|S^j)$. Our idea to evaluate a lower-bound of the complexity $K(S^i|S^j)$ is to build a naive mapping from S^i to S^j. To do so, we consider the confusion matrix $\Omega^{i,j}$ that maps the clusters of S^i to the clusters of S^j.

$$\Omega^{i,j} = \begin{pmatrix} \omega^{i,j}_{1,1} & \cdots & \omega^{i,j}_{1,K_j} \\ \vdots & \ddots & \vdots \\ \omega^{i,j}_{K_i,1} & \cdots & \omega^{i,j}_{K_i,K_j} \end{pmatrix} \text{ where } \omega^{i,j}_{a,b} = |S^i_a \cap S^j_b| \tag{13}$$

From there an *argmax* on each line of $\Omega^{i,j}$ in equation 13 gives us the majority mapping rule for each cluster of $\mathcal{A}^i$ into a cluster of $\mathcal{A}^j$. Using this method, a compression is obtained by defining a general mapping transforming all labels of S^i into labels of S^j and correcting the errors afterwards. The time complexity to compute all the rules between all solutions vectors using this method is in $\mathcal{O}(N)$ for solutions vectors of length N.

As depicted in figure 1, the transformation from solution S^j into solution S^i is described with the help of a set of associative rules mapping a cluster from S^j into a cluster from S^j. In general, such a mapping does not have any noticeable property: in particular, it is neither injective nor surjective. We define a mapping as a function $\mathcal{R}_{j,i} : \{1, \ldots, K_j\} \mapsto \{1, \ldots, K_i\}$. We propose to encode the mapping as a key-value set $\langle (1, \mathcal{R}_{j,i}(1)), \ldots, (K_j, \mathcal{R}_{j,i}(K_j)) \rangle$. A cluster by cluster mapping between two solutions is often not sufficient to offer a full description of a transformation from one solution into another: some exceptions have to be added to describe the exact transformation. An exception is encoded as a tuple $(n, k_i) \in \{1, \ldots, N\} \times K_i$ and overwrites the transformation rule. The set of exceptions for the transformation of S^j into S^i is denoted $\mathcal{E}_{j,i}$

We use this language of rules and exceptions to describe solutions in the term $K(S^i|S^j)$: the pure complexity is upper-bounded by the complexity of the rule $\mathcal{R}_{j,i}$ and the associated exceptions:

$$K(S^i|S^j) \leq K\left(\mathcal{R}_{j,i}\right) + \sum_{e \in \mathcal{E}_{j,i}} K(e) \tag{14}$$

In equation 14, the complexity terms for rules and exceptions are defined as the sum of the individual complexities of components of the corresponding tuple: $K\left(\mathcal{R}_{j,i}\right) = K(k_i) + K(k_j)$ and $K(e) = K(n) + K(k_i)$. We choose to encode all elements of a same set with the same number of bits. Any element of a set of p elements can be encoded on a prefix-machine with $K(p)$ bits. As stated in section 3.1 of [LV08], we have $K(p) = C(p) + C(C(p)) + \mathcal{O}(C(C(C(p))))$ where $C(.)$ designates the non-prefix complexity. We have in particular $C(p) \leq \log p$, hence $K(p) \leq \log p + \log\log p + \mathcal{O}(\log\log\log p)$. In particular, knowing that in our cases of interest the size of the sets cannot be arbitrarily large, we can suppose that there exists a constant c independent of p such that $K(p) \leq \log p + c$. We do not use this constant in practice, since we are only interested in variations of complexity. Consequently, we will use the following upper-bound for the solution transfer term:

$$\begin{aligned} K(S^i|S^j) \leq K^j \times \left(\log K^j + \log K^i\right) \\ + |\mathcal{E}_{j,i}| \times \left(\log N + \log K^i\right) \end{aligned} \tag{15}$$

Given these elements, optimizing equation 10 consists in searching for the error corrections that would have the most positive impact on the collaborative term $\sum_{j \neq i} K(S^i|S^j)$ with a minimal impact on the local term $K(X^i|M^i)$. Corrections that do not improve the collaborative term or have a negative impact are ignored.

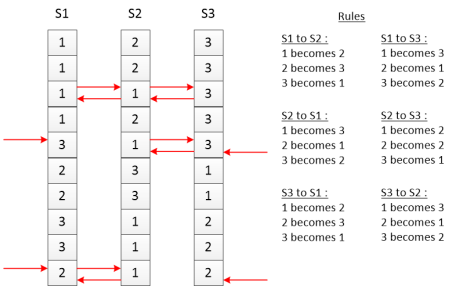

Figure 1: Examples of majority rules (on the right) and potential errors to correct (in red)

5.2 Description of the algorithm

We decompose the algorithm into three main steps:

1. **Local optimization**: Local algorithms compute clustering solutions to the corresponding view.

2. **Solution mapping**: Mappings (as defined in section 5.1) are found for any pair of solutions $(S^i|S^j)$

3. **Mapping optimization**: The mappings are slightly corrected in order to make global complexity decrease.

The local optimization step consists in a parallel run of all local clustering algorithms. Because there is no collaboration in the local term in equation 10, algorithms can run without any interaction. We notice that we do not aim at minimizing the expression of complexity directly, but we use standard algorithms instead: The clustering algorithms are seen as research biases for the minimization of complexity.

The solution mapping involves a one-by-one pairing of solutions. It can be decomposed into two steps: First the algorithm determines the rules by selecting the maximal cluster associations based on the confusion matrix (as explained in section 5.1 and equation 13). The time complexity of this step is $\mathcal{O}(N \times J^2)$. The complete algorithm is detailed in Algorithm 1.

Algorithm 1: SolutionMapping

Input: A set of J clustering solutions $S^1, \ldots, S^J$
Output: A set of rules $\{\mathcal{R}_{j,i}\}_{1 \leq i,j \leq J}$ and exceptions $\{\mathcal{E}\}_{1 \leq i,j \leq J}$

```
for i = 1...J do
    for j = 1...J do
        Compute Ω^{i,j}
        for k = 1...K^i do
            R_{j,i}[k] ← arg max_l Ω^{i,j}_{k,l}
        for n = 1...N do
            if R_{j,i}[S^j[n]] ≠ S^i[n] then
                E_{j,i}[n] ← S^i[n]
return {R_{j,i}}_{1≤i,j≤J}, {E_{j,i}}_{1≤i,j≤J}
```

The mapping optimization is the most complex step of the method. It consists in removing exceptions one by one in the obtained set $\{\mathcal{E}_{j,i}\}_{1 \leq i,j \leq J}$. Removing an exception results in a single change inside a clustering solution. The system decides to remove an exception

if this deletion leads to a reduction in complexity. Because a deletion modifies the solutions, the deletion order has importance in this algorithm. Thus, the naive algorithm cannot be used here. Instead, we choose a greedy approach selecting only the exception leading to the highest complexity reduction. The obtained solution is not guaranteed to be the global minimum but a only local minimum. At each step, the algorithm explores the set of exceptions and computes the difference in total complexity induced by the deletion of the current exception. The algorithm is exposed in Algorithm 2.

At each step, the algorithm has access to a finite list of exceptions and removes the exception which corresponds to the highest complexity reduction: From one step to another, the complexity can only decrease. Because the number of possible solutions is finite and the total complexity is necessarily non-negative, the algorithm must converge in a finite number of steps. Hence, no stop criterion has to be given.

Algorithm 2: GreedyMappingOptimization

Input: A set of J models $(\theta^1, S^1), \ldots, (\theta^J, S^J)$; A set of rules $\{\mathcal{R}_{j,i}\}_{i,j}$ and exceptions $\{\mathcal{E}\}_{i,j}$
Output: Modified solutions $S^1, \ldots, S^J$
$\Delta K_{min} = 0$
while *Exceptions left* **do**
  for $e = (j, i, n, k) \in \mathcal{E}$ **do**
    $\Delta K \leftarrow K(X^i[n]|\theta^i, S^i[n]) - K(X^i[n]|\theta^i, \mathcal{R}_{j,i}\left[S^j[n]\right])$
    $\tilde{\mathcal{E}} \leftarrow \mathcal{E} \backslash e$
    for $l = 1 \ldots J$ **do**
      if *Transformation from* S^i *to* S^l *admits an exception in* n **then**
        if *Exception is corrected with new value* **then**
          $\Delta K \leftarrow \Delta K - (K(N) + K(K^l))$
          Remove n from $\tilde{\mathcal{E}}_{l,i}$
        else $\tilde{\mathcal{E}}_{l,i}[n] \leftarrow \mathcal{R}_{j,i}\left[S^j[n]\right]$
      else $\Delta K \leftarrow \Delta K + (K(N) + K(K^l)); \tilde{\mathcal{E}}_{l,i}[n] \leftarrow \mathcal{R}_{j,i}\left[S^j[n]\right]$
      if *Transformation from* S^l *to* S^i *admits an exception in* n **then**
        if *Exception is corrected with new value* **then**
          $\Delta K \leftarrow \Delta K - (K(N) + K(K^i))$
          Remove n from $\tilde{\mathcal{E}}_{i,l}$
        else $\tilde{\mathcal{E}}_{i,l}[n] \leftarrow \mathcal{R}_{j,i}\left[S^j[n]\right]$
      else $\Delta K \leftarrow \Delta K + (K(N) + K(K^i)); \tilde{\mathcal{E}}_{i,l}[n] \leftarrow \mathcal{R}_{j,i}\left[S^j[n]\right]$
      if $\Delta K < \Delta K_{min}$ **then**
        $\Delta K_{min} \leftarrow \Delta K; \mathcal{E}_{min} \leftarrow \mathcal{E}$
  if *A modification has been found* **then**
    $\mathcal{E} \leftarrow \mathcal{E}_{min}$; Modify S
return *Modified solutions* S

6 Applications

6.1 Datasets

In this section, we propose an applicative setting in which we used our proposed method on various multi-view data sets, real and artificial.

We considered the following data sets:

- The Wisconsin Data Breast Cancer (UCI): This data set contains 569 instances with 30 parameters and 2 classes. These 30 parameters contain 10 descriptors for 3 different cells (10 each) of the same patient. This data set can easily be split into 3 views: one for each cell.

- The Spam Base data set (UCI): The Spam Base data set contains 4601 observations described by 57 attributes and a label column: Spam or not Spam (1 or 0). The different attributes can be split into views containing word frequencies, letter frequencies and capital run sequences attributes.

- The VHR Strasbourg data set[1] [RP14]: It contains the description of 187.058 segments extracted from a very high resolution satellite image of the French city of Strasbourg. Each segment is described by 27 attributes that can be split between radiometical attributes, shape attributes, and texture attributes. Furthermore, the color attributes can also be split between Red, Blue and near-infrared

attributes. The data set is provided with a partial hybrid ground-truth containing 15 expert classes.

- The Battalia3 data set[1] (artificial): Battalia3 is an artificial dataset created using the exoplanet random generator from the online game Battalia.fr; This data set describes 2000 randomly generated exoplanets with 27 numerical attributes and their associated class (6 classes). The attributes can be split between system and orbital parameters (7 attributes), planet characteristics (10 attributes) and atmospheric characteristics (10 attributes).

- The "MV2" data set (artificial)[1]: A data set created specifically to test this kind of algorithm. It features 2000 randomly generated data, split into 4 views of 6 attributes each, and a total of 4 classes. All attributes were generated either from Gaussian distributions with parameters linked to the matching class, or are random noise, or are linear combinations of other attributes.

6.2 Experimental results

To assess the effectiveness of our proposed method, in this section we propose an experiment in which we compare it with four other collaborative and multi-view method from the literature: The collaborative clustering framework for heterogeneous algorithms (CCHA) [SGBC15] with EM algorithms for the Gaussian Mixture Model (diagonal covariance matrix) collaborating together, a re-implementation of the multi-view EM algorithm [BS05] for the Gaussian Mixture Model with diagonal covariance matrix as well, the collaborative GTM algorithm [GGB12] (complete variance-covariance matrix) and the collaborative SOM algorithm [Nis09]. For our method, we used a gaussian mixture model (complete variance-covariance matrix) in each view.

The 3 methods are compared using two unsupervised indexes: the Davies-Bouldin index [D.L74] (DBI) and the Silhouette index [Rou87] (Sil.), both of which assess in different ways the quality of the cluster in term of compacity and whether or not they are well separated. The Davies-Bouldin index is a positive not normalized index the value of which is better when it is lower. The Silhouette index is a normalized index which takes values between -1 and 1, 1 being the best possible value.

Furthermore, since all data sets were acquired from originally supervised problems, they were all provided with available labels. Consequently, in our experiments, we also used the Rand Index [Ran71] based on the original classes as an external index.

In Table 1, we show the average results achieved on the unsupervised indexes at the end for the multi-view or collaborative process. The results for the supervised indexes (Rand index) are shown in Table 2. Both the Davies-Bouldin index and the Silhouette index where computed using the partitions found on the local views and the complete data as reference.

As one can see, our method achieves competitive results on both supervised and unsupervised indexes when compared with other state of the art multi-view and collaborative frameworks. Furthermore, the average indexes reached by our algorithm even show this we do slightly better than the other methods. Clustering being a mature domain, small improvements are already a step in the right direction. Futhermore, the strength of our method does not lie only in its capacity to achieve competitive results -as proved in these experiments-, but more in its genericity and strong mathematical background.

Finally, we would like to mention that comparing multi-view methods originally designed for different types of applications and with vastly different architectures is a difficult task. Therefore, this shorts experiments may not be fully representative of the full potential of any of the methods compared in this section, ours included.

7 Conclusion

In this paper, we proposed a new approach for multi-view and collaborative clustering based on algorithmic theory of information. We introduced the Minimum Description Length Principle as a useful inductive principle in the context of collaborative clustering. The MDL principle states that the optimal model (hence the optimal clustering solution) corresponds to a minimal value for the description length of the model and of the data, measured by Kolmogorov complexity. Because Kolmogorov complexity is not calculable, we defined our objective function as an upper-bound of pure complexity obtained with straightforward simplifying hypotheses. We proposed a minimization algorithm based on a greedy exploration of the description space and compared its performances on standard data sets with the state-of-the-art algorithms. Our method outperforms standard algorithms on most data sets and presents highly similar results on the others. However, it offers a very general and theoretically grounded

[1] Available from Dr. J. Sublime ResearchGate account

Dataset	**Our Model**		MV-EM		CCHA		GTM_{col}		SOM_{col}	
	DBI	Sil.	DBI	Sil.	DBI	Sil.	DBI	Sil.	DBI	Sil.
WDBC	**0.98**	**0.55**	1.63	0.42	1.63	0.42	1.8	0.37	1.68	0.41
SpamBase	**3.08**	**0.19**	4.77	0.086	4.73	0.085	4.60	0.093	4.35	0.113
VHR Strasbourg	3.46	0.14	3.21	0.12	**2.89**	**0.175**	4.15	0.073	3.78	0.098
Battalia3	**2.29**	0.25	2.43	0.16	2.83	0.14	2.68	**0.35**	2.51	0.34
MV2	1.61	0.37	**1.34**	0.35	**1.34**	0.35	1.61	0.38	1.44	**0.39**

Table 1: Experimental results: raw average results on unsupervised indexes

Dataset	**Our Model**	MV-EM	CCHA	GTM_{col}	SOM_{col}
	Rand	Rand	Rand	Rand	Rand
WDBC	0.73	0.79	0.87	0.96	**0.97**
SpamBase	0.76	0.74	**0.86**	0.83	0.84
VHR Strasbourg	**0.78**	0.73	0.75	0.69	0.70
Battalia3	**0.86**	0.78	0.80	0.78	0.79
MV2	**0.93**	**0.93**	**0.93**	0.90	0.90

Table 2: Experimental results: raw average results on the Rand Index

framework to address the issue of unsupervised ensemble learning.

As a general framework, the proposed methodology offers a large variety of perspectives. To take advantage of its generality, an extension to other classes of clustering algorithms (e.g. density based clustering, spectral clustering...) is needed: A proper definition of the quantity $K(X|M)$ is needed for these algorithms. Besides, we proposed some simplifying hypotheses but other less restrictive hypotheses might be found and lead to more accurate results. As a complement, algorithmic issues have to be overcome in future works. On the one hand, an adaptation to large data sets has to be considered to obtain a scalable method. On the other hand, the mapping optimization presented in this paper is based on a greedy approximation: An exact computation cannot be performed in a naive way, but could be reached by more subtle algorithms.

Acknowledgments

The authors would like to thank Associate Professor Nistor Grozavu for providing experimental results with his algorithms and Associate Professor Jean-Louis Dessalles for his insightful remarks.

This research is supported by the programme Futur & Ruptures (Institut Mines-Telecom).

References

[BS05] Steffen Bickel and Tobias Scheffer. Estimation of mixture models using co-em. In João Gama, Rui Camacho, Pavel Brazdil, Alípio Jorge, and Luís Torgo, editors, *Machine Learning: ECML 2005, 16th European Conference on Machine Learning, Porto, Portugal, October 3-7, 2005, Proceedings*, volume 3720 of *Lecture Notes in Computer Science*, pages 35–46. Springer, 2005.

[DFVW11] Benoît Depaire, Rafael Falcon, Koen Vanhoof, and Geert Wets. PSO Driven Collaborative Clustering: a Clustering Algorithm for Ubiquitous Environments. *Intelligent Data Analysis*, 15:49–68, January 2011.

[D.L74] D.W. Bouldin D.L. Davies. A cluster separation measure. *IEEE Trans. Pattern Anal. Machine Intell.*, 1 (4):224–227, 1974.

[FWG07] G. Forestier, C. Wemmert, and P. Gancarski. Collaborative multi-strategical classification for object-oriented image analysis. In *Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications in conjunction with IbPRIA*, pages 80–90, June 2007.

[Gác88] Peter Gács. *Lecture notes on descriptional complexity and randomness*. Boston University, Graduate School of Arts and Sciences, Computer Science Department, 1988.

[GB10] Nistor Grozavu and Younès Bennani. Topological collaborative clustering. *Australian Journal of Intelligent Information Processing Systems*, 12(3), 2010.

[GGB12] Mohamad Ghassany, Nistor Grozavu, and Younès Bennani. Collaborative clustering using prototype-based techniques. *International Journal of Computational Intelligence and Applications*, 11(3), 2012.

[GSIM09] R. Ghaemi, N. Sulaiman, H. Ibrahim, and N. Mustapha. A survey: Clustering ensembles techniques. *PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY*, 38, February 2009.

[KI11] Abhishek Kumar and Hal Daumé III. A co-training approach for multi-view spectral clustering. In Lise Getoor and Tobias Scheffer, editors, *ICML*, pages 393–400. Omnipress, 2011.

[KSH01] T. Kohonen, M. R. Schroeder, and T. S. Huang, editors. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd edition, 2001.

[Llo06] S. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theor.*, 28(2):129–137, September 2006.

[LV08] Ming Li and Paul M.B. Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Publishing Company, Incorporated, 3 edition, 2008.

[Nis09] Younès Bennani Mustapha Lebbah Nistor Grozavu. From variable weighting to cluster characterization in topographic unsupervised learning. In *in Proc. Proc. of IJCNN09, International Joint Conference on Neural Network*, 2009.

[Ped02] Witold Pedrycz. Collaborative fuzzy clustering. *Pattern Recognition Letters*, 23(14):1675–1686, 2002.

[Ran71] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association.*, pages 846–850, 1971.

[Ris78] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465 – 471, 1978.

[Rou87] R.J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics.*, 20:53–65, 1987.

[RP14] S. Rougier and A. Puissant. Improvements of urban vegetation segmentation and classification using multi-temporal pleiades images. *5th International Conference on Geographic Object-Based Image Analysis*, page 6, 2014.

[SGBC15] Jérémie Sublime, Nistor Grozavu, Younès Bennani, and Antoine Cornuéjols. Collaborative clustering with heterogeneous algorithms. In *2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-18, 2015*, 2015.

[SMM17] Jérémie Sublime, Basarab Matei, and Pierre-Alexandre Murena. Analysis of the influence of diversity in collaborative and multi-view clustering. In *2017 International Joint Conference on Neural Networks, IJCNN 2017*, 2017.

[VR11] Sandro Vega-Pons and José Ruiz-Shulcloper. A survey of clustering ensemble algorithms. *IJPRAI*, 25(3):337–372, 2011.

[WB68] C. S. Wallace and D. M. Boulton. An information measure for classification. *The Computer Journal*, 11(2):185–194, 1968.

[Wol96] David H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Comput.*, 8(7):1341–1390, October 1996.

[ZV15] Arthur Zimek and Jilles Vreeken. 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.