Collaborative clustering with background knowledge

Germain Forestier, Pierre Gançarski, Cédric Wemmert

This is the author's version of an article published in Data & Knowledge Engineering. The final authenticated version is available online at: http://dx.doi.org/10.1016/j.datak.2009.10.004.

Collaborative clustering with background knowledge

Germain Forestier[1], Pierre Gançarski[1], Cédric Wemmert[1]

[1] *University of Strasbourg, LSIIT, UMR7005, Pôle API, Bd Sébastien Brant, BP 10413, 67412 Illkirch, Cedex, France*

Abstract

The aim of collaborative clustering is to make different clustering methods collaborate, in order to reach at an agreement on the partitioning of a common dataset. As different clustering methods can produce different partitioning of the same dataset, finding a consensual clustering from these results is often a hard task. The collaboration aims to make the methods agree on the partitioning through a refinement of their results. This process tends to make the results more similar. In this paper, after the introduction of the collaboration process, we present different ways to integrate background knowledge into it. Indeed, in recent years, the integration of background knowledge in clustering algorithms has been the subject of a lot of interest. This integration often leads to an improvement of the quality of the results. We discuss how such integration in the collaborative process is beneficial and we present experiments in which background knowledge is used to guide collaboration.

Keywords: collaborative clustering, unsupervised learning, classification, pattern recognition, knowledge-guided clustering

1. Introduction

Clustering is an important machine learning tool for discovering hidden patterns, structures and relationships between data objects in an unsupervised way. It has been widely used in pattern recognition fields, mainly to classify groups of measurements or observations. It finds applications in bioinformatics, information retrieval, medicine, image analysis or financial markets, to structure information and discover hidden knowledge.

Over the last fifty years, a huge number of new clustering algorithms have been developed, and existing methods have been modified and improved [1, 2, 3, 4, 5]. This abundance of methods can be explained by the ill-posed nature of clustering. Indeed, each clustering algorithm is biased by the objective function used to build the clusters. Consequently, different methods can, from the same data, produce very different clustering results. Furthermore, even the same algorithm can produce different results, according to its parameters and initialization. A relatively recent approach to circumvent the problem is based on the idea that the information offered by different sources and different clustering, are complementary [6]. Thus, the combination of different clusterings may increase their efficiency and accuracy. A single classification is produced from results of methods having different points of view: each individual clustering opinion is used to find a consensual decision. Each decision can be processed from a different source or media.

In the same way, we address the problem of the collaboration between different clustering methods. Collaboration is a process where two or more actors work together to achieve a common goal by sharing knowledge. In our collaborative clustering method, called Samarah, different clustering methods work together to reach an agreement on their clustering. Each clustering modifies its results according to all the other clusterings until all the clusterings proposed by the different methods are strongly similar. Thus, they can be more easily unified, for example, through a voting algorithm.

Email address: forestier@unistra.fr (Germain Forestier)

Different studies showed that this method provides interesting results on artificial data sets and on real life problems. However, a lot of work is currently focusing on integrating background knowledge into the clustering process. This work highlights the benefits of knowledge integration into this process by showing that the use of such knowledge to drive the process leads to more accurate results. The aim of this paper is to present an extension of SAMARAH, which takes into account background knowledge during the collaboration of the methods. The background knowledge is used to drive the collaboration between the clustering methods and allows an improvement of the final results.

To evaluate benefits of using the collaborative clustering method SAMARAH with (and without) background knowledge integration, we used classical approaches based on quality indexes and a more recent approach called *cascade evaluation*. Cascade evaluation [7] is a new approach to evaluate the quality and the interest of clustering results. The method is based on the enrichment of a set of labeled datasets by the results of clusterings, and the use of a supervised method to evaluate the benefit of adding such new information to the data sets.

The cascade evaluation of the SAMARAH method highlights that collaboration increases the quality of the refined results in both cases: with and without the use of background knowledge during the collaboration

This paper is organized as follows. First, we describe work related to our approach (Section 2). In particular, we introduce clustering methods using background knowledge (Section 2.4). Section 3 describes the collaborative clustering method SAMARAH and presents the knowledge integration into it. Then, in Section 4, after a brief introduction to cascade evaluation principles, the evaluations of the collaborative clustering on various datasets from the UCI repository are detailed and discussed. Finally conclusions and perspectives are drawn in Section 5.

2. Related works

In recent years, a lot of work has focused on the use of multiple clusterings to improve the unsupervised classification process. Indeed, many different algorithms exist and may provide different results from the same dataset. Consequently, it is often difficult to design a single algorithm whose results reflect what users need and expect. To cope with this problem, the ensemble clustering approaches (Section 2.1) consist in designing function which summuarize several clustering into a single one. The aim is to find the average partition which is the most similar to all the results of the ensemble. However, the ensemble clustering methods only focus on the creation of the consensus and do not modify or create new partitions. Consequently, they consider that the initial provided partititions are the only ones necessary.

Alternatively, the multiobjective approaches (Section 2.2) see the clustering process as an optimization of different objectives. In this methods, a vast amount of different clustering results are explored, and the ones which best match the objectives are keept. A genetic algorithm is generaly used and new clusterings are created by mixing different results together. However, the objective use in the optimization are not always the objetive of the algorithm used to create the initial partation, and thus induce a new bias.

In the fuzzy collaborative clustering approaches (Section 2.3), the fuzzy c-means alogrithm is use to cluster different views of the data in a collaborative way. This work only focuses on the fuzzy c-means algorithm which limits its use. However, its prodive strong theorical basis which better highlight the benefit of the collaboration in specific environement.

The use of different clusterings result often require a strong implication of the user, as there is a lot of parameter to choose. To reduce the need of the user and improve the quality of the results, a lot of work has also focused on the use of background knowledge (Section 2.4). Background knowlegde has many different representations like a set of labeled patterns, link between patterns, the number of expected clusters, the expected size of the clusters, etc.

2.1. Ensemble clustering

The aim of ensemble clustering is to generate multiple clusterings and to merge them to produce a final consensus clustering. The initial clusterings are generally generated by applying different algorithms, using different parameters of the same algorithm, or by random sampling of the dataset. Ensemble clustering is

often used to improve the accuracy of the data clustering or to find complex shaped clusters. Indeed, using multiple clusterings allows ensemble clustering algorithms to better grasp the underlying distribution of the data space.

Strehl et al. [8] formulated the consensus clustering as the partition that maximize the shared information among the ensemble of initial clusterings. This information is measured through the Average Normalized Mutual Information (ANMI) which uses the information theory framework. Three different clustering ensemble strategies based on graph theory are presented : Cluster-based Similarity Partitioning Algorithm (CSPA), Hyper Graph Partitioning Algorithm (HGPA), and Meta Clustering Algorithm (MCLA). Further work reveals that these approaches are sensitive to cluster sizes and seek only for balanced-size clusters (i.e. all the clusters tend to have the same number of data objects).

Another approach, called evidence accumulation, is presented by Fred et al. [9]. The main idea is to produce a co-association matrix from the different initial clusterings. This matrix gives the information of the number of times that two data objects have been put together in the same cluster. A hierarchical clustering is then used, using the co-association matrix as a distance matrix, to cluster the objects into the final partition. Various approaches using the co-association matrix have been presented in the literature and seem to outperform the graph based methods.

Topchy et al. [10] described how to create a new feature space from an ensemble clustering by interpreting the multiple clusterings as a new set of categorical features. The KMEANS algorithm is applied on this new standardized feature space using a category utility function to evaluate the quality of the consensus. Topchy et al. [11] also demonstrated the efficiency of combining partitions generated by weak clustering algorithms that use data projections and random data splits.

Hadjitodorov et al. [12] presented an evaluation of different heuristics to produce the initial set of clusterings. The most often selected heuristics were random feature extraction, random feature selection and random number of clusters assigned to each member of the ensemble.

Ayad and Kamel [13] present a cumulative voting method to come to a consensus from partitions with a variable number of clusters. They described several cumulative vote weighting schemes and corresponding algorithms, to compute an empirical probability distribution summarizing the partitions. The empirical study shows the efficiency of the method compared to others consensus clustering algorithms.

More voting methods are given by Nguyen et al. [14]. Three iterative algorithms are presented: Iterative Voting Consensus (IVC), Iterative Probabilistic Voting Consensus (IPVC) and Iterative Pairwise Consensus (IPC). These algorithms use a feature map built from the set of base clusterings and apply an EM-like consensus clustering.

To directly address the correspondence problem (among the clusters of different clusterings) in combining multiple clusterings, Zhang et al. [15] introduced a new framework based on soft correspondence. The proposed algorithm provides a consensus clustering method as well as correspondence matrices that give the relations between the clusterings of the ensemble.

Hu et al. [16] proposed a method which uses Markov random fields and maximum likelihood estimation to define a metric distance between clusterings. They present two combining methods based on this new similarity to find a consensus.

More recently, Pedrycz et al. [17] proposed a consensus-driven fuzzy clustering method. The authors consider the proximity matrices induced by the corresponding partition matrices. An optimization scheme is presented in detail along with a way of forming a pertinent criterion. This criterion governs an intensity of collaboration to guide the process of consensus formation.

The ensemble clustering approaches do not generally address the problem of the generation of the initial results, and the algorithms used to create the initial results are not used in the combination process. Consequently, ensemble clustering approaches introduce a new bias, relative to the objective function chosen when merging the different clusterings.

2.2. Multiobjective clustering

The aim of multiobjective clustering is to optimize simultaneously several clustering criteria. The idea is to have a better grasp of the notion of cluster by explicitly defining them with different objective functions.

Algorithms are able to produce a set of trade-off solutions between the different objectives. These solutions correspond to different compromises of the objectives used.

Thus, the method MOCK (Multi-Objective Clustering with automatic K-determination) [18] uses two objectives: the first one is to maximize the compactness of the clusters, and the second one their connectivity. A multiobjective evolutionary algorithm is used to optimize these two criteria simultaneously. The method uses a Pareto based approach [19] which consists in selecting the non-dominated solutions of the Pareto front. At the end of the evolution, the solutions on the Pareto front are the set of solutions provided by the algorithm. A heuristic is then used to select the best potential solution by using the number of clusters of the solutions on the front. In [20], the authors present how to integrate background knowledge through a third objective optimization which uses a subset of labeled samples. This semisupervised version outperformed the solution without background knowledge.

Faceli et al. [21] described the multiobjective method called MOCLE (Multi-Objective Clustering Ensemble) which integrates the same objective function (maximization of compactness and connexity of the clusters) as MOCK. But a special crossover operator which uses ensemble clustering techniques is added: the purpose of the MOCLE method is to produce a set of solutions which are a trade-off of the different objectives, while the MOCK method produces a single solution.

Law et al. [22] proposed a method which uses different clustering methods using different objectives. The final result is produced by selecting clusters among the results proposed by the different methods. A resampling method is used to estimate the quality of the clusters.

A semisupervised extension of MOCLE has also been proposed recently [23]. The prior knowledge about a known structure of the data is integrated by means of an additional objective function that takes external information into account. The new objective aims at creating pure clustering according to known object class label. The objective is optimized along with the previous one on the data and the use of background knowlege improved the result of the method.

An evolutionary version of the KMEANS algorithm is used by [24], driven by a semisupervised objective. This objective is a weighted sum of the mean squared error (MSE) and the purity of the clusters according to a subset of available samples. Different criteria are investigated to quantify this purity.

In [22], the authors proposed a method which uses different clustering methods using different objectives. The final result is produced by selecting clusters among the results proposed by the different methods. A resampling method is used to estimate the quality of the clusters.

2.3. Fuzzy collaborative clustering

A fuzzy clustering architecture is introduced by Pedrycz et al. [25], in which several subsets of patterns can be processed together to find a common structure to all of them. In this system, different subsets of the initial data are processed independently. Then, each partition matrix is modified according to the other matrices found: each result produced on a subset is modified according to results found on the other subsets. Extensive experiments of the method are also proposed in [26] along with algorithmic details. An application of this collaborative fuzzy clustering method to semantic web content analysis has be proposed in [27]. The authors discuss of a collaborative proximity-based fuzzy clustering and show how this type of clustering is used to discover a structure of web information in the spaces of semantics and data.

A fuzzy collaborative framework is also proposed [28], where rough sets are used to create a collaborative paradigm in which several subsets of patterns are processed together to find a common structure. A clustering algorithm is developed by integrating the advantages of both fuzzy sets and rough sets. A quantitative analysis of the experimental results is also provided for synthetic and real-world data.

To tackle the problem of distributed data, [29] proposed a framework to cluster distributed classifier. They show that clustering distributed classifiers as a pre-processing step for classifier combination enhances the achieved performance of the ensemble.

2.4. Clustering with background knowledge

Many approaches have been investigated to use background knowledge to guide the clustering process.

In constrained clustering, knowledge is expressed as *must-link* and *cannot-link* constraints and is used to guide the clustering process. A *must-link* constraint gives the information that two data objects should be in

the same cluster, and *cannot-link* means the opposite. This kind of knowledge is sometimes easier to obtain than a classical subset of labeled samples. Wagstaff et al. [30] presented a constrained version of the KMEANS algorithm which uses such constraints to bias the assignment of the objects to the clusters. At each step, the algorithm tries to agree with the constraints given by the user. These constraints can also be used to learn a distance function biased by the knowledge about the links between the data objects [31]. The distance between two data objects is reduced for a must-link and increased for a cannot-link. Huang et al. [32] presented an active learning framework for semi-supervised document clustering with language modeling. The approach uses a gain-directed document pair selection method to select cleverly the constraints. In order to minimize the amount of constraints required, Griga et al. [33] defined an active mechanism for the selection of candidate constraints. The active fuzzy constrained clustering method is presented and evaluated on a ground truth image database to illustrate that the clustering can be significantly improved with few constraints. Recent works on constrained clustering are focused on evaluating the utility (i.e the potential interest) of a set of constraints [34, 35].

Kumar and Kummamuru [36] introduced another kind of knowledge through a clustering algorithm that uses supervision in terms of relative comparisons, e.g. x is closer to y than to z. Experimental studies on high-dimensional textual data sets demonstrated that the proposed algorithm achieved higher accuracy and is more robust than similar algorithms using pairwise constraints (*must-link* and *cannot-link*) for supervision.

Klein et al. [37] allowed instance-level constraints (i.e. must-link, cannot-link) to have space level inductive implications in order to improve the use of the constraints. This approach improved the results of the previously studied constrained KMEANS algorithms and generally requires less constraints to obtain the same accuracies.

Basu et al. [38] presented a pairwise constrained clustering framework as well as a new method for actively selecting informative pairwise constraints, to get improved clustering performance. Experimental and theoretical results confirm that this active querying of pairwise constraints significantly improves the accuracy of clustering, when given a relatively small amount of supervision.

Another way to integrate background knowledge is to use a small set of labeled samples. Basu et al. [39] used a set of samples to *seed* (i.e. to initialize) the clusters of the KMEANS algorithm. Two algorithms, Seeded KMEANS and Constrained KMEANS, are presented. In the first one, the labeled samples are used to initialize the clusters and the clusters are updated during the clustering process such as in the KMEANS algorithm. In the second one, the labeled samples used during the initialization stay in their assigned cluster, and only the unlabeled samples can change of cluster during the cluster affectation step of KMeans. The choice between these two approaches must be done according to the knowledge about noise in the dataset.

To tackle the problem of incorporating partial background knowledge into clustering, when the labeled samples have moderate overlapping features with the unlabeled data, Gao et al. [40] formulated a new approach as a constrained optimization problem. The authors introduced two learning algorithms to solve the problem, based on hard and fuzzy clustering methods. An empirical study shows that the proposed algorithms improve the quality of clustering results despite a limited number of labeled samples.

Basu et al. [41] also proposed a probabilistic model for semisupervised clustering, based on Hidden Markov Random Fields (HMRF), that provides a principled framework for incorporating supervision into prototype-based clustering. Experimental results on several text data sets demonstrate the advantages of this framework.

Another approach, called supervised clustering [42], uses the class information about the objects as an additional feature, to build clusters with a high class-based purity. The goal of supervised clustering is to identify class-uniform clusters having high probability densities. Supervised clustering is used to create summaries of datasets and for enhancing existing classification algorithms.

Different kinds of background knowledge are introduced by Pedrycz et al. [43], namely partial supervision, proximity-based guidance and uncertainty driven knowledge hints. The authors discuss about different ways of exploiting and effectively incorporating these background knowledge (known as *knowledge hints*) in the fuzzy c-means algorithm. In [44], Bouchachia and Pedrycz presented an extention of the fuzzy collaborative clustering which takes into account background knowledge through labeled objects. One of the advantages of the method is to take into account the classes splitted in several clusters. During the collaboration step, the method identify if a class correspond to various clusters and add or remove clusters according

to this information. More recentely, Pedrycz [45] presented some concepts and algorithms to collaborative and knowledge-based fuzzy clustering. The fuzzy c-means algorithm (FCM) was used as an operational model to explain the approach. Interesting linkages between information granularity, privacy and security of data in collaborative clustering were also discussed. The problem of data privacy when clustering multiple datasets was also recently discussed in [46]. An application of fuzzy clustering with partial knowledge to organize and classify digital images is also proposed in [27]. The author present an operational frameork of fuzzy clustering using the fuzzy c-means algorithm with an augemented objective function using background knowledge. Experiments are carried out on collections of images composed of 2000 images.

In this Section, we presented different works on using multiple clusterings: ensemble clustering (Section 2.1), multi-objective clustering (Section 2.2) and collaborative fuzzy-clustering (Section 2.3). The ensemble clustering approaches do not generally address the problem of the generation of the initial results, and the algorithms used to create the initial results are not used in the combination process. Consequently, ensemble clustering approaches introduce a new bias, relative to the objective function chosen when merging the different clusterings. The same problem appears with multi-objective clustering approaches where the optimized objectives are not the objectives of the methods used to generate the initial results. Collaborative fuzzy-clustering offers strong theorical basis on collaborative clustering but is only developped for fuzzy c-means which limit its use.

Finaly, we presented several work on the integration of background knowledge in clustering algorithm (Section 2.4). Different kinds of representation of the knowledge and different kinds of integration exist. However, every work claims that using background knowledge improves substancialy the results of clustering algorithms. The challenge task is to design methods able to leverage different kinds of knowledge. We prorpose such a method in the next section, where the collaborative clustering method Samarah is presented along with the different ways to integrate background knowledge into it.

3. Knowledge-guided collaborative clustering

As seen in Section 2, many techniques for combining clusterings exist. Unfortunately, only a few of them can handle the combination of clusterings having different numbers of clusters, because there is no obvious correspondence between the clusters of the different results.

Moreover, the proposed methods almost always aim to build a consensus among an ensemble of partitions or clusterings, without casting doubt on their quality. We think that a first step of collaboration of the clustering method before the consensus computation can help to obtain better results.

Thus, we propose a method consisting of a collaborative clustering process, based on an automatic and mutual refinement of several clustering results.

In this section, we first present the existing unsupervised collaborative clustering method called Samarah. Then, we present how we integrate knowledge into this collaborative clustering process.

3.1. Collaborative process overview

Computing a consensual result from clustering results having different numbers of clusters is a difficult task. This is mainly due to the lack of a trivial correspondence between the clusters of the different results. To address this particular problem, we present a framework where different clustering methods work together, in a collaborative way, to find an agreement about their proposals.

This collaborative process consists of an automatic and mutual refinement of the clustering results, until all the results have almost the same number of clusters, and all the clusters are statistically similar with a good internal quality. At the end of this process, as the results have comparable structures, it is possible to define a correspondence function between the clusters, and to apply a unifying technique, such as a voting method [47].

Before the description of the collaborative method, we introduce the correspondence function and the similarity measure used in the system.

There is no problem to associate classes of different supervised classifications, as a common set of class labels is given for all the classifications. Unfortunately, in the case of clustering, the results may not have a

same number of clusters, and no information is available about the correspondence between different clusters of different clusterings.

To address this problem, we have defined an intercluster correspondence function, which associates to each cluster from a result, a cluster from each of the other results. This cluster, in each result, is called the corresponding cluster.

Let $\breve{R} = \{\mathcal{R}^i\}_{1 \leq i \leq m}$ be the set of results given by the m different algorithms. Let $\{\mathcal{C}_k^i\}_{1 \leq k \leq n_i}$ be the clusters of the result $\mathcal{R}^i$.

The *corresponding cluster* $CC(\mathcal{C}_k^i, \mathcal{R}^j)$ of $\mathcal{C}_k^i$ in the result $\mathcal{R}^j$, $i \neq j$, is defined as

$$CC\left(\mathcal{C}_k^i, \mathcal{R}^j\right) = \arg \max_{\mathcal{C}_l^j \in \mathcal{R}^j} S\left(\mathcal{C}_k^i, \mathcal{C}_l^j\right) \tag{1}$$

where S is the intercluster similarity which evaluates the similarity between two clusters of two different results.

A large number of criteria exists to evaluate the similarity between two clustering results, like the Kappa index [48] or the Rand index [49], and more recently the Jaccard index[50] or the Fowlkes-Mallows index [51]. However, these criteria only give a global evaluation of the similarity between two partitions. In order to find the most similar cluster of one result in another one, we have to use an index based on the similarity between the two results **and** the similarity between each cluster of each result. To achieve this goal, we introduced the intercluster similarity between two clusters of two different results, which takes into account both aspects: the similarity through the confusion matrix between the two results (α in Eq.3), and the distribution of the cluster into the clusters of the second result through a distribution coefficient (ρ in Eq.3).

The confusion matrix (or matching matrix) is commonly used to compare two partitions or clustering results. The *confusion matrix* $\Omega^{i,j}$ between two results $\mathcal{R}^i$ and $\mathcal{R}^j$ is a $n_i \times n_j$ matrix defined by:

$$\Omega^{i,j} = \begin{pmatrix} \alpha_{1,1}^{i,j} & \dots & \alpha_{1,n_j}^{i,j} \\ & \vdots & \\ \alpha_{n_i,1}^{i,j} & \dots & \alpha_{n_i,n_j}^{i,j} \end{pmatrix} \text{ where } \alpha_{k,l}^{i,j} = \frac{\left|\mathcal{C}_k^i \bigcap \mathcal{C}_l^j\right|}{\left|\mathcal{C}_k^i\right|} \tag{2}$$

The *intercluster similarity* between two clusters $\mathcal{C}_k^i$ and $\mathcal{C}_l^j$ is evaluated by observing their intersection (Eq. 2) and by taking into account the distribution ρ (Eq. 4) of the cluster $\mathcal{C}_k^i$ in all the clusters of $\mathcal{R}^j$ as follows:

$$S\left(\mathcal{C}_k^i, \mathcal{C}_l^j\right) = \rho_k^{i,j} \alpha_{l,k}^{j,i} \tag{3}$$

where

$$\rho_k^{i,j} = \sum_{r=1}^{n_j} (\alpha_{k,r}^{i,j})^2 \tag{4}$$

The entire collaborative clustering process is broken down in three main phases:

1. *Initial clusterings* - Each method computes its result independently.
2. *Results refinement* - A phase of convergence of the results, which consists of conflict evaluations and resolutions, is iterated as long as the quality of the results and their similarity increase:
   (a) Detection of the conflicts, by evaluating the dissimilarities between couples of results;
   (b) Local resolution of some conflicts;
   (c) Global management of the local modifications in the global result (if they are relevant).
3. *Consensus computation* - The refined results are unified using a voting algorithm.

The entire algorithm of the method is detailed and explained in Algorithm 1.

Algorithm 1 Collaborative clustering process

Let $\breve{R} = \{\mathcal{R}^i\}_{1 \leq i \leq m}$ be the initial set of clusterings
Let $\breve{K} = \texttt{conflicts}(\breve{R})$ be the set of conflicts in $\breve{R}$ as defined in Eq.5
Let $\breve{R}^{\texttt{best}} := \breve{R}$ be the best temporary solution
Let $\breve{K}^{\texttt{best}} := \breve{K}$ be the conflicts of the best temporary solution
while $|\breve{K}| \geq 0$ **do**
  $\mathcal{K}_k^{i,j} := \arg\max_{\mathcal{K}_l^{r,s} \in \breve{K}} CI\left(\mathcal{K}_l^{r,s}\right)$
  $\breve{R} := \texttt{conflictResolution}\left(\breve{R}, \mathcal{K}_k^{i,j}\right)$ (Alg.2)
  if $\mathbf{\Gamma}(\breve{R}) > \mathbf{\Gamma}(\breve{R}^{\texttt{best}})$ **then**
    $\breve{R}^{\texttt{best}} := \breve{R}$
    $\breve{K}^{\texttt{best}} := \breve{K} := \texttt{conflicts}(\breve{R})$
    $bt := 0$
  else if $\breve{R}^{t+1} = \breve{R}^t$ **then**
    $\breve{K} := \breve{K} \setminus \mathcal{K}_k^{i,j}$
  else
    $bt := bt + 1$
    $\breve{K} := \breve{K} \setminus \mathcal{K}_k^{i,j}$
    if $bt > |\breve{K}|$ **then**
      $\breve{R} := \breve{R}^{\texttt{best}}$
      $\breve{K} := \breve{K}^{\texttt{best}} \setminus \mathcal{K}_k^{i,j}$
    end if
  end if
end while
consensus computation

3.1.1. Initial clusterings

Each clustering method computes a clustering of the data using its initial parameters: all data objects are grouped into different clusters. According to the base method selected, different parameters need to be set.

3.1.2. Results refinement

(a) Detection of the conflicts - The detection of the conflicts consists in seeking in $\breve{R}$ all the couples $\left(\mathcal{C}_k^i, \mathcal{R}^j\right)$, $i \neq j$, such as $S\left(\mathcal{C}_k^i, CC\left(\mathcal{C}_k^i, \mathcal{R}^j\right)\right) < 1$, which means that the cluster $\mathcal{C}_k^i$ can not be exactly found in the result $\mathcal{R}^j$.

$$\texttt{conflicts}(\breve{R}) = \left\{(\mathcal{C}_k^i, \mathcal{R}^j) : i \neq j, S\left(\mathcal{C}_k^i, CC\left(\mathcal{C}_k^i, \mathcal{R}^j\right)\right) < 1\right\} \tag{5}$$

Each conflict $\mathcal{K}_k^{i,j}$ is identified by one cluster $\mathcal{C}_k^i$ and one result $\mathcal{R}^j$. Its importance, $CI\left(\mathcal{K}_k^{i,j}\right)$, is computed according to the intercluster similarity:

$$CI\left(\mathcal{K}_k^{i,j}\right) = 1 - S\left(\mathcal{C}_k^i, CC\left(\mathcal{C}_k^i, \mathcal{R}^j\right)\right) \tag{6}$$

(b) Local resolution of some conflicts - The conflict resolution algorithm is detailed precisly in Algorithm 2.

The *most important* conflict (i.e. having the greatest conflict importance) is selected in the set of existing conflicts according to the conflict importance coefficient (Eq. 6).

The local resolution of a conflict $\mathcal{K}_k^{i,j}$ consists in applying an operator on each result involved in the conflict, $\mathcal{R}^i$ and $\mathcal{R}^j$, to try to improve their similarity. The operators which can be applied to a result are the following:

- *merging* of clusters: some clusters are merged together,
- *splitting* of a cluster into subclusters: a clustering is applied to the objects of a cluster to produce subclusters,
- *reclustering* of a group of objects: one cluster is removed and its objects are reclassified in all the other existing clusters.

The operator to apply is chosen according to the number of clusters involved in the conflict, i.e. the number of clusters such as $S(\mathcal{C}_k^i, \mathcal{C}_l^j) > p_{cr}$, where $0 \leq p_{cr} \leq 1$ is given by the user. The p_{cr} parameter represents the percentage above which the intersection between the two clusters is considered as significant. For example, $p_cr = 0.2$ means that if $\mathcal{C}_k^i \cup \mathcal{C}_l^j$ represents less than 20% of the objects of $\mathcal{C}_k^i$, $\mathcal{C}_l^j$ is not considered as a significant representative of $\mathcal{C}_k^i$.

However, the application of the two operators (each one on a different result) is not always relevant. Indeed, it does not always increase the similarity of the results involved in the processed conflict. Moreover, the iteration of the conflict resolutions step may lead to a trivial but consensual solution. For example, the clusterings can converge towards a solution where all the results have only one cluster, including all the objects to classify, or towards a solution where all the results have one cluster for each object. These two solutions are not relevant and must be avoided.

So, we defined a criterion γ, called *local similarity criterion*, to evaluate the similarity between two results and their quality. It is based on the intercluster similarity S (Eq. 3) and a quality criterion δ (detailed in Section 3.2, Eq. 23). The criterion δ evaluates the quality of the clustering (e.g. inertia, number of clusters, ...) to avoid that the method ends up with a trivial solution as those presented before:

$$\gamma^{i,j} = \frac{1}{2}\left(p_s.\left(\frac{1}{n_i}\sum_{k=1}^{n_i}\omega_k^{i,j} + \frac{1}{n_j}\sum_{k=1}^{n_j}\omega_k^{j,i}\right) + p_q.\left(\delta^i + \delta^j\right)\right) \tag{7}$$

where

$$\omega_k^{i,j} = S\left(\mathcal{C}_k^i, CC\left(\mathcal{C}_k^i, \mathcal{R}^j\right)\right) \tag{8}$$

and, p_q and p_s are given by the user ($p_q + p_s = 1$).

Let $\mathcal{R}^{i'}$ (resp. $\mathcal{R}^{j'}$) be the result $\mathcal{R}^i$ (resp. $\mathcal{R}^j$) after having applied the operators. The local similarity criterion is computed on each of the 4 couples of results: $(\mathcal{R}^i, \mathcal{R}^j), (\mathcal{R}^{i'}, \mathcal{R}^{j'}), (\mathcal{R}^{i'}, \mathcal{R}^j), (\mathcal{R}^i, \mathcal{R}^{j'})$. The best couple is accepted as the local solution of the conflict.

(c) Global management of the local modifications - After the resolutions of the local conflicts, a global application of the modifications proposed by the refinement step is decided if their application improve the quality of the global result. The *global agreement coefficient* $\mathbf{\Gamma}$ is evaluated according to all the local similarities between each couple of results as follows:

$$\mathbf{\Gamma} = \frac{1}{m}\sum_{i=1}^{m}\Gamma^i \tag{9}$$

where

$$\Gamma^i = \frac{1}{m-1}\sum_{\substack{j=1 \\ j\neq i}}^{m}\gamma^{i,j} \tag{10}$$

Three cases can occur:

Algorithm 2 Conflict resolution

Require: $\breve{R}$ the ensemble of clusterings
Require: $\mathcal{K}_k^{i,j}$ the conflict to solve
Ensure: $\breve{R}^\star = \texttt{conflictResolution}\left(\breve{R}, \mathcal{K}_k^{i,j}\right)$ the new ensemble after the resolution

let $\kappa = \{\mathcal{C}_l^j, \forall 1 \leq l \leq n_j : S(\mathcal{C}_k^i, \mathcal{C}_l^j) > p_{cr}\}$
if $|\kappa| > 1$ **then**
  $\mathcal{R}^{i'} = \mathcal{R}^i \setminus \{\mathcal{C}_k^i\} \cup \texttt{split}(\mathcal{C}_k^i, |\kappa|)$
  $\mathcal{R}^{j'} = \mathcal{R}^j \setminus \kappa \cup \texttt{merge}(\kappa, \mathcal{R}^j)$
else
  $\mathcal{R}^{i'} = \texttt{recluster}(\mathcal{R}^i \setminus \{\mathcal{C}_k^i\})$
end if
$\{\mathcal{R}^{i\star}, \mathcal{R}^{j\star}\} = \arg\max \gamma^{I,J}$ for $I \in \{i, i'\}, J \in \{j, j'\}$
$\breve{R}^\star = \breve{R} \setminus \{\mathcal{R}^i, \mathcal{R}^j\} \cup \{\mathcal{R}^{i\star}, \mathcal{R}^{j\star}\}$

- The resolution step gives a better solution than all previous ones (line 8). In this case, the best temporary solution is the one proposed by the conflict resolution step. As the global results have changed, the conflicts list is recomputed (line 10).
- The resolution step proposes the same solution which means that no operators application is relevant to solve this conflict (line 12). Then, the conflict is removed from the list and the algorithm iterates.
- If the solution proposed by the conflict resolution gives a worth global agreement coefficient, it is accepted to avoid to fall in a local maximum (line 14). But, if no conflict resolution enables to find a better solution (after having resolved the first half part of the conflicts list), all the results are reinitialized to the best temporary solution (line 17).

The process is iterated until some conflicts still remain in the conflicts list (line 5).

3.1.3. Consensus computation

After the refinement step, all the results tend to have the same number of clusters, which should be similar. Thus, in a final step, we use an original voting algorithm to compute a consensus result from the different results. This multi-view voting algorithm enables to combine in one unique result, many different clusterings that do not have necessarily the same number of clusters.

The basic idea is that for each object to cluster, each result $\mathcal{R}^i$ votes for the cluster it has found for this object, $\mathcal{C}_k^i$ for example, and for its corresponding cluster $CC(\mathcal{C}_k^i, \mathcal{R}^j)$ in each other result $\mathcal{R}^j$. The maximum of these values indicates the *best cluster* for the object, for example $\mathcal{C}_l^j$. It means that this object should be in the cluster $\mathcal{C}_l^j$ according to the opinion of the majority of the methods.

For each object p a voting matrix is computed as:

$$\mathcal{V}(p) = \left\{(v_1^i(p), \ldots, v_{n_i}^i(p)), 1 \leq i \leq m\right\} \tag{11}$$

where

$$v_k^i(p) = \sum_{j=1}^{m} vote(p, \mathcal{C}_k^i, \mathcal{R}^j) \tag{12}$$

and

$$vote(p, \mathcal{C}_k^i, \mathcal{R}^m) = \begin{cases} 1 & \text{if } (i = m \text{ and } p \in \mathcal{C}_k^i) \\ & \text{or } p \in CC(\mathcal{C}_k^i, \mathcal{R}^m) \\ 0 & \text{else} \end{cases} \tag{13}$$

The object p is then assigned to the cluster $\breve{\mathcal{V}}$, defined as:

$$\breve{\mathcal{V}}(p) = \arg\max_{\mathcal{C}_k^i} v_k^i(p) \tag{14}$$

3.2. Background knowledge integration

In this section we explain how we integrated background or domain knowledge into our collaborative clustering process. The aim is to make the method able to deal with two types of constraints: class label-based constraints and relationship between objects-based constraints (also called link-constraints). In class label-based constraints, a small subset of labeled samples is available. The goal is to try to have only one class represented in each cluster. In link-constraints, the goal is to respect the constraints provided on the objects.

Firstly (Section 3.2.1) , we will see some examples of background knowledge integration present in the literature and then (Section 3.2.2) we will see how we integrated it into the SAMARAH method.

3.2.1. Examples of knowledge integration

In the literature, different methods have already been proposed to take into account background knowledge. For example, to integrate link-constraints in the algorithm Pairwise Constrained K-means, Basu et al. [41] defines the following objective function:

$$\begin{aligned} obj_{pckm} = \sum_{j=1}^{n_k^i} \text{dispersion}(\mathcal{C}_j^i) \quad & + \textstyle\sum_{(x_i,x_j)\in\mathbb{M}} [l_i \neq l_j] \\ & + \textstyle\sum_{(x_i,x_j)\in\mathbb{C}} [l_i = l_j] \end{aligned} \tag{15}$$

where $\mathbb{M}$ is the set of must-link constraints and $\mathbb{C}$ is the set of cannot-link constraints. This objective function includes a dispersion measure computed as the classical mean squared error (first term), but also two other components (second and third terms) reflecting the agreement according to the sets of available constraints ($\mathbb{M}$ and $\mathbb{C}$). The functions $[l_i \neq l_j]$ and $[l_i = l_j]$ return 1 if the constraint between the couple of objects (x_i, y_i) is respected, and 0 else.

An approach is proposed by Demiriz et al. [24] to integrate class label-based constraints. This method uses a genetic algorithm to minimize an objective function, which is a geometric mean between cluster dispersion and cluster impurity according to available samples. It is defined as:

$$obj_{gen} = \sum_{j=1}^{n_k^i} \left(\alpha \times \text{dispersion}(\mathcal{C}_j^i) + \beta \times \text{impurity}(\mathcal{C}_j^i)\right) \tag{16}$$

where the cluster dispersion is usually the mean squared error, and the cluster impurity is a measure of the impurity of a cluster according to its composition of available labeled samples. This impurity measure is low if all the known samples of a cluster are from the same class. On the contrary, the value increases as the cluster contains objects from various classes (and the cluster will be considered as impure). This impurity is evaluated thanks to the Gini index:

$$\text{gini}(\mathcal{C}_k^i) = 1 - \sum_{l=0}^{n_c} \left(\frac{P_{kl}}{n_k^i}\right)^2 \tag{17}$$

where P_{kl} is the number of objects belonging to the l^{th} class in the cluster $\mathcal{C}_k^i$, and n_k^i is the number of objects in the cluster $\mathcal{C}_k^i$. The weights α and β in Eq. 16 allow the user to choose if he prefers to give more or less importance to the available knowledge.

In supervised clustering [42], a similar idea is used. A penalty measure is added to the impurity measure to deal with the case of various number of clusters in the results and avoid results with very high or very low number of clusters:

$$obj_{sc} = \sum_{j=1}^{n_k^i} \left(\alpha \times \text{impurity}(\mathcal{C}_j^i) + \beta \times \text{penalty}(\mathcal{C}_j^i)\right) \tag{18}$$

where the impurity measure is defined by:

$$\text{impurity}(\mathcal{C}_k^i) = \frac{1}{n} \sum_{\substack{l=0 \\ l \neq cmax}}^{n_c} P_{kl} \tag{19}$$

where P_{kl} is the number of objects belonging to the l^{th} class in the cluster $\mathcal{C}_k^i$, n_k^i is the number of objects in the cluster $\mathcal{C}_k^i$, and $cmax$ is the most represented class in the cluster $\mathcal{C}_k^i$

$$cmax = \arg\max_l(P_{kl}) \tag{20}$$

The penalty is defined as

$$\text{penalty}(\mathcal{C}_k) = \begin{cases} \sqrt{\frac{n_k - n_c}{n}} & \text{if } n_k \geq n_c \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

where n_c is the number of classes in the known samples. Here again, the weights α and β in Eq. 18 are chosen by the user.

3.2.2. Knowledge integration in the SAMARAH *method*

In the SAMARAH method, during the refinement step, the local similarity criterion $\gamma^{i,j}$ (Eq. 7) is used to evaluate if the modifications of a couple of results is relevant. This criterion includes a quality criterion δ^i which represents the quality of the result $\mathcal{R}^i$. This criterion is used to balance the refinement step between the similarity and the quality of the expected results. It is computed for two aspects of the results: the internal and external qualities. The internal evaluation consists in evaluating the quality of the result through a unsupervised measure. The external evaluation consists in evaluating the quality of a result thanks to external knowledge, such as an estimation of the number of clusters, some labeled samples or some constraints.

The previous version of SAMARAH already includes internal knowledge but only includes an estimation of the number of clusters as external knowledge. To take into account more external knowledge, we have extended the quality criterion by integrating an evaluation of the agreement of the results with different kinds of constraints as follows:

$$\delta^i = \sum_{c=1}^{N_c} q_c(\mathcal{R}^i) \times p_c \tag{22}$$

where N_c is the number of constraints to respect, q_c is the criterion used to evaluate the result according to the c-th constraint ($q_c(.) \in [0, 1]$) and p_c is the relative importance given by the user to the c-th constraint ($p_1 + p_2 + \cdots + p_{N_c} = 1$). By default, each constraint is given with a weight of $\frac{1}{N_c}$.

Thus, any constraint can be integrated in the process if it can be defined as a function taking its values in $[0, 1]$. We described below some frequently encountered constraints that can be used.

Cluster quality-based constraints - These constraints are based on the intrinsic quality of the clusters such as inertia or predictivity and also take into account the number of clusters. Indeed criterion such as inertia or compacity need to be balanced with an evaluation of the number of clusters. An example

of a criterion which includes quality of the clusters and the number of clusters is given below:

$$q_{qb}(\mathcal{R}^i) = \frac{p^i}{n^i} \sum_{k=1}^{n^i} \tau_k^i \tag{23}$$

where n^i is the number of clusters of $\mathcal{R}^i$, τ_k^i defines the internal quality of the k-th cluster and p^i is the external quality of the result. The internal quality of the k-th cluster is given as:

$$\tau_k^i = \begin{cases} 0 \text{ if } \frac{1}{n_k^i} \sum_{l=1}^{n_k^i} \frac{d(x_{k,l}^i, g_k^i)}{d(x_{k,l}^i, g^i)} > 1 \\ 1 - \frac{1}{n_k^i} \sum_{l=1}^{n_k^i} \frac{d(x_{k,l}^i, g_k^i)}{d(x_{k,l}^i, g^i)} \text{ else} \end{cases} \tag{24}$$

where n_k^i is the cardinal of $\mathcal{C}_k^i$, g_k^i is the gravity center of $\mathcal{C}_k^i$, g^i is the gravity center of another cluster, the closest from $x_{k,l}^i$ and d is the distance function. The measure is computed on each cluster to evaluate the overall quality of the clustering result. To take into account the number of clusters n^i, the criterion p^i is defined as:

$$p^i = \frac{n_{\text{sup}} - n_{\text{inf}}}{|n_i - n_{\text{inf}}| + |n_{\text{sup}} - n_i|} \tag{25}$$

where $[n_{\text{inf}}, n_{\text{sup}}]$ is given by the user, as the range of expected number of clusters.

Class label-based constraints - These constraints correspond to the case where a few sets of labeled samples are available. To evaluate the agreement between results and such constraints, we can use any index which enables us to evaluate the similarity between a clustering and a labeled classification (where all the classes are known, and each object belongs to one of these classes). In our case, we only compare results with a given partial partition $\mathbb{R}$ which represents the known labeled objects. In the implementation of the SAMARAH method, we mainly used the Rand index [49] and another index known as WG agreement index [52]. Information theoric measures could also be used [53].

The Rand index is a measure of the similarity between two data partitions defined by:

$$Rand(\mathcal{R}^i, \mathbb{R}) = \frac{a+b}{\binom{n}{2}} \tag{26}$$

where n is the number of objects to classify, a is the number of pairs of objects which are in the same cluster in $\mathcal{R}^i$ and in the known result, and b is the number of pairs of objects which are not in the same cluster in the proposed result $\mathcal{R}^i$ and in the known result $\mathcal{R}^j$. The sum of these two measurements (a and b) can been seen as the number of times that the two partitions are in agreement. This index takes values in $[0, 1]$: 1 indicates that the two partitions are identical. The defined constraint is:

$$q_{rand}(\mathcal{R}^i) = Rand(\mathcal{R}^i, \mathbb{R}) \tag{27}$$

The WG agreement index is defined by

$$WG(\mathcal{R}^i, \mathbb{R}) = \frac{1}{n} \sum_{k=1}^{n_i} S\left(C_k^i, \mathcal{R}^j\right) \left|C_k^i\right| \tag{28}$$

where n is the number of objects to classify and $\mathcal{R}^j$ is the reference partition (e.g. labeled classification, another clustering, etc.). This index takes values in $[0, 1]$: 1 indicates that all the objects in the clustering $\mathcal{R}^i$ are well classified according to the class label of the objects in $\mathcal{R}^j$. The defined constraint is:

$$q_{wg}(\mathcal{R}^i) = WG(\mathcal{R}^i, \mathbb{R}) \tag{29}$$

Link-based constraints - These constraints correspond to the case where knowledge is expressed as

must-link and *cannot-link* constraints between objects (see [30, 31]). In this case, the ratio of respected constraints against violated constraints can easily be computed as

$$q_{link}(\mathcal{R}^i) = \frac{1}{n_r} \sum_{j=1}^{n_r} v(\mathcal{R}^i, l_j) \tag{30}$$

where n_r is the number of constraints between the objects, l_j is a *must-link* or *cannot-link* constraint and $v(\mathcal{R}^i, l_j) = 1$ if $\mathcal{R}^i$ respects the constraint l_j, 0 otherwise.

Note that such constraints can be extracted from class-label constraints. For example, a must-link constraint could be created for all the couples of objects belonging to the same cluster, and a cannot-link constraint could be created for all the couples of objects belonging to different clusters.

3.3. Example

In this section, we present a simple example on a 2D dataset. The aim is to illustrate how the different kinds of knowledge presented in the previous section can improve the collaboration between the clustering methods. We used the 9-Diamonds dataset (Fig. 1) from [54] which is available for download on the website of the machine learning group of the University of Houston [1].

We used the KMeans algorithm to cluster this dataset in various number of clusters ranging from 2 to 18 (9 being the actual number of cluster). Then we considered couple of clustering results and we computed for each couple a value of agreement between the two clustering results. The goal is illustrate the search space of the different possible solutions (i.e. couple of clustering results). In the followings figures (Fig. 2, 3, 4, 5) the higher the values are, the higher the agreement is. We illustrated here how different kinds of knowlege can modify the shape of the search-space and consequently, help the collaboration.

The Fig. 2 presents the local agreement using only the similarity of the result (7). One can see that using only the similarity creates several local minima in this search space as the results with a low number of clusters are strongly similar. To reduce this problem, the knowledge of the range of expected number of clusters can be used.

The Fig. 3 presents the local agreement using the similarity along with some knowledge about the number of clusters (25) (i.e set here to $[7; 11]$). One can see that the search space leverages this information and that the value for the number of cluster outsite the range are strongly reduced.

If some labeled patterns are available, a measure of quality of the clustering can be added to the evaluation to guide the collaboration. This is illustrated on Fig. 4 where the WG index (29) is computed assuming 5% of labeled objects. On can see that this information improves substantialy the shape of the search space.

Finaly, different kinds of knowledge can be used together as in Fig. 5 where the range of expected number of clusters along with the knowlege of some labeled objets are used. The resulting search-space is cleary easier to explore and contains less local minimas, the optimal solution (9 clusters) being strongly highlighted.

4. Evaluation of the proposed method

4.1. Protocol of experiments

In this section, we present two evaluations of the method proposed in this paper. The aim of these experiments is twofold. Firstly, we want to show the relevance of collaborative clustering to improve the quality of a set of clustering results, thanks to our collaborative process. Secondly, we want to show the interest in integrating background knowledge in this collaboration, to produce even better results.

Two kinds of experiment have been carried out:

[1] http://www.tlc2.uh.edu/dmmlg/

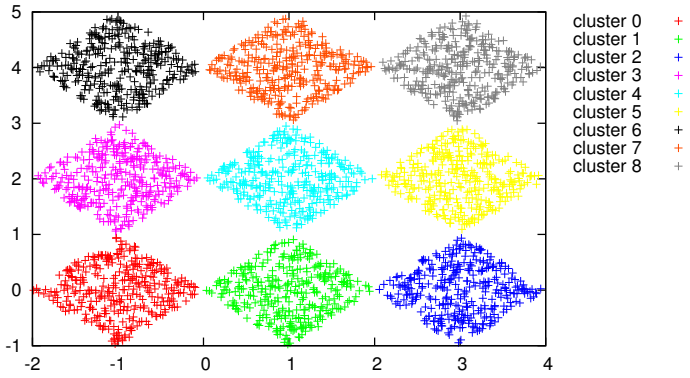
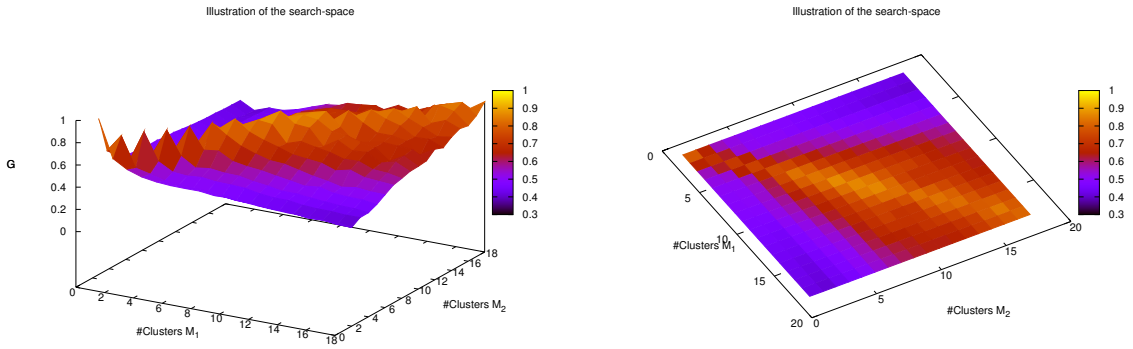
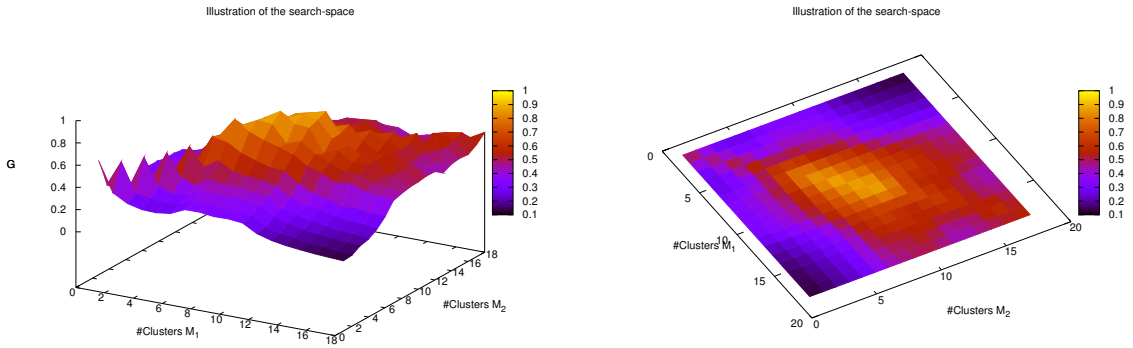


Figure 1: 9-Diamonds dataset



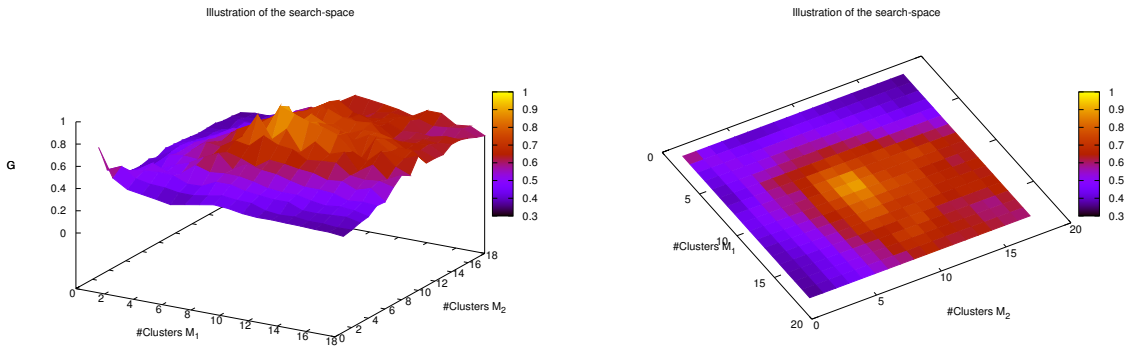
(a) I (b) II

Figure 2: The search-space using only the similarity.



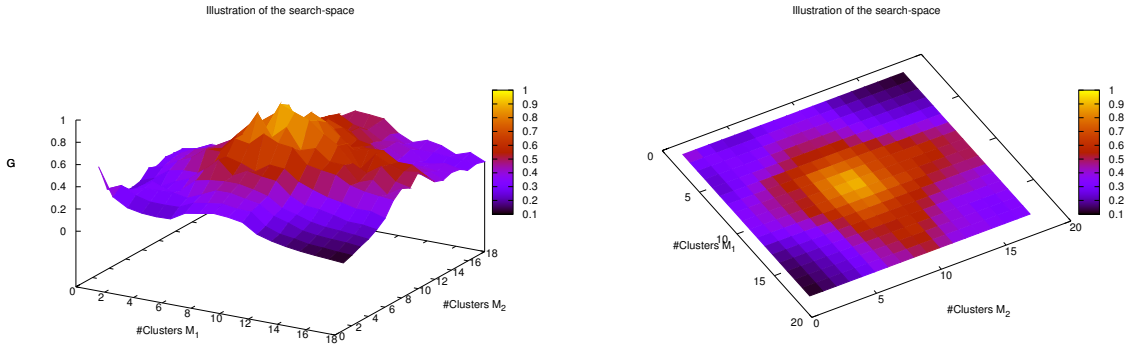
(a) I (b) II

Figure 3: The search-space using the similarity and the range of expected number of clusters.



(a) I (b) II

Figure 4: The search-space using the knowlege of some labeled patterns.



(a) I (b) II

Figure 5: The search-space using the range of expected number of clusters and the knowledge of labeled patterns.

1. The first one consisted in the evaluation of the quality of a set of clustering results, first, without collaboration, then with collaboration, and finally, with collaboration integrating background knowledge. Different classical quality indexes were used to evaluate the quality of these sets of clustering results. The details of these experiments are described in Section 4.3.
2. The second one consisted in the evaluation using the Cascade Evaluation approach [7]. This approach is based on the enrichment of a set of datasets by the clustering results, and the use of a supervised method to evaluate the interest of adding such new information to the datasets. The details of these experiments are described in Section 4.4.

For all the experiments with SAMARAH, we used the KMEANS algorithm [55] as base clustering method (Section 3.1). Five methods were randomly initialized with a number of clusters randomly picked in $[2; 10]$. The refinement step was set up to find results with a number of clusters in $[2; 10]$ (i.e. $n_{\text{inf}} = 2, n_{\text{sup}} = 10$ in Eq. 25).

To evaluate benefits of the background knowledge integration in the refinement step as presented in Section 3.2, we randomly picked up 10% of the datasets as a subset of available samples. This subset was

used to drive the refinement step through the quality evaluation of the results. The datasets used in both the experiments are presented in the next section.

4.2. Data sets

Seven different datasets from the UCI machine learning repository [56] were used in the experiments:

1. Iris data set, which contains 3 classes of 50 instances, where each class refers to a type of iris plant;
2. Wine database, which contains 3 classes of wines, characterized by 13 chemicals attributes (178 instances);
3. Ionosphere database, which consists in 351 information about 16 high-frequency antennas classified into 2 classes;
4. Pima Indians Diabetes database, referred as Pima, which consists in 768 patients discriminated into 2 classes according to World Health Organization criteria;
5. Sonar, which has been used to learn to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock (2 classes, 307 instances), using 60 real attributes (each attribute represents the energy within a particular frequency band);
6. Vehicle data set, which contains 4 classes of 946 vehicles to classify given a set of 18 features extracted from their silhouette at different angles;
7. Segment database, composed of 2500 instances of 3×3 regions extracted from 7 images of texture; each region is characterized by 19 attributes.

4.3. Comparison using quality indexes

In this first experiment, we evaluated the quality of the sets of clustering results, without collaboration (referred as $\neg col$), refined with collaboration (referred as col), and refined with collaboration using background knowledge (referred as $kcol$). We used six different quality indexes to evaluate the quality of each set:

- the Rand index [49]
- the Jaccard index [50]
- the Falks-Mallow index [51]
- the Wemmert-Gançarski index [52]
- the F-Measure index [51]
- the Kappa index [48]

For each set of clusterings ($\neg col$, col and $kcol$), the mean of the quality of each element of the set was computed, to define the quality of the set. We carried out 100 runs computing at each run the set of initial clusterings (with the parameters defined in Section 4.1) and the refined set through the collaborative process and through the collaborative process using background knowledge. The evaluation of the results are given in Table 1, where the values correspond to the means, the standard deviations and the maximum accurary of the results on the 100 runs. On Fig. 6, we illustrate the results for the Rand index for all the experiments and all the datasets.

As one can see, the quality of the refined set (col) is almost always better than the quality of the unrefined set ($\neg col$). Indeed, for each dataset the quality is better according to at least 5 out of the 6 quality indexes. Furthermore, the quality of the sets provided by the collaboration process using background knowledge ($kcol$) gives even better results.

All these results show, firstly, that the refinement step of our collaborative clustering method improve the quality of the results, and secondly, that the use of background knowledge in the collaboration helps the process to produce better results.

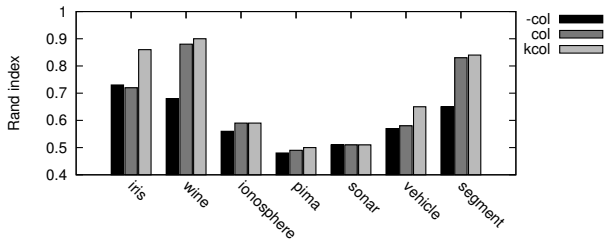


Figure 6: Rand quality index for the three experiments ($\neg col$, col and $kcol$) for all datasets

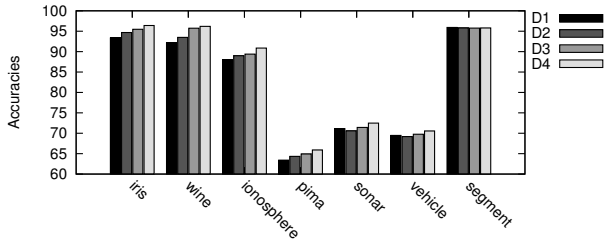


Figure 7: Accuracies of cascade evaluation for all datasets

4.4. Comparison using cascade evaluation

The cascade evaluation [7] is a new approach to evaluate the quality and the interest of clustering results. The method is based on the enrichment of a set of datasets by the results of clustering, and the use of a supervised method to evaluate the interest of adding such new information to the datasets.

The method consists in evaluating and comparing the result of a supervised classifier when it is helped or not by the information issued from a clustering. If the result of the classifier is improved by the information added by the clustering, the authors assume that the clustering embeds a meaningful information. Furthermore, different clustering results can be compared, the one improving the most the result according to the classifier accuracy is consequently the one embedding the most interesting information.

We used this paradigm to conduct a cascade evaluation of the collaborative clustering. The aim of this evaluation is to show the relevance of the refinement step presented above. We want to see if the refinement improves the different results through the collaborative step. We are consequently interested to show that the set of refined results contains a more accurate knowledge compared to the initial unrefined set of results.

To evaluate the benefits of using the information provided by our method in supervised algorithms, we created different datasets from the initial one, and then we classified them with a supervised algorithm. Let D be the initial dataset to classify : $D = \{o_i\}_{1 \leq i \leq l}$ where the object o_i is characterized by the m attributes $A_1(o_i), \ldots, A_m(o_i)$. Let $\mathcal{C}_j(o_i)$ be the cluster of the object o_i in the j^{th} *initial* clustering result. Let $\mathcal{R}_j(o_i)$ be the cluster of the object o_i in the j^{th} *refined* clustering result. Let $\mathcal{K}_j(o_i)$ be the cluster of the object o_i in the j^{th} *refined with knowledge* clustering result. From each initial dataset D, three datasets were created to integrate knowledge provided from the different clusterings:

- $D^1 = \{O_i^1\}_{1 \leq i \leq l}$
  where $O_i^1 = (A_1(o_i), \ldots, A_m(o_i), \mathcal{C}_1(o_i), \ldots, \mathcal{C}_n(o_i))$

- $D^2 = \{O_i^2\}_{1 \leq i \leq l}$
  where $O_i^2 = (A_1(o_i), \ldots, A_m(o_i), \mathcal{R}_1(o_i), \ldots, \mathcal{R}_n(o_i))$

- $D^3 = \{O_i^3\}_{1 \leq i \leq l}$
  where $O_i^3 = (A_1(o_i), \ldots, A_m(o_i), \mathcal{K}_1(o_i), \ldots, \mathcal{K}_n(o_i))$

Table 1: Quality of the results of the different sets of results $\neg col$, col and $kcol$.

		Rand	Jacc	FM	WG	F-M	K
iris	($\neg col$)	0.73 (±0.03) ◇ 0.78	0.44 (±0.02) ◇ 0.46	0.64 (±0.02) ◇ 0.65	0.54 (±0.03) ◇ 0.59	0.60 (±0.01) ◇ 0.62	0.43 (±0.01) ◇ 0.45
	(col)	0.85 (±0.00) ◇ 0.87	0.64 (±0.01) ◇ 0.67	0.78 (±0.00) ◇ 0.80	0.69 (±0.03) ◇ 0.72	0.78 (±0.01) ◇ 0.80	0.67 (±0.01) ◇ 0.70
	($kcol$)	**0.86 (±0.01) ◇ 0.87**	**0.65 (±0.02) ◇ 0.68**	**0.79 (±0.01) ◇ 0.81**	**0.71 (±0.03) ◇ 0.74**	**0.79 (±0.02) ◇ 0.81**	**0.69 (±0.02) ◇ 0.72**
wine	($\neg col$)	0.68 (±0.09) ◇ 0.76	0.46 (±0.05) ◇ 0.55	0.66 (±0.03) ◇ 0.72	0.59 (±0.04) ◇ 0.66	0.62 (±0.04) ◇ 0.69	0.39 (±0.10) ◇ 0.52
	(col)	0.88 (±0.04) ◇ 0.94	0.71 (±0.09) ◇ 0.83	0.83 (±0.06) ◇ 0.91	0.75 (±0.06) ◇ 0.83	0.83 (±0.06) ◇ 0.91	0.74 (±0.10) ◇ 0.86
	($kcol$)	**0.90 (±0.02) ◇ 0.94**	**0.76 (±0.05) ◇ 0.83**	**0.86 (±0.03) ◇ 0.91**	**0.79 (±0.07) ◇ 0.87**	**0.86 (±0.03) ◇ 0.91**	**0.78 (±0.04) ◇ 0.86**
ionosphere	($\neg col$)	0.56 (±0.01) ◇ 0.58	0.34 (±0.03) ◇ 0.38	0.53 (±0.03) ◇ 0.57	**0.30 (±0.03) ◇ 0.32**	0.50 (±0.04) ◇ 0.55	0.14 (±0.02) ◇ 0.16
	(col)	**0.59 (±0.03) ◇ 0.61**	0.34 (±0.08) ◇ 0.41	0.53 (±0.07) ◇ 0.59	0.21 (±0.05) ◇ 0.28	0.50 (±0.09) ◇ 0.58	0.20 (±0.04) ◇ 0.23
	($kcol$)	**0.59 (±0.01) ◇ 0.60**	**0.37 (±0.03) ◇ 0.39**	**0.55 (±0.02) ◇ 0.57**	0.22 (±0.03) ◇ 0.26	**0.54 (±0.03) ◇ 0.56**	**0.21 (±0.02) ◇ 0.22**
pima	($\neg col$)	0.48 (±0.00) ◇ 0.49	0.22 (±0.02) ◇ 0.25	0.38 (±0.02) ◇ 0.41	0.17 (±0.01) ◇ 0.18	0.35 (±0.03) ◇ 0.39	**0.40 (±0.01) ◇ 0.41**
	(col)	0.49 (±0.00) ◇ 0.50	**0.29 (±0.00) ◇ 0.30**	**0.46 (±0.00) ◇ 0.46**	**0.20 (±0.01) ◇ 0.21**	**0.45 (±0.01) ◇ 0.46**	0.37 (±0.00) ◇ 0.38
	($kcol$)	**0.50 (±0.00) ◇ 0.50**	**0.29 (±0.01) ◇ 0.30**	**0.46 (±0.01) ◇ 0.47**	**0.20 (±0.01) ◇ 0.20**	**0.45 (±0.01) ◇ 0.46**	0.37 (±0.00) ◇ 0.38
sonar	($\neg col$)	**0.51 (±0.01) ◇ 0.52**	0.25 (±0.04) ◇ 0.27	0.41 (±0.04) ◇ 0.47	**0.22 (±0.04) ◇ 0.29**	0.39 (±0.05) ◇ 0.45	**0.02 (±0.01) ◇ 0.04**
	(col)	**0.51 (±0.00) ◇ 0.51**	0.26 (±0.02) ◇ 0.28	0.42 (±0.02) ◇ 0.44	0.20 (±0.01) ◇ 0.021	0.41 (±0.02) ◇ 0.43	0.01 (±0.01) ◇ 0.0.02
	($kcol$)	**0.51 (±0.01) ◇ 0.52**	**0.31 (±0.05) ◇ 0.37**	**0.47 (±0.06) ◇ 0.54**	0.16 (±0.03) ◇ 0.18	**0.47 (±0.06) ◇ 0.54**	0.01 (±0.01) ◇ 0.04
vehicle	($\neg col$)	0.57 (±0.06) ◇ 0.65	0.21 (±0.01) ◇ 0.22	0.37 (±0.02) ◇ 0.40	0.25 (±0.05) ◇ 0.31	0.35 (±0.01) ◇ 0.36	0.50 (±0.10) ◇ 0.62
	(col)	0.58 (±0.07) ◇ 0.69	**0.22 (±0.01) ◇ 0.24**	**0.38 (±0.02) ◇ 0.40**	**0.25 (±0.06) ◇ 0.22**	0.35 (±0.01) ◇ 0.39	0.51 (±0.12) ◇ 0.68
	($kcol$)	**0.65 (±0.02) ◇ 0.66**	**0.22 (±0.01) ◇ 0.24**	0.37 (±0.01) ◇ 0.39	0.16 (±0.01) ◇ 0.17	**0.37 (±0.01) ◇ 0.39**	**0.63 (±0.02) ◇ 0.64**
segment	($\neg col$)	0.65 (±0.06) ◇ 0.71	0.30 (±0.03) ◇ 0.34	0.49 (±0.03) ◇ 0.53	0.46 (±0.03) ◇ 0.49	0.45 (±0.04) ◇ 0.49	0.61 (±0.07) ◇ 0.68
	(col)	0.83 (±0.02) ◇ 0.85	0.38 (±0.03) ◇ 0.41	0.56 (±0.03) ◇ 0.59	0.48 (±0.05) ◇ 0.54	0.55 (±0.03) ◇ 0.58	**0.83 (±0.03) ◇ 0.84**
	($kcol$)	**0.84 (±0.03) ◇ 0.87**	**0.39 (±0.03) ◇ 0.42**	**0.58 (±0.03) ◇ 0.60**	**0.52 (±0.07) ◇ 0.60**	**0.56 (±0.03) ◇ 0.60**	**0.83 (±0.03) ◇ 0.87**

Table 2: Evaluation of the integration of background knowledge using cascade evaluation

	D	D^1	D^2	D^3
iris	93.33%	94.67% (±0.47) ◇ 94.67	95.47% (±0.30) ◇ 96.00	**96.40% (±0.60) ◇ 96.67**
wine	92.13%	93.48% (±1.52) ◇ 95.51	95.73% (±0.50) ◇ 96.07	**96.18% (±0.73) ◇ 97.19**
ionosphere	88.03%	89.00% (±1.25) ◇ 90.60	89.40% (±0.93) ◇ 90.88	**90.94% (±0.31) ◇ 91.17**
pima	63.41%	64.35% (±0.57) ◇ 65.23	64.92% (±0.76) ◇ 65.89	**65.89% (±0.95) ◇ 67.45**
sonar	71.15%	70.58% (±0.79) ◇ 71.63	71.44% (±0.87) ◇ 71.63	**72.50% (±0.86) ◇ 74.04**
vehicle	69.47%	69.17% (±0.87) ◇ 70.35	69.77% (±0.91) ◇ 71.23	**70.55% (±0.82) ◇ 71.61**
segment	**95.93%**	95.79% (±0.17) ◇ 96.02	95.77% (±0.15) ◇ 95.97	95.79% (±0.14) ◇ 95.97

As supervised algorithm, we chose the tree-based classifier C4.5 [57] for his ability to handle numeric and categorical attributes, and we made 100 runs of 10-fold cross validations, each on the three versions of each dataset.

The results of this evaluation are presented in Table 2 where the values are the average values, the standard deviations and the maximum values of the accuracy for the four versions of each dataset (i.e. the normal dataset D, the one embedded with the clustering D^1, the one embedded with the refined clustering D^2, and the one embedded with the refined clustering integrating background knowledge D^3). The histograms of the accuracies are presented on Fig. 7. One can see that the datasets embedded by the refined clusterings gives the best results on 6 of the 7 datasets. One can notice that the refinement step degrades the results without clustering only when the initial clustering results also degrade the result. This can be explained by the lack of concordance between the class of the objects and their distribution in the data space. Consequently, adding the clustering information just added noise to the dataset.

Furthermore, one can observe an increase of the stability, as the standard deviation significantly decreases, when the refined results are used instead of the initial ones. The results refined using background knowledge are better in means than the results obtained without it. However the results are less stable (higher standard deviation) on the half of the datasets. This can be explained by the high degree of randomness in the selection of the samples used as background knowledge. If these samples are well distributed on the data space and among the different clusters, they will carry a better information and will make helping the collaboration easier. We assume that this issue can be solved if the samples are actually provided by the expert itself (and not selected randomly). The expert should be able to provide high quality examples. An active learning approach could also be used where the expert could, during the collaboration, gives information about the clusters involved in strong conflict.

5. Conclusion

In many clustering problems, the user is able to provide some background knowledge which can be used to guide the algorithm to obtain more accurate results. Moreover, it has been accepted that ensemble clustering algorithms give more robust results to such problems. Unfortunately, no ensemble clustering method that includes information given by the user has been defined yet.

In this article, we have presented a new method of collaborative clustering, that integrates and benefits from background knowledge on the given problem. The user can express the knowledge through different

kinds of constraints. To illustrate this, we have proposed a formalization of three main types of constraints: cluster quality, class label and link-based constraints. Then, we presented how this information can be used during the collaboration step, to guide the different methods in their search for a better solution.

Finally, we have shown by different experiments that the collaboration between the clustering methods provides better results than the single classical method, and that introducing knowledge to control the collaboration gives even more accurate results.

References

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

[2] Rauber, A., Pampalk, E., Paralic, J.. Empirical evaluation of clustering algorithms. Journal of Information and Organizational Sciences (JIOS) 2000;24(2):195–209.

[3] Berkhin, P.. Survey of clustering data mining techniques. Tech. Rep.; Accrue Software; San Jose, CA; 2002.

[4] Xu, R., Wunsch, D.. Survey of clustering algorithms. IEEE Transactions on Neural Networks 2005;16(3):645–678.

[5] Li, C., Biswas, G.. Unsupervised learning with mixed numeric and nominal data. IEEE Transactions on Knowledge and Data Engineering 2002;14(4):673–690.

[6] Kittler, J.. On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 1998;20(3):226–239.

[7] Candillier, L., Tellier, I., Torre, F., Bousquet, O.. Cascade evaluation of clustering algorithms. In: European Conference on Machine Learning; vol. 4212/2006. Springer Berlin / Heidelberg; 2006, p. 574–581.

[8] Strehl, A., Ghosh, J.. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. Journal on Machine Learning Research 2002;3:583–617.

[9] Fred, A., Jain, A.. Combining multiple clusterings using evidence accumulation. IEEE Transactions on Pattern Analysis and Machine Intelligence 2005;27(6):835–850. doi:10.1109/TPAMI.2005.113.

[10] Topchy, A.P., Jain, A.K., Punch, W.F.. Combining multiple weak clusterings. In: International Conference on Data Mining. IEEE Computer Society; 2003, p. 331–338.

[11] Topchy, A., Jain, A., Punch, W.. Clustering ensembles: models of consensus and weak partitions. IEEE Transactions on Pattern Analysis and Machine Intelligence 2005;27(12):1866–1881. doi:10.1109/TPAMI.2005.237.

[12] Hadjitodorov, S.T., Kuncheva, L.I.. Selecting diversifying heuristics for cluster ensembles. In: International Workshop on Multiple Classifier Systems; vol. 4472. 2007, p. 200–209.

[13] Ayad, H., Kamel, M.S.. Cumulative voting consensus method for partitions with variable number of clusters. IEEE Transactions on Pattern Analysis and Machine Intelligence 2008;30(1):160–173.

[14] Nguyen, N., Caruana, R.. Consensus clusterings. In: International Conference on Data Mining. IEEE Computer Society; 2007, p. 607–612.

[15] Long, B., Zhang, Z., Yu, P.S.. Combining multiple clusterings by soft correspondence. In: International Conference on Data Mining. IEEE Computer Society; 2005, p. 282–289.

[16] Hu, T., Yu, Y., Xiong, J., Sung, S.Y.. Maximum likelihood combination of multiple clusterings. Pattern Recognition Letters 2006;27(13):1457–1464.

[17] Pedrycz, W., Hirota, K.. A consensus-driven fuzzy clustering. Pattern Recognition Letters 2008;29(9):1333–1343.

[18] Handl, J., Knowles, J.. An evolutionary approach to multiobjective clustering. IEEE Transactions on Evolutionary Computation 2007;11(1):56–76.

[19] Konak, A., Coit, D., Smith, A.. Multi-objective optimization using genetic algorithms: A tutorial. Reliability Engineering & System Safety 2006;91(9):992–1007.

[20] Handl, J., Knowles, J.D.. On semi-supervised clustering via multiobjective optimization. In: Genetic and Evolutionary Computation Conference. 2006, p. 1465–1472.

[21] Faceli, K., de Carvalho, A., de Souto, M.. Multi-objective clustering ensemble. International Conference on Hybrid Intelligent Systems 2006;:51–51.

[22] Law, M., Topchy, A., Jain, A.. Multiobjective data clustering. In: IEEE Conference on Computer Vision and Pattern Recognition; vol. 2. 2004, p. 424–430.

[23] Faceli, K., de Carvalho, A.C.P.F., de Souto, M.C.P.. Multi-objective clustering ensemble with prior knowledge. vol. 4643. Springer; 2007, p. 34–45.

[24] Demiriz, A., Bennett, K., Embrechts, M.. Semi-supervised clustering using genetic algorithms. 1999, p. 809–814.

[25] Pedrycz, W.. Collaborative fuzzy clustering. Pattern Recognition Letters 2002;23:1675–1686.

[26] Pedrycz, W., Rai, P.. A multifaceted perspective at data analysis: A study in collaborative intelligent agents. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 2008;38(4):1062–1072.

[27] Loia, V., Pedrycz, W., Senatore, S.. Semantic web content analysis: A study in proximity-based collaborative clustering. Fuzzy Systems, IEEE Transactions on 2007;15(6):1294–1312.

[28] Mitra, H., Banka, , Pedrycz, W.. Rough-fuzzy collaborative clustering. IEEE Transactions on Systems, Man, and Cybernetics 2006;36:795–805.

[29] Tsoumakas, G., Angelis, L., Vlahavas, I.. Clustering classifiers for knowledge discovery from physically distributed databases. Data & Knowledge Engineering 2004;49(3):223 – 242.

[30] Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.. Constrained k-means clustering with background knowledge. In: International Conference on Machine Learning. 2001, p. 557–584.

[31] Bilenko, M., Basu, S., Mooney, R.J.. Integrating constraints and metric learning in semi-supervised clustering. In: International Conference on Machine Learning. 2004, p. 81–88.
[32] Huang, R., Lam, W.. An active learning framework for semi-supervised document clustering with language modeling. Data & Knowledge Engineering 2009;68(1):49–67.
[33] Grira, N., Crucianu, M., Boujemaa, N.. Active semi-supervised fuzzy clustering. Pattern Recognition 2008;41(5):1851–1861.
[34] Davidson, I., Wagstaff, K.L., Basu, S.. Measuring constraint-set utility for partitional clustering algorithms. In: European Conference on Principles and Practice of Knowledge Discovery in Databases. 2006, p. 115–126.
[35] Wagstaff, K.L.. Value, cost, and sharing: Open issues in constrained clustering. In: International Workshop on Knowledge Discovery in Inductive Databases. 2007, p. 1–10.
[36] Kumar, N., Kummamuru, K.. Semisupervised clustering with metric learning using relative comparisons. IEEE Transactions on Knowledge and Data Engineering 2008;20(4):496–503.
[37] Klein, D., Kamvar, S., Manning, C.. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In: In The Nineteenth International Conference on Machine Learning. 2002, p. 307–314.
[38] Basu, S., Banerjee, A., Mooney, R.J.. Active semi-supervision for pairwise constrained clustering. In: SIAM International Conference on Data Mining. 2004, p. 333–344.
[39] Basu, S., Banerjee, A., Mooney, R.J.. Semi-supervised clustering by seeding. In: International Conference on Machine Learning. 2002, p. 19–26.
[40] Gao, J., Tan, P., Cheng, H.. Semi-supervised clustering with partial background information. In: SIAM International Conference on Data Mining. 2006,.
[41] Basu, S., Bilenko, M., Mooney, R.J.. A probabilistic framework for semi-supervised clustering. In: International Conference on Knowledge Discovery and Data Mining. 2004, p. 59–68.
[42] Eick, C.F., Zeidat, N.,, , Zhao, Z.. Supervised clustering - algorithms and benefits. In: International Conference on Tools with Artificial Intelligence. 2004, p. 774–776.
[43] Pedrycz, W.. Fuzzy clustering with a knowledge-based guidance. Pattern Recognition Letters 2004;25(4):469–480.
[44] Bouchachia, A., Pedrycz, W.. Data clustering with partial supervision. Data Min Knowl Discov 2006;12(1):47–78.
[45] Pedrycz, W.. Collaborative and knowledge-based fuzzy clustering. International Journal of Innovative, Computing, Information and Control 2007;1(3):1–12.
[46] Fung, B.C., Wang, K., Wang, L., Hung, P.C.. Privacy-preserving data publishing for cluster analysis. Data & Knowledge Engineering 2009;68(6):552 – 575.
[47] Wemmert, C., Gancarski, P.. A multi-view voting method to combine unsupervised classifications. Artificial Intelligence and Applications 2002;:362–324.
[48] Cohen, J.. A coefficient of agreement for nominal scales. Educational and Psychological Measurement 1960;20(1):37.
[49] Rand, W.M.. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 1971;66:622–626.
[50] Tan, P., Steinbach, M., Kumar, V.. Introduction to Data Mining, (First Edition). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.; 2005.
[51] Halkidi, M., Vazirgiannis, M., Batistakis, Y.. On clustering validation techniques. Journal of Intelligent Information Systems 2001;17(2-3):107–145.
[52] Wemmert, C., Gançarski, P., Korczak, J.. A collaborative approach to combine multiple learning methods. International Journal on Artificial Intelligence Tools (World Scientific) 2000;9(1):59–78.
[53] Luo, P., Xiong, H., Zhan, G., Wu, J., Shi, Z.. Information-theoretic distance measures for clustering validation: Generalization and normalization. IEEE Transactions on Knowledge and Data Engineering 2009;21(9):1249–1262.
[54] Salvador, S., Chan, P.. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In: IEEE International Conference on Tools with Artificial Intelligence. 2004, p. 576–584.
[55] Macqueen, J.B.. Some methods of classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. 1967, p. 281–297.
[56] Asuncion, A., Newman, D.. Uci machine learning repository. 2007.
[57] Quinlan, J.. Improved use of continuous attributes in {C4.5}. Journal of Artificial Intelligence Research 1996;4:77–90.