

Semi-supervised collaborative clustering with partial background knowledge

Germain Forestier, Cédric Wemmert and Pierre Gançarski
LSIIT - CNRS - University of Strasbourg - UMR 7005
Pôle API, Bd Sébastien Brant - 67412 Illkirch, France
Email: {forestier,wemmert,gancarski}@lsiit.u-strasbg.fr

Abstract

In this paper we present a new algorithm for semi-supervised clustering. We assume to have a small set of labeled samples and we use it in a clustering algorithm to discover relevant patterns. We study how our algorithm works against two other semi-supervised algorithms when the data are multimodal. Then, we study the case where the user is able to produce few samples for some classes but not for each class of the dataset. Indeed, in complex problems, the user is not always able to produce samples for each class present in the dataset. The challenging task is consequently to use the set of labeled samples to discover other members of these classes, but also to keep a degree of freedom to discover unknown clusters, for which samples are not available. We address this problem through a series of experiments on synthetic datasets, to show the relevance of the proposed method.

1 Introduction

During the last years, an intensive work has been done in the field of data mining, especially on data classification [20]. Two main kinds of classification algorithms have emerged: the *supervised* classification algorithms [7] and the *unsupervised* classification algorithms [11]. In supervised classification, a fixed set of classes is available and examples for each class are used to build a classification function. This classification function is a predictive model used to find the class of new available patterns. In unsupervised classification (or *clustering*), an unlabeled dataset is partitioned into groups of similar items (i.e. *clusters*), generally by optimization of an objective function, which minimizes intra-class similarity and maximizes inter-class dissimilarity.

These two strategies for data classification have qualities and drawbacks. For supervised approaches, a large set of labeled samples is often needed to build an efficient classification function, but problems like overfitting can occur.

Concerning unsupervised approaches, the determination of the number of clusters [13], the method used to build the clusters [21] and the capacity to find complex shaped clusters [8] are still intensive fields of research. Consequently, many clustering algorithms have been proposed in the literature and new clustering algorithms continue to appear every year.

To address the problems related to these two approaches, a new kind of algorithms has been investigated during the last ten years under the names of *semi-supervised* classification [14] (or *semi-supervised* learning [22]) and *semi-supervised* clustering [1]. In semi-supervised classification, the training step uses unlabeled samples in addition to the available labeled samples, which generally leads to a better definition of the classification function. The function is consequently more generalizable. In semi-supervised clustering, the partitioning task is helped by constraints provided by the user or by a small set of labeled data. The constraints can be provided under the form of *must-link* and *cannot-link* constraints [17] giving respectively the information that two instances must be grouped together or be placed in different clusters. The use of a small set of labeled data has also been investigated [1], mainly to help the initialization of clustering algorithms.

In this paper, we are focused on this last kind of algorithms. We propose a new semi-supervised clustering algorithm using a collaborative clustering approach [9], which takes advantage of labeled samples. To the best of our knowledge, barely all the semi-supervised clustering algorithms assume a coherence between the available samples and the data space (i.e. the objects of a same class must all be placed in one area in the dataspace). Unfortunately, in complex data, this assumption is not always true. Indeed, when the user provides samples, he is not always aware of the complexity of the data space, and he does not know if the samples he provides are coherent. Two objects labeled of the same class can be localized in two different area in the data space, reflecting a multimodality, inherent to complex data. An advantage of the proposed method is its capacity to deal with such examples.

Moreover, in complex problems, the user is not always able to produce samples for each class of the dataset. All of the existing semi-supervised algorithms assume to have at least few samples for *each* expected class. We claim that this assumption is too strong and not realistic in many problems. Indeed, the user could be able to produce samples for certain classes, but can also expect to discover new classes into his data. By using a supervised or semi-supervised classification algorithm, the user loses the ability to understand his data and to discover new interesting patterns.

The outline of this paper is the following. In the Section 2, we review the main works of the literature about semi-supervised clustering. Then, in Section 3 we detail the proposed approach and in Section 4, we present some experiments to evaluate this method. Finally, conclusions and future works are drawn in Section 5.

2 Clustering with background knowledge

Many approaches have been investigated to use background knowledge to guide the clustering process. In constraint clustering, knowledge is expressed as *must-link* and *cannot-link* constraints. A *must-link* constraint gives the information that two data objects should be in the same cluster, and *cannot-link* means the contrary. This kind of knowledge is sometime easier to obtain than a classical subset of labeled samples.

In [17], Wagstaff et al. present a constrained version of the k-means algorithm which uses such constraints to bias the affectation of the objects to the clusters. At each step, the algorithm tries to agree with the constraints given by the user.

The constraints can be used to tune the algorithm, but also to learn a distance function which is biased by the knowledge about the constraints between the data objects, as presented in [4]. The distance between two data objects is reduced for a *must-link* and increased for a *cannot-link*. Recent works on constraint clustering are focused on evaluating the utility (i.e the potential interest) of a set of constraints [5, 18].

In [1], a subset of data objects is used to *seed* (i.e to initialize) the clusters of the k-means algorithm. Two algorithms, *seeded kmeans* and *constrained kmeans*, are presented. In the first algorithm, the samples are only used to initialize the clusters and can eventually be affected to another class during the clustering process. In the second algorithm, the clusters are initialized with the samples, which stay affected to their initial cluster. 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, where the labeled examples have moderate overlapping features with the unlabeled

data, Gao et al. [10] formulate a new approach as a constrained optimization problem. They propose 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, with limited labeled examples.

In [2], a pairwise constrained clustering framework is presented, 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.

Basu et al. [3] also propose 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 the proposed framework.

A new method, to allow instance-level constraints to have space level inductive implications to improve the use of the constraints, is given in [12]. The method improves the previously studied constrained k-means algorithm, generally requiring less than half as many constraints to achieve a given accuracy on a range of real-world data, while also being more robust when over-constrained.

In [15], the authors propose 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.

Another approach called supervised clustering [6], uses the feature of the data objects as well as the class information to build clusters with a high class purity. The goal of supervised clustering is to identify class-uniform clusters that have high probability densities.

Different kinds of background knowledge are introduced in [16], namely partial supervision, proximity-based guidance and uncertainty driven knowledge hints. The authors discuss the ways to exploit and effectively incorporate these background knowledge (called *knowledge hints*) in the fuzzy c-means algorithm.

3 Semi-supervised collaborative clustering

The proposed approach is divided in two main steps: a collaborative clustering of the data based on an existing collaborative clustering method [9], and a cluster labeling process which affects a class to each cluster, thanks to a subset of available samples.

In the first step, different clustering algorithms are applied on the data, with different initializations and different parameters. These clustering results are then refined

through a collaborative process to find an agreement about the classification of the data. This collaborative approach is used to discover the structure of the dataset. Indeed, by using different methods with different number of clusters, the methods will automatically find a good estimation of the number of clusters and the final classification will reflect the structure of the data. In the second step, a cluster labeling process is used to affect a class to each cluster.

In the next section, we describe the collaborative clustering process and then, the clusters labeling step.

3.1 Collaborative clustering

It is difficult to compute a consensual result from clustering results with different numbers of clusters or different structures (flat partitioning or hierarchical result), because of the lack of a trivial correspondence between the clusters of these different results. To address the problem, we present in this section a framework where different clustering methods work together in a collaborative way, to find an agreement about their proposals.

This collaborative process consists in 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 [19].

Before the description of the collaborative method, we introduce the correspondence function which is used in the system.

3.1.1 Intercluster correspondence function

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 unsupervised classifications, the results may not have a same number of clusters, and no information is available about the correspondence between the different clusters of the different results.

To address the problem, we have defined a new inter-cluster correspondence function, which associates to each cluster from a result, a cluster from each of the other results.

Let $\{\mathcal{R}^i\}_{1 \leq i \leq m}$ be the set of results given by the 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 a cluster \mathcal{C}_k^i from \mathcal{R}^i in the result \mathcal{R}^j , $i \neq j$, is the cluster from \mathcal{R}^j

which is the most similar to \mathcal{C}_k^i :

$$CC(\mathcal{C}_k^i, \mathcal{R}^j) = \mathcal{C}_\ell^j \quad \text{with } S(\mathcal{C}_k^i, \mathcal{C}_\ell^j) = \max \left(\left\{ S(\mathcal{C}_k^i, \mathcal{C}_l^j), \forall l \in [1, n_j] \right\} \right) \quad (1)$$

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

It is calculated from the recovery of the clusters in two steps. First, the intersection between each couple of clusters $(\mathcal{C}_k^i, \mathcal{C}_l^j)$, from two different results \mathcal{R}^i and \mathcal{R}^j , are calculated and stored in the *confusion matrix* $\mathcal{M}^{i,j}$:

$$\mathcal{M}^{i,j} = \begin{pmatrix} \alpha_{1,1}^{i,j} & \cdots & \alpha_{1,n_j}^{i,j} \\ \vdots & \ddots & \vdots \\ \alpha_{n_i,1}^{i,j} & \cdots & \alpha_{n_i,n_j}^{i,j} \end{pmatrix} \quad \text{where } \alpha_{k,l}^{i,j} = \frac{|\{\mathcal{C}_k^i \cap \mathcal{C}_l^j\}|}{|\mathcal{C}_k^i|} \quad (2)$$

Then, the similarity $S(\mathcal{C}_k^i, \mathcal{C}_l^j)$ between two clusters \mathcal{C}_k^i and \mathcal{C}_l^j is evaluated by observing the relationship between the size of their intersection and the size of the cluster itself, and by taking into account the distribution of the data in the other clusters as follows:

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

3.1.2 Collaborative process overview

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

1. *Initial clusterings* - Each clustering method computes a clustering of the data using its parameters.
2. *Results refinement* - A phase of convergence of the results, which consists of conflicts evaluation and resolution, is iterated as long as the quality of the results and their similarity increase.
3. *Unification* - If needed, the refined results are unified using a voting algorithm.

Initial clusterings. During this first step, each clustering method is initialized with its own parameters and a clustering is performed: all data objects are grouped into different clusters.

Results refinement. The mechanism we propose for refining the results is based on the concept of distributed local resolution of conflicts, by the iteration of four phases:

- Detection of the conflicts by evaluating the dissimilarities between couples of results;
- Choice of the conflicts to solve;

- Local resolution of these conflicts;
- Management of the local modifications in the global result (if they are relevant).

Conflicts detection - The detection of the conflicts consists in seeking all the couples $(\mathcal{C}_k^i, \mathcal{R}^j)$, $i \neq j$, such as $\mathcal{C}_k^i \neq CC(\mathcal{C}_k^i, \mathcal{R}^j)$. One conflict $\mathcal{K}_k^{i,j}$ is identified by one cluster \mathcal{C}_k^i and one result \mathcal{R}^j .

We associate to each conflict a measurement of its importance, the *conflict importance coefficient*, calculated according to the intercluster similarity:

$$CI(\mathcal{K}_k^{i,j}) = 1 - S(\mathcal{C}_k^i, CC(\mathcal{C}_k^i, \mathcal{R}^j)) \quad (4)$$

Choice of the conflicts to solve - During an iteration of refinement of the results, several local resolutions are performed in parallel. A conflict is selected in the set of existing conflicts and its resolution is started. This conflict, like all those concerning the two results involved in the conflict, are removed from the list of the conflicts. This process is iterated, until the list of the conflicts is empty.

Different heuristics can be used to choose the conflict to solve, according to the conflict importance coefficient (Eq. 4). We choose to try to solve the most important conflict first.

Local resolution of a conflict - The local resolution of a conflict $\mathcal{K}_k^{i,j}$ consists of applying an operator on each result involved in the conflict, \mathcal{R}^i and \mathcal{R}^j , to try to make them more similar.

The operators that can be applied to a result are the following:

- *merging* of clusters: some clusters are merged together (all the objects are merged in a new cluster that replaces the clusters merged),
- *splitting* of a cluster in 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 corresponding clusters of the cluster involved in the conflict. The *corresponding clusters (CCS)* of a cluster is an extension of the definition of the corresponding cluster (Eq. 1).

$$CCS(\mathcal{C}_k^i, \mathcal{R}^j) = \left\{ \mathcal{C}_l^j \mid S(\mathcal{C}_k^i, \mathcal{C}_l^j) > p_{cr}, \forall l \in [1, n_j] \right\} \quad (5)$$

where p_{cr} , $0 \leq p_{cr} \leq 1$, is given by the user.

Algorithm 1 Operator application

```

1: let  $n = |CCS(\mathcal{C}_k^i, \mathcal{R}^j)|$ 
2: let  $\mathcal{R}^{i'}$  (resp.  $\mathcal{R}^{j'}$ ) be the result of the application of an
   operator on  $\mathcal{R}^i$  (resp.  $\mathcal{R}^j$ )
3: if  $n > 1$  then
4:    $\mathcal{R}^{i'} = \mathcal{R}^i \setminus \{\mathcal{C}_k^i\} \cup \{\text{split}(\mathcal{C}_k^i, n)\}$ 
5:    $\mathcal{R}^{j'} = \mathcal{R}^j \setminus CCS(\mathcal{C}_k^i, \mathcal{R}^j) \cup \{\text{merge}(CCS(\mathcal{C}_k^i, \mathcal{R}^j))\}$ 
6: else
7:    $\mathcal{R}^{i'} = \text{reclustering}(\mathcal{R}^i, \mathcal{C}_k^i)$ 
8: end if

```

Having found the corresponding clusters of the cluster involved in the conflict, an operator is chosen and applied as follows:

But the application of the two operators is not always relevant. Indeed, it does not always increase the similarity of the results implied in the conflict treated, and especially, the iteration of conflict resolutions may lead to a trivial solution where all the methods are in agreement. For example, they can converge towards a result with only one cluster including all the objects to classify, or towards a result having 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, based on the intercluster similarity S (Eq. 3) and a quality criterion δ :

$$\gamma^{i,j} = \frac{1}{2} \left(p_s \cdot \left(\frac{1}{n_i} \sum_{k=1}^{n_i} \omega_k^{i,j} + \frac{1}{n_j} \sum_{k=1}^{n_j} \omega_k^{j,i} \right) + p_q \cdot (\delta^i + \delta^j) \right) \quad (6)$$

where

$$\omega_k^{i,j} = \sum_{l=1}^{n_j} S(\mathcal{C}_k^i, CC(\mathcal{C}_k^i, \mathcal{R}^j)) \quad (7)$$

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

At the end of each conflict resolution, the local similarity criterion enables to choose which couple of results are to be kept: the two new results, the two old results, or one new result with one old result.

Global management of the local modifications - After the resolutions of all these local conflicts, a global application of the modifications proposed by the refinement step, is decided if it improves the quality of the global result. The *global agreement coefficient* of the results is evaluated according to all the local similarity between each couple of results. It evaluates the global similarity of the results and their quality.

$$\Gamma = \frac{1}{m} \sum_{i=1}^m \Gamma^i \quad (8)$$

where

$$\Gamma^i = \frac{1}{m-1} \sum_{\substack{j=1 \\ j \neq i}}^m \gamma^{i,j} \quad (9)$$

Even if the local modifications decrease this global agreement coefficient, the solution is accepted to avoid to fall in a local maximum. If the coefficient is decreasing too much, all the results are reinitialized to the best temporary solution (the one with the best global agreement coefficient). In this case, the conflicts that have not been solved are removed to avoid oscillations of the agreement coefficient. The global process is iterated until some conflicts can be solved.

Unification. In the final step, all the results tend to have the same number of clusters, which are increasingly similar. Thus, we use a voting algorithm [19] to compute a unified result combining the different results. This multi-view voting algorithm enables to combine in one unique result, many different clustering results that have not 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 the corresponding cluster of \mathcal{C}_k^i in all the other results. The maximum of these values indicates the *best cluster* for the object, for example \mathcal{C}_l^j . That means that this object should be in the cluster \mathcal{C}_l^j according to the opinion of all the methods. After having done the vote for all objects, a new cluster is created for each *best cluster* found if a majority of the methods have voted for this cluster.

3.2 Clusters labeling

After the collaborative clustering and the unification of the different results, we obtain a clustering result \mathcal{R}^u composed of n_u clusters $\{\mathcal{C}_k^u\}_{1 \leq k \leq n_u}$.

Let $\mathcal{S} = \{s_l\}_{1 \leq l \leq n_s}$ be the set of samples available for the dataset. For each sample s_l of \mathcal{S} , the known class (label) is given by $cl(s_l)$.

To label the clusters of \mathcal{R}^u , we apply a voting method as follow:

$$cl(\mathcal{C}_k^u) = \arg \max_k \{|\mathcal{C}_k^u \cap \mathcal{S}^k|\} \quad (10)$$

where

$$\mathcal{S}^k = \{s_l \in \mathcal{S} : cl(s_l) = k\} \quad (11)$$

Table 1. Results on the four datasets.

dataset	Proposed	SKMeans	CKMeans
<i>Dataset 1</i>	96.75(± 1.43)	97.75(±0.00)	97.96(±0.22)
<i>Dataset 2</i>	98.36(± 0.48)	61.81(±7.21)	62.75(±7.50)
<i>Dataset 3</i>	91.18(±10.81)	55.56(±0.69)	55.57(±0.75)
<i>Dataset 4</i>	97.98(±3.87)	70.3(±5.09)	67.12(±3.64)

If no sample is available in a cluster, that is

$$\forall k, \mathcal{C}_k^u \cap \mathcal{S}^k = \emptyset$$

a new label is given to this cluster. This enables to discover new patterns in the data.

4 Experimental results

In this section, we present different experiments carried out to illustrate how the method works. In the first experiment, we used three synthetic datasets (see Fig. 1), each containing two classes. In *Dataset 1*, the classes are well separable in the data space, while in *Dataset 2* one of the classes is bimodal and is composed of two clusters. Finally, in *Dataset 3*, the two classes are multimodal and are composed of separated groups of objects in the data space.

For this experiment we used the KMeans algorithm as the base method for the collaborative system. We set up the method with four different initializations of the KMeans algorithm with different numbers of clusters, randomly picked in [2, 10]. Then, the collaborative process presented in Section 3 is run to find an agreement among these different methods, which will reflect the structure of the dataset. Finally, the cluster labeling process (Section 3.2) is used to affect a class to each cluster (if it is possible).

The Tab. 1 presents the accuracy obtained after the application of our clustering method. The values are the average and standard deviation of 100 runs. For comparison purpose, we also applied two different semi-supervised clustering algorithms, namely *seeded kmeans* (SKMeans) and *constrained kmeans* (CKMeans) [1]. These two algorithms were set up to find two clusters as they are set up according to the available samples. For this experiment, 5% of the dataset was given as background knowledge, randomly chosen at each run.

One can see on Tab. 1 that when the dataset is simple (*Dataset 1*), the methods are almost equivalent. However, when the data are more complex and multimodal, classical semi-supervised algorithm logically failed to discover the underlying structure of the classes. In these cases (*Dataset 2 and 3*), the proposed method clearly outperformed the two others.

The Fig. 2 shows the accuracy according to the percentage of available samples. As one can see, the proposed

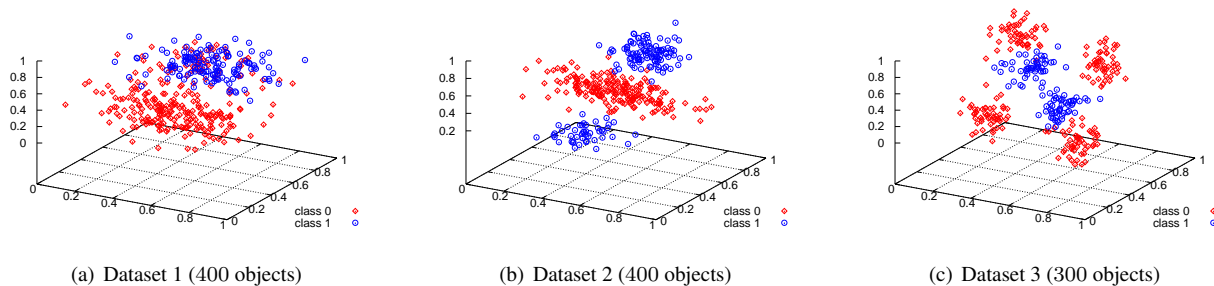


Figure 1. The three dataset used for the experiments.

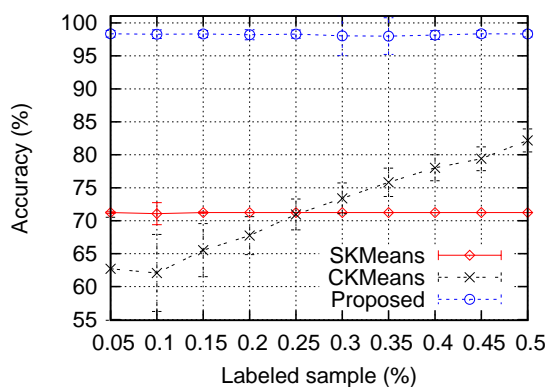


Figure 2. The accuracy according to available sample on *Dataset 2*.

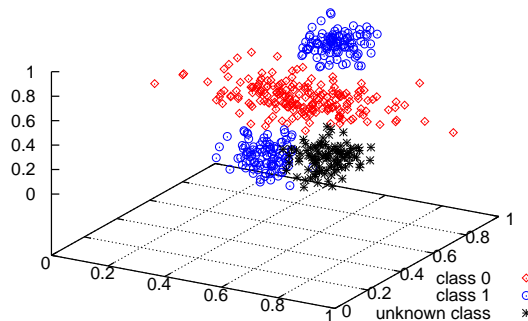


Figure 3. The *Dataset 4* (500 objects) which contains one unknown cluster.

method shows a great stability, the accuracy being stable from 5% to 50% of available samples.

Another experiment has been made, when samples are available for some classes but not for all. The Fig. 3 presents the *Dataset 4*, which is similar to the *Dataset 2* with the addition of one unknown cluster. The challenge here, is to use the available samples to identify the two classes, but also to discover the third cluster, corresponding to a hidden pattern, where no background knowledge is available. The results for the *Dataset 4* are presented in Tab. 1. Here again, the proposed method outperforms the others to discover interesting patterns for the user, but also to identify unknown underlying knowledge.

5 Conclusions

In this paper, we have presented a new semi-supervised clustering algorithm. This algorithm is decomposed in two steps. Firstly a collaborative clustering of the data reveals the underlying structure of the dataset. Secondly, a clusters

labeling, using a small subset of available samples, affects a class to each cluster.

The method is able to use samples provided by the user, which are not necessarily grouped together in the data space. This kind of knowledge can occur in mining complex multimodal data. The user is not always aware of the shape of the data space and can consequently provide sample placed in various places of the dataset for the same class. Furthermore, the method is able to use background knowledge provided by the user, but can keep a degree of freedom to discover hidden patterns. This feature allows the users to use the knowledge they have on their data but also to discover relevant unknown patterns.

In future works, we plan to improve the method through different modifications. For example, a problem occurs when no available sample is present in a cluster, consequently this cluster can not be labeled. This problem can be solved by setting to these clusters the labels of the nearest labeled cluster (if all the clusters need to be labeled). An-

other modification will consist to label the clusters of each result before the unification at the end of the collaboration process. A classical supervised voting method could then be used to affect a class to each object. This approach should be compare to the one presented in this paper.

Finally, more experiments will be conducted, especially on real life problems, as for example object-based remote sensing image interpretation.

References

- [1] S. Basu, A. Banerjee, and R. J. Mooney. Semi-supervised clustering by seeding. In *International Conference on Machine Learning*, pages 19–26, 2002.
- [2] S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *SIAM International Conference on Data Mining*, 2004.
- [3] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *International Conference on Knowledge Discovery and Data Mining*, pages 59–68, 2004.
- [4] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *International Conference on Machine Learning*, pages 81–88, 2004.
- [5] I. Davidson, K. L. Wagstaff, , and S. Basu. Measuring constraint-set utility for partitional clustering algorithms. In *European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 115–126, 2006.
- [6] C. F. Eick, N. Zeidat, , and Z. Zhao. Supervised clustering - algorithms and benefits. In *International Conference on Tools with Artificial Intelligence*, pages 774–776, 2004.
- [7] Z. Fan-Zi and Q. Zheng-Ding. A survey of classification learning algorithm. *International Conference on Signal Processing*, 2:1500–1504, 2004.
- [8] A. Fred and A. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005.
- [9] P. Gancarski and C. Wemmert. Collaborative multi-step mono-level multi-strategy classification. *Journal on Multimedia Tools and Applications*, 35(1):1–27, october 2007.
- [10] J. Gao, P.-N. Tan, , and H. Cheng. Semi-supervised clustering with partial background information. In *SIAM International Conference on Data Mining*, 2006.
- [11] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.
- [12] D. Klein, S. Kamvar, and C. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering, 2002.
- [13] R. Kothari and D. Pitts. On finding the number of clusters. *Pattern Recognition Letters*, 20(4):405–416, 1999.
- [14] B. Krishnapuram, D. Williams, Y. Xue, A. J. Hartemink, L. Carin, and M. A. T. Figueiredo. On semi-supervised classification. In *Neural Information Processing Systems*, 2004.
- [15] M. Law, A. Topchy, and A. Jain. Multiobjective data clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 424–430, 2004.
- [16] W. Pedrycz. Fuzzy clustering with a knowledge-based guidance. *Pattern Recognition Letters*, 25(4):469–480, 2004.
- [17] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *International Conference on Machine Learning*, pages 557–584, 2001.
- [18] K. L. Wagstaff. Value, cost, and sharing: Open issues in constrained clustering. In *International Workshop on Knowledge Discovery in Inductive Databases*, pages 1–10, 2007.
- [19] C. Wemmert and Gancarski. A multi-view voting method to combine unsupervised classifications. *Artificial Intelligence and Applications*, pages 362–324, 2002.
- [20] I. H. Witten and F. Eibe. *Data mining : pratical machine learning tools and techniques*. Morgan Kaufmann, 2005.
- [21] R. Xu and D. Wunsch. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.
- [22] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.