Data Clustering With Partial Supervision

Abdelhamid Bouchachia
Dept. of Informatics, University of Klagenfurt
Universitaetsstr. 65, A-9020 Klagenfurt, Austria
hamid@isys.uni-klu.ac.at

Witold Pedrycz
Dept. of Electrical and Computer Engineering, University of Alberta
Edmonton, T6G 2V4, Canada
pedrycz@ee.ualberta.ca

Abstract. Clustering with partial supervision finds its application in situations where data is neither entirely nor accurately labeled. This paper discusses a semi-supervised clustering algorithm based on a modified version of the fuzzy C-Means (FCM) algorithm. The objective function of the proposed algorithm consists of two components. The first concerns traditional unsupervised clustering while the second tracks the relationship between classes (available labels) and the clusters generated by the first component. The balance between the two components is tuned by a scaling factor. Comprehensive experimental studies are presented. First, the discrimination of the proposed algorithm is discussed before its reformulation as a classifier is addressed. The induced classifier is evaluated on completely labeled data and validated by comparison against some fully supervised classifiers, namely support vector machines and neural networks. This classifier is then evaluated and compared against three semi-supervised algorithms in the context of learning from partly labeled data. In addition, the behavior of the algorithm is discussed and the relation between classes and clusters is investigated using a linear regression model. Finally, the complexity of the algorithm is briefly discussed.

Keywords: Clustering, partial supervision, classification, class discrimination, linear regression.

1. Introduction

One of the most interesting techniques in pattern recognition, data mining and knowledge discovery is clustering. The aim of clustering is to find the hidden structure underlying a given collection of data points. The task consists of partitioning the data set into clusters such that similar data points (feature vectors) are grouped into the same cluster. As in other machine learning paradigms, clustering can be applied using two different modes:

- Supervised: the process of assigning data points to groups is known as classification. This process relies on the availability of knowledge about the data being analyzed. Knowledge here represents the set of labels associated with data. If data is continuous (i.e. signal),

Data Mining and Knowledge Discovery, Vol. 12, No. 1, 2006

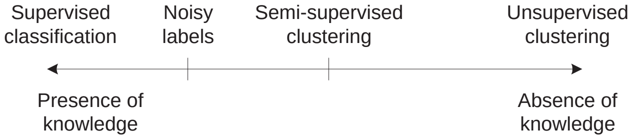


Figure 1. Clustering spectrum with respect to knowledge

the process is known as regression. In supervised mode, the class label and the number of classes are predefined.

– Unsupervised: the process of assigning unlabeled data points to clusters using some similarity measure (i.e. distance-based, density-based, etc.) is known as clustering. This process is self-supervised. Ideally, two criteria have to be satisfied, namely intra-cluster similarity (homogeneity) and inter-cluster dissimilarity.

These modes appear to be extreme in the sense that the former requires complete knowledge of the data while the later uses no knowledge at all. Acquiring knowledge (labeling) of the data points is always an expensive and error-prone task that takes time and human effort. In many situations, the data is neither perfectly nor completely labeled. Thus, we may attempt to benefit from the available knowledge (labeled data) to cluster unlabeled data. This form of combining labeled and unlabeled data to generate the structure of the whole data set is known as *semi-(or partially) supervised clustering* (see Fig. 1).

Labeled data points, considered as a small subset of the data, are used to guide the process of grouping and, at the same time, to boost the accuracy of unsupervised clustering. The goal is then to relate clusters belonging to the same class. Unlabeled data, on the other hand, which is generally large in size, helps to boost the performance of supervised classification and to discover the hidden structure of the data. Such a structure might not always be accessible to the data analyst. The problem can be envisioned as shown in Fig. 2. Two classes are considered. The first consists of four clusters (represented by circles) and the second consists of two clusters (represented by asterisks). The goal is to relate clusters that are in the same class.

Typical applications for clustering with partial supervision are *medical diagnosis* where the doctor needs assistance from the machine to identify some suspiciously labeled data points or when it is hard to acquire labels of the whole medical data, *image processing* where only some objects or regions of the image are labeled, *web retrieval* where labeling all documents is a very costly activity, and *information filtering* where knowledge about the user's profile is limited. These applications

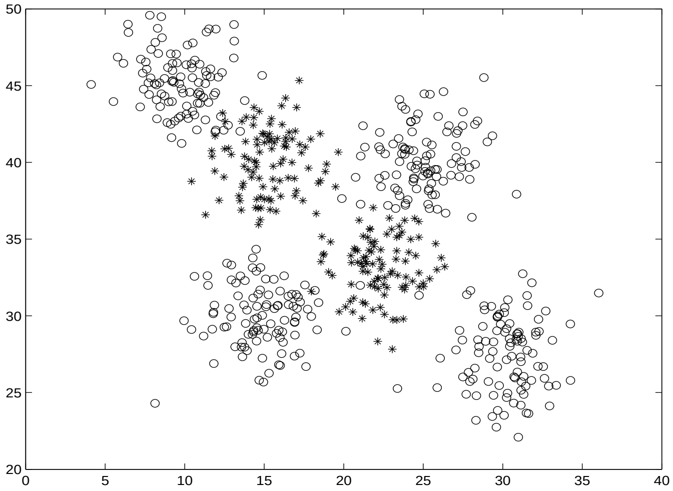


Figure 2. Typical clustering situation

share one common aspect, that is how to take advantage of the presence of some knowledge about the data. The proposed approach is applied in a medical context as will be discussed in Sec. 4.

The paper is organized as follows. Section 2 surveys the state of the art in the framework of semi-supervised clustering. Section 3 introduces the details of our semi-supervised clustering algorithm. Section 4 discusses the analysis of the algorithm according to various aspects. Its discrimination power is presented in Sec. 4.1. Section 4.2 shows how the algorithm is formulated as a classifier and discusses its performance. In Sec. 4.3, a comparison of the algorithm performance against that of some fully supervised classification algorithms is presented. Section 4.4 describes the evaluation of the algorithm on partly labeled data. Then, in Sec. 4.5, the algorithm is compared against three semi-supervised learning algorithms on partly labeled data. Furthermore, the behavior of the algorithm is outlined in Sec. 4.6. Two further aspects namely, the relationship between clusters and classes via a linear regression model and the complexity of the algorithm are discussed in Sec. 4.7 and Sec. 4.8 respectively. Finally, Sec. 5 highlights some future work and concludes the paper.

2. Related Work

Several semi-supervised algorithms have been proposed. They can be categorized based on the computational model used. The most known models are: *the seeding model*, *the probabilistic model*, *the objective function optimization model*, *genetic algorithms*, *support vector machines*, and *graph-based model*. Some illustrative examples based on these models are presented below.

Basu et al. (2002) proposed two semi-supervised algorithms based on a seeding mechanism. These algorithms rely on the k-means algorithm. In the first algorithm, the seeds are used to initialize the partition centers and then updated during the clustering process. In the second algorithm, once initialized with the seeds, the centers are not updated. The idea here is that when seeds are noise-free, centers may be kept unchanged.

Nigam et al. (2000) investigated a probabilistic approach for text classification. The approach combines the Expectation-Maximization (EM) algorithm and a naive Bayes classifier. The algorithm trains the classifier using the labeled data only. Then, the labels of the unlabeled samples are iteratively estimated and the classifier is re-trained using all labeled data until convergence. Blum and Mitchell (1998) applied the co-training technique for document clustering. In co-training, the feature set is split into two independent subsets. Each of these is used to train a particular algorithm. Similar work is presented by Amini and Gallinari (2003), where a variant form of the EM algorithm is applied to train discriminant classifiers on data that are not completely labeled.

Jeon and Landgrebe (1999) suggested a partially-supervised classification algorithm to discriminate a particular class of interest. The goal was to design a classification algorithm given only the labeled data points. The proposed algorithm relies on three steps. In the first step, each data point is assigned a weight representing the likelihood of not being in the class of interest. In the second step, the clusters are initialized using a probabilistic unsupervised algorithm. In the last step, clusters are refined and adjusted.

Support vector machines (SVM) have also been used to perform clustering with partial supervision. Klinkenberg (2001) discussed the problem of information filtering as a task that requires the use of unlabeled documents to reduce the need for labeled documents known to be of interest to users. A SVM algorithm is first trained on a window of labeled data and then on a window of unlabeled data, taking care to choose the appropriate size of the window. This approach can be generalized to that called semi-supervised classification based on kernel clustering (Bennett and Demiriz, 1999; Chapelle et al., 2002).

Demiriz et al. (1999) applied genetic algorithms to combine supervised classification and unsupervised clustering. The basic idea in this work is to minimize an objective function that is a linear combination of the cluster dispersion and the cluster impurity of the form:

$$a * cluster\_dispersion \ + \ b * cluster\_impurity \tag{1}$$

The first component of this formula is concerned with unsupervised clustering while the second component controls the purity of the generated clusters and is therefore concerned with supervised classification.

Graph-based methods depart from the idea that the data (labeled and unlabeled) is represented as a graph. The samples are represented as nodes which are interconnected by weighted edges. The weights indicate the similarity between samples. Basically, these approaches use a loss function and a regularization factor (Blum et al., 2004; Zhu et al., 2005) to propagate labels of the labeled samples to the unlabeled samples lying in the vicinity.

More relevant to our work is the approach investigated by Pedrycz and Waletzky (1997) which is a typical example of the objective function optimization model for clustering with partial supervision. In this work, a modified version of the FCM algorithm was proposed to deal with the problem of partial supervision. The objective function was extended to include a second term that expresses the relationship between classes and clusters. In this objective function, labeled and unlabeled data are identified by means of a boolean vector: $b = [b_k]$, $k = 1, 2, ..., N$, where N is the size of the data set. $b_k = 1$ if sample x_k is labeled, otherwise 0. Likewise, the membership values of the labeled samples are arranged in a matrix $F = [f_{ik}]$ such that $i = 1, 2, ..., C$ (the number of clusters) and $k = 1, 2, ..., N$. The objective function is:

$$J(U,V) = \sum_{i=1}^{C}\sum_{k=1}^{N} u_{ik}^{m} \left\|x_k - v_i\right\|^2 + \alpha \sum_{i=1}^{C}\sum_{k=1}^{N} (u_{ik} - f_{ik}b_k)^m \left\|x_k - v_i\right\|^2 \tag{2}$$

where N is the size of the data set, C is the number of clusters, and U designates the partition matrix such that each u_{ik} indicates the membership degree of the data point k to cluster i. $u_{ik} = 1$ means full membership of the data point x_k to cluster i, $u_{ik} = 0$ means that x_k does not belong to cluster i, and $0 < u_{ik} < 1$ means that x_k partly belongs to cluster i. V represents the set of prototypes v_i associated with clusters. The superscript m is the degree of fuzziness associated with the partition matrix ($m > 1$) (the higher m is, the fuzzier the membership of data points to clusters). The parameter α is a scaling factor to maintain the balance between the supervised and unsupervised components of the objective function. If $\alpha = 0$, the objective function reduces to that of FCM. This modified version of the FCM algorithm assumes that the number of classes is predetermined and is reflected in the matrix F, hence the number of clusters equals the number of classes.

The approach suggested in this paper overcomes the limitation observed in the approach investigated by Pedrycz and Waletzky (1997). The point of departure is to avoid the assumption that the number of clusters determined by the clustering algorithm should be the same as the number of classes reflected by data labels. In many real-world situations the available labels do not reflect the whole structure of the data. Our proposal deals with such situations.

3. Semi-Supervised Clustering Model

The algorithm suggested here relies on FCM (Bezdek, 1981). Basically, it extends the objective function of FCM to capture the hidden and the visible data structures. The hidden data structure is discovered using the FCM objective function as the first term of the proposed objective function. The second term takes into account the visible data structure reflected by the available labels. Thus, the objective function becomes:

$$J(U,V) = \sum_{i=1}^{C}\sum_{k=1}^{N} u_{ik}^{m} d_{ik}^{2} + \alpha \sum_{i=1}^{C}\sum_{k=1}^{N} (u_{ik} - \tilde{u}_{ik})^{m} d_{ik}^{2} \tag{3}$$

such that:

$$\sum_{i=1}^{C} u_{ik} = 1 \ \ \forall\, k, \quad 0 < \sum_{k=1}^{N} u_{ik} < N, \ \ \forall\, i \tag{4}$$

and

$$d_{ik}^{2} = \|x_{\mathrm{k}} - v_i\|_2^2 = (x_k - v_i)^T (x_k - v_i) \tag{5}$$

It is natural to assume that a class can be partitioned into several clusters. If H designates the number of classes (labels), then $C \geq H$. Each class h contains a number of clusters C_h, hence:

$$\sum_{h=1}^{H} C_h = C \tag{6}$$

The terms $\tilde{u}_{ik}$ of the matrix $\tilde{U}$ are iteratively computed as follows:

$$\tilde{u}_{ik}^{(r)} = \tilde{u}_{ik}^{(r-1)} - \beta \frac{\partial Q(F,\tilde{U})}{\partial \tilde{u}_{ik}} \tag{7}$$

where r is the iteration counter and

$$Q(F,\tilde{U}) = \sum_{h=1}^{H}\sum_{k=1}^{N} \delta_k \left(f_{hk} - \sum_{i \in \pi_h} \tilde{u}_{ik} \right)^2 \tag{8}$$

such that

$$\tilde{u}_{ik} \in [0, 1]$$

In Eq. 8, $F = [f_{hk}]$ is an $H \times N$ binary matrix such that $f_{hk} = 1$ if data point x_k belongs to class h, otherwise 0. This matrix serves to represent the labeling information of the data. δ_k is a binary vector that specifies whether the data point is labeled or not ($\delta_k = 1$ if the sample k is labeled, 0 otherwise). π_h is the set of clusters belonging to class h. The way the set π_h is specified will be discussed later. In Eq. 7, β is a strictly positive parameter representing a learning rate. The minimization of Q (Eq. 8) involving:

$$\gamma_{hk} = f_{hk} - \sum_{i \in \pi_h} \tilde{u}_{ik} \tag{9}$$

aims at minimizing the difference between the assigned membership and the sum of all membership degrees for the labeled point with respect to all clusters involved within the same class. Unlabeled data is handled by only the unsupervised component of the objective function since $\delta_k = 0$. Therefore, the clustering process is guided more by the labeled data. Eq. 7 optimizes the amounts $\tilde{u}_{ik}$, exploiting the computed difference and the learning rate β, with the overall goal of reducing the difference between the actual membership grade u_{ik} of a labeled data point k to a cluster i and the evolving membership $\tilde{u}_{ik}$ expressed in the second term of Eq. 3. The resulting matrix $\tilde{U}$ is used to compute the second term in Eq. 3, that is the difference between U and $\tilde{U}$. The algorithm consists of two optimizations $Min\{J(U, V)\}$ and $Min\{Q(F, \tilde{U})\}$. Let us outline the way the two performance indices interact:

1. Initialization: at the beginning of the clustering process, the matrix $\tilde{U}$ (containing the terms $\tilde{u}_{ik}$) is initialized with the actual partition matrix U.

2. Optimization: iterate the following two steps:

   a) Optimize Eq. 7 using Eq. 8 (Optimization of $\tilde{U}$).

   b) Optimize Eq. 3 using the output of step 2a.

For the sake of simplicity, we set the fuzziness degree m in Eq. 3 to 2. Note also that we will use the Euclidean distance throughout this paper. Therefore, it is advisable to normalize the values in the data set before calculating the distance. In a real-world data set, attributes (features) can be measured against different scales. For example, one attribute can measure the "age" of a person and another attribute can measure the "height". Discrepancies resulting from the difference in the scale (or domain) of the attributes can distort the distance calculations. One can use decimal scaling (by dividing each feature value by the same power of 10), Min-Max normalization (by mapping the feature value *val* to *val′* in the range $[new\_minF, new\_maxF]$, where new_minF and new_maxF are computed using the actual *min* and *max* of the data), Z-scores (by replacing the feature value with the standardized difference from the mean of the feature), and logarithmic normalization (by replacing the feature value with a base b logarithm). The goal of normalization is to convert all the values in the data set to the same proportional scale. Note also that the Euclidean distance (L_2 metric) assumes that only the overall distance is important, while the Manhattan metric (L_1) assumes that every difference is equally important, and the L_∞ metric assumes that only the largest difference is important. Thus, the normalization process aims at rendering all dimensions (features) equally important. Further details on various distance metrics can be found in Hathaway et al. (2000).

To optimize Eq. 3 such that the condition in Eq. 4 is satisfied, we partially differentiate $J(U, V)$ with respect to the partition matrix U and the prototypes v_i. Applying the Lagrangian multiplier for each $k = 1, 2, ..., N$, we have:

$$J(U, V, \lambda) = \sum_{i=1}^{C}\sum_{k=1}^{N} u_{ik}^2 d_{ik}^2 + \alpha \sum_{i=1}^{C}\sum_{k=1}^{N} (u_{ik} - \tilde{u}_{ik})^2 d_{ik}^2 - \lambda(\sum_{i=1}^{C} u_{ik} - 1)$$

By setting $\frac{\partial J(U,V,\lambda)}{\partial u_{st}} = 0$ for a given point t and a cluster s, we get:

$$2u_{st}d_{st}^2 + 2\alpha(u_{st} - \tilde{u}_{st})d_{st}^2 - \lambda = 0$$

leading to:

$$u_{st} = \frac{2\alpha\tilde{u}_{st}d_{st}^2 + \lambda}{2(1+\alpha)d_{st}^2} = \frac{\alpha\tilde{u}_{st}}{(1+\alpha)} + \frac{\lambda}{2(1+\alpha)d_{st}^2} \tag{10}$$

By setting $\frac{\partial J(U,V,\lambda)}{\partial \lambda_t} = 0$, we have $\sum_{i=1}^{C} u_{it} = 1$, leading to:

$$\lambda \sum_{i=1}^{C} \frac{1}{2(1+\alpha)d_{it}^2} + \frac{\alpha}{1+\alpha}\sum_{i=1}^{C} \tilde{u}_{it} = 1$$

$$\lambda = \frac{1 - \frac{\alpha}{1+\alpha}\sum_{i=1}^{C} \tilde{u}_{it}}{\sum_{i=1}^{C} \frac{1}{2(1+\alpha)d_{it}^2}}$$

Substituting the expression of λ in Eq. 10, we get:

$$u_{st} = \frac{\alpha\tilde{u}_{st}}{(1+\alpha)} + \frac{\frac{1 - \frac{\alpha}{(1+\alpha)}\sum_{i=1}^{C} \tilde{u}_{it}}{\sum_{i=1}^{C} \frac{1}{2(1+\alpha)d_{it}^2}}}{2(1+\alpha)d_{st}^2}$$

and u_{st} becomes:

$$u_{st} = \frac{\alpha \tilde{u}_{st}}{(1+\alpha)} + \frac{1 - \frac{\alpha}{(1+\alpha)} \sum_{i=1}^{C} \tilde{u}_{it}}{\sum_{i=1}^{C} \frac{d_{st}^2}{d_{it}^2}} \tag{11}$$

Then, by setting $\frac{\partial J(U,V,\lambda)}{\partial v_s} = 0$ (here 0 is the null vector) and taking the expression of d_{ik}^2 in Eq. 5 into account, we have:

$$-2 \sum_{k=1}^{N} u_{sk}^2 (x_k - v_s) - 2\alpha \sum_{k=1}^{N} (u_{sk} - \tilde{u}_{sk})^2 (x_k - v_s) = 0$$

Thus, v_s is expressed as follows:

$$v_s = \frac{\sum_{k=1}^{N} (u_{sk}^2 + \alpha (u_{sk} - \tilde{u}_{sk})^2) x_k}{\sum_{k=1}^{N} (u_{sk}^2 + \alpha (u_{sk} - \tilde{u}_{sk})^2)} \tag{12}$$

Clearly Eqs. 11 and 12 depend on the optimal value of $\tilde{u}_{st}$, which is obtained by differentiating Q in Eq. 8:

$$\frac{\partial Q(F, \tilde{U})}{\partial \tilde{u}_{st}} = -2 \sum_{h=1}^{H} \delta_t \left(f_{ht} - \sum_{i \in \pi_h} \tilde{u}_{it} \right) * \begin{cases} 1 & if \ s \in \pi_h \\ 0 & otherwise \end{cases}$$

Finally, the learning rule in Eq. 7 is transformed into:

$$\tilde{u}_{st}^{(r)} = \tilde{u}_{st}^{(r-1)} + 2\beta \delta_t \sum_{h=1}^{H} \left(f_{ht} - \sum_{i \in \pi_h} \tilde{u}_{it}^{(r-1)} \right) * \begin{cases} 1 & if \ s \in \pi_h \\ 0 & otherwise \end{cases} \tag{13}$$

The expression γ_{ik} in Eq. 9 aims at minimizing the difference between the hard membership degree {0, 1} of a data point k to a class h and the sum of membership degrees of k to all clusters belonging to class h. By fixing the number of clusters for each class, we can test the desired combination of clusters and then find the optimal one. The desired combination is specified by means of π_h (see Eq. 13). It is part of the initialization process. Now we need to know π_h or, more specifically, the amount:

$$\psi_h = \sum_{i \in \pi_h} \tilde{u}_{ik} \tag{14}$$

To compute ψ_h, a mapping matrix $M_{H \times C}$ and a corresponding matrix $P_{H \times C}$ are used. The former specifies the relationship between the classes and clusters while the latter specifies the number of data points from each class in each cluster. A row in P corresponds to a class index and a column corresponds to a cluster index. A cell $P(h, i)$ represents the number of data points of class h appearing in cluster i. A data point k from class h belongs to cluster i if the membership degree of k to i is the highest. The membership degrees are provided in the partition matrix U. Thus, to each class a list of clusters and the number of data points from this class in each cluster of this list will be associated. To force the algorithm to generate a given combination of clusters, say $(C_1, ..., C_h, ..., C_H)$, the mapping matrix M and its corresponding matrix P are used. After sorting P in an ascending order, each row of M contains the list of clusters ordered by dominance. The number of clusters per class specified in the requested combination is taken into account. The set π_h of ψ_h contains those dominant clusters in class h.

After having formulated all required expressions, the clustering process can then be formulated. It consists of the steps shown in Alg. 1.

Algorithm 1 : The semi-supervised clustering algorithm

```
1. Apply the standard FCM on the whole data set (both labeled and
unlabeled data points) to get the partition matrix $U^{(0)}$.
2. Determine the set $\pi_h$ of each class using the notion of dominance
explained below.
3. Compute the mapping matrix $M_{(H \times C)}$ that relates classes to
clusters: $M(h,i) = 1$ if cluster $i$ is in class $h$, otherwise 0.
4. Initialize $\tilde{U}^{(0)}$ with $U^{(0)}$ and set the iteration counter $r = 1$.
repeat
   repeat
      a. Compute $\tilde{U}^{(r)}$ using Eq. 13 (i.e. Optimize $Q(F, \tilde{U})$)
   until $||\tilde{U}^{(r)} - \tilde{U}^{(r-1)}|| < \tau$ where $\tau$ is a small threshold
   repeat
      b. Compute $V^{(r)}$ using Eq. 12
      c. Compute $U^{(r)}$ using Eq. 11 (steps b. and c. correspond to
      optimizing $J(U, V)$)
   until $||U^{(r)} - U^{(r-1)}|| < \epsilon$ where $\epsilon$ is a small threshold
   d. Compute the mapping matrix $M^{(r)}$
until $M^{(r)} = M^{(r-1)}$ or $r = MaxIter$
```

4. Analysis of the Algorithm

To analyze this algorithm, two data sets are used: a synthetic data set and a real medical data set. The synthetic data set is generated according to some statistical characteristics, namely a mean and a covariance (μ, Σ). Two classes, as shown in Tab. I and plotted in Fig. 3,

Table I. Characteristics of the synthetic data set

Characteristics		μ (dim1)	μ (dim2)	Σ (dim1)	Σ (dim2)
Classes	H_1	3.0	1.0	7.0	0.5
	H_2	4.0	4.5	1.0	1.0
		2.0	-2.5	1.0	1.0

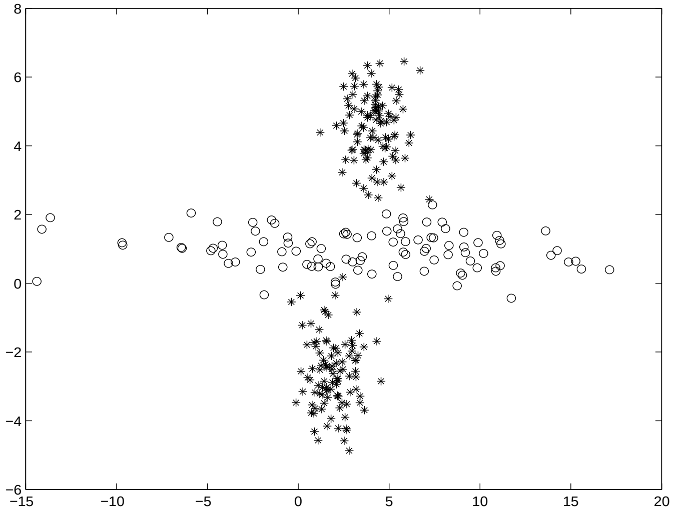


Figure 3. The synthetic data set (o: class 1, *: class 2)

are used in the experiments: the first class, H_1, consists of one cluster and the second class, H_2, consists of two clusters. Each cluster consists of 100 data points. The real medical data set consists of 206 magnetic resonance (MR) spectra (360 MHz, 37^o) of human brain neoplasms distributed into three classes as follows: 95 meningiomas (M), 37 control samples of non-tumorous brain tissue from patients with epilepsy (E), and 74 astrocytomas (A). The dimensionality is 550 (obtained in the region of 0.3-4.0 ppm). A sample of this data set is illustrated in Fig 4.

We use the synthetic data in order to evaluate data sets having spatial distributions that are difficult to handle using simple clustering algorithms (as will be shown in Sec. 4), and to control their size. The medical data set is used in order to check the algorithm for real-world applications where data are possibly highly dimensional. A motivation for using this data set is that fully supervised machine learning algorithms applied to automatically classify this data set have not shown high classification performance as will be discussed in Sec. 4.3. A further

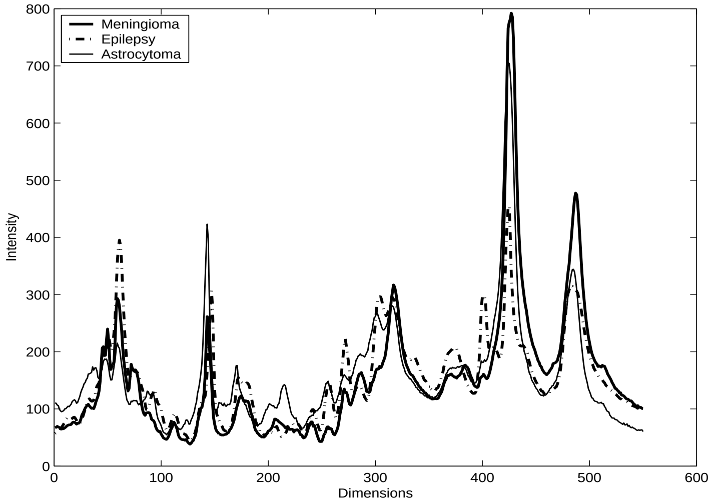


Figure 4. Magnetic resonance (MR) spectra of human brain neoplasms (3 data points, each belongs to a class)

motivation for using it is that the MR spectra are known to be an effective medical diagnostic tool. However, this efficiency depends on the interpretation (labeling) of the spectra themselves. It may happen that spectra are hard to label or are simply mislabeled due to the difficulty in distinguishing the spectra accurately, hence their relevancy as a typical application for the proposed algorithm.

As explained in Section 3, the algorithm consists of a two-step optimization process. The process of optimizing J, which requires the optimization of Q, is repeated iteratively for a number of iterations ($MaxIter$). In our experimental setting, $\beta = 0.06$ (for smooth optimization) and $MaxIter = 20$ (the clustering process converges at most after 20 iterations).

4.1. Discrimination Power of the Algorithm

It is interesting to see how good the algorithm is at discriminating classes from each other. This section shows how the scaling parameter α affects the linkage between clusters and classes in such a way that each of the generated clusters is homogeneous (pure), containing points from the same class only. To discuss the discrimination capability of the algorithm, let us consider the case where data are completely labeled and see how the algorithm behaves. First, to identify misclassified points in clusters, we rely on the label of the majority in each shared cluster (containing points with different labels). Points having a label

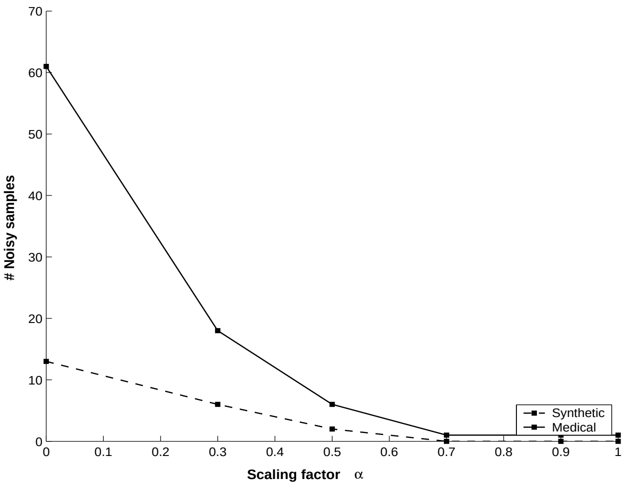
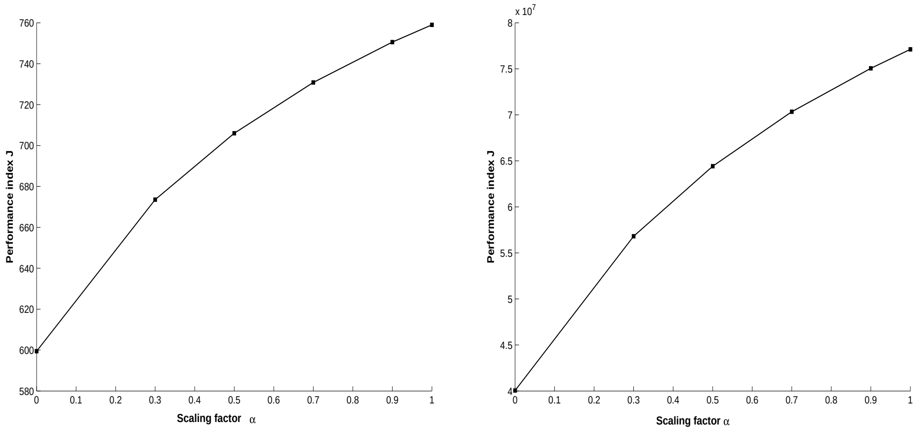


Figure 5. Effect of the scaling factor on the number of noisy points

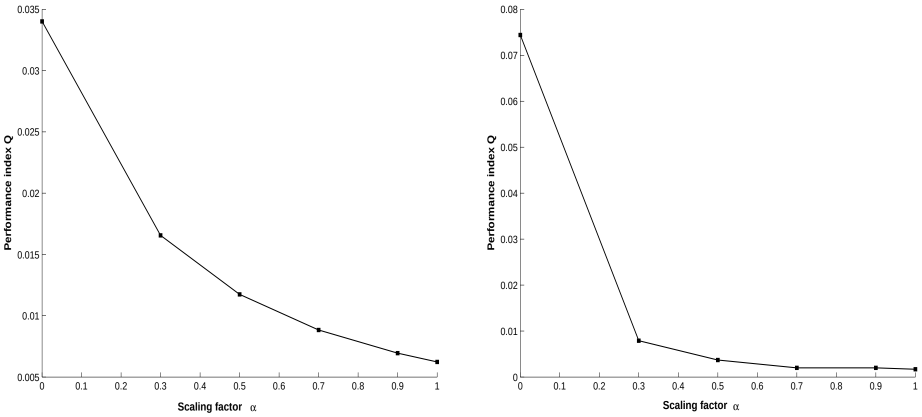


a. Synthetic b. Medical

Figure 6. Effect of the scaling factor on performance index J

different of that of the majority are considered misclassified (or noisy). Now, if we cluster the synthetic data into 6 clusters (4 for H_1, and 2 for H_2) and the medical data set into 6 clusters (2 clusters for each class), set the scaling factor α to various values: 0 (that is equivalent to the standard FCM), then 0.3, 0.5, 0.7, 0.9, and 1, and set $\beta = 0.06$ and $MaxIter = 20$, we obtain the results shown in Fig. 5.

When α is set to 0 (corresponding to FCM), the number of misclassified data points is 13 for the synthetic data and 61 for the medical data. As the scaling factor α increases, the number of misclassified points decreases until each of them is assigned to one of the clusters that fully belongs to the class having the same label. This means that

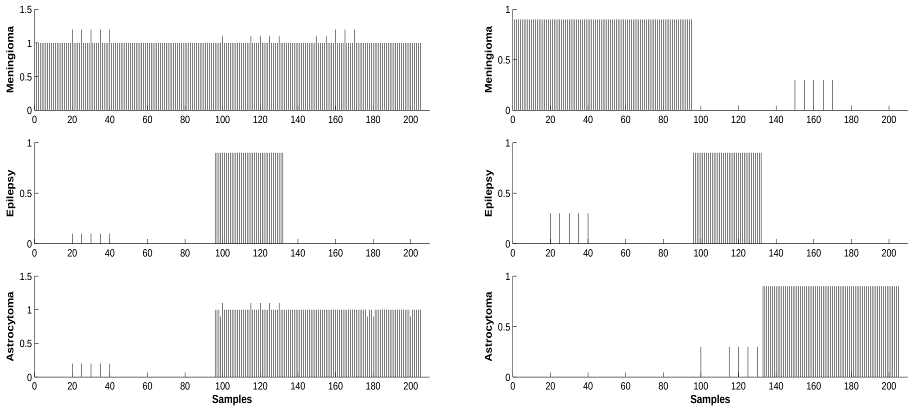


a. Synthetic b. Medical

Figure 7. Effect of the scaling factor on performance index Q

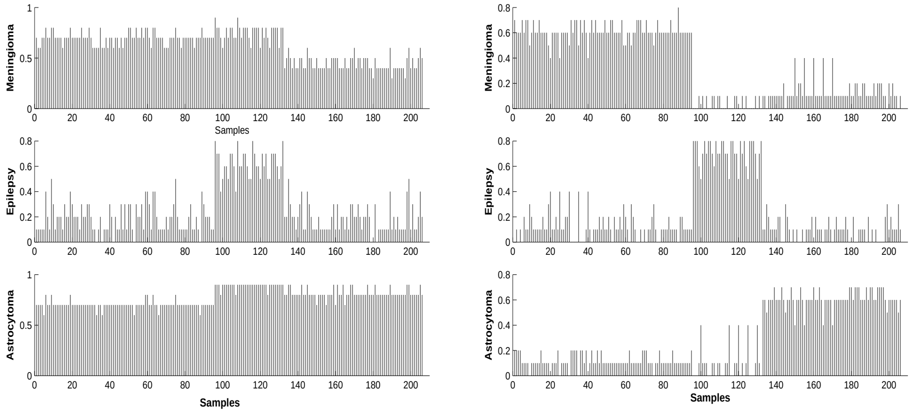
the number of misclassified points can be seen as a monotonically decreasing function of α (see Fig. 5). In other words, by assigning higher values to α, we are placing more confidence in the accuracy of the data labels. Furthermore, the first performance index J is a monotonically increasing function of α (see Fig. 6). Due to the existence of an external force (component 2 of Eq. 3), which aims at attracting data points to clusters whose prototypes become more central to the data with the same label, the distance increases between natural centers (i.e. those generated by the standard FCM) and those generated by combining both supervised and unsupervised components of Eq. 3. This force relates a data point with a given label to a cluster that is not necessarily close to it. The algorithm tends to put data points with the same label in the same container. Furthermore, the algorithm aims at discovering the real structure of the data. Non-adjacent clusters of the same class become labeled uniformly (with the same label). In contrast, the performance index Q decreases as the scaling factor α increases (see Fig. 7). As explained earlier, Q achieves its minimum when for all points k, the terms γ_{ik} (see Eq. 9) reach their minima. The membership degree of each point of a given class should be maximally spread among the clusters part of the same class. In other words, each point joins a pure cluster inside a class. Hence, reducing the number of misclassified points implies a decrease in Q.

On the other hand, the expression ψ_h (Eq. 14) can be used to illustrate the evolution of clusters in terms of label discrimination. In fact, this expression sums up the membership degree of a point k spread over all clusters i belonging to class h. A heterogeneous cluster is involved as many times in the computation of ψ_h as the number of distinct labels



a. $\alpha = 0.3$ b. $\alpha = 0.9$

Figure 8. Effect of the scaling factor on the evolution of $\tilde{U}$



a. $\alpha = 0.3$ b. $\alpha = 0.9$

Figure 9. Effect of the scaling factor on the evolution of U

contained in that cluster, i.e., if a cluster contains two different labels, it is considered twice: to compute the ψ_h values of points in H_1 and those of points in H_2. The matrix U in Eq. 14 can also be used instead of $\tilde{U}$, hence: $\zeta_h = \sum_{i \in \pi_h} u_{ik}$. Figure 8 illustrates the evolution of the overlap between classes during clustering for the medical data set. Note how the configuration of the histograms changes as α increases. Figs. 8b and 9b ($\alpha = 0.9$) show that ψ_h and ζ_h values of points from H_1 are much higher than those of the same points with respect to H_2. Consider points from class H_2 of the medical data. They have higher ψ_h and ζ_h values with respect to H_2 than with respect to H_1 and H_3 (see Figs. 8b and 9b). Likewise, the same observation applies for points from the other classes. Hence, points can easily be distinguished when $\alpha = 0.9$, i.e. when clusters become pure. As α increases, the ψ_h and ζ_h values of the points of a given class strengthen and those of points in the other classes weaken.

4.2. From Clustering to Classification

In this section, the proposed semi-supervised clustering algorithm is formulated as a classifier and its classification performance is discussed. To the best of our knowledge, the idea of using FCM-like clustering algorithms to build classifiers has never been studied. Here, we will use only labeled data since the primary goal is to validate the induced classifier which will be used later in the context of semi-supervised learning to observe the effect of both labeled and unlabeled data on its accuracy (see Sec. 4.4). The key issue in evaluating the performance accuracy in assigning data points coming from the testing pool to the correct cluster, that is, how often the classifier decision meets the actual assignment given a testing data set. The performance rate R is:

$$R = \frac{\text{Number of correctly assigned data points}}{\text{Size of the testing data set}} \tag{15}$$

An assignment is correct if the sample from the testing data is assigned to one of the clusters of the correct class (i.e. the target). To do that, the membership degree of the testing sample to clusters of the same class are summed up. The class to which the current sample has the higher membership degree is retained as a winning class (to be compared against the actual class of the sample). The testing process is portrayed in Fig. 10 and described by Alg. 2.

To evaluate the algorithm, for each of the data sets, n-fold cross validation testing is applied to measure the classification accuracy of the algorithm. That is, the data points are partitioned into n approximately equal-sized groups. The points in $n-1$ groups are used for training the classifier. The induced classifier is tested on the points in the hold-out group. This is then repeated for every other combination of $n-1$ groups. The overall accuracy is the average of the n computed accuracy values. Here, a 5-cross validation is used.

To validate the obtained classification results, we use the paired t-test of statistical significance which checks if the average of the change between two observations differs statistically and significantly from zero. To perform this test, the differences d_i between the paired observations, i $(i = 1, .., m)$, are evaluated in order to compute their mean $\overline{d}$. These paired differences should be normally distributed. Here, we use relatively large samples $(m = 30)$ drawn from the same interval

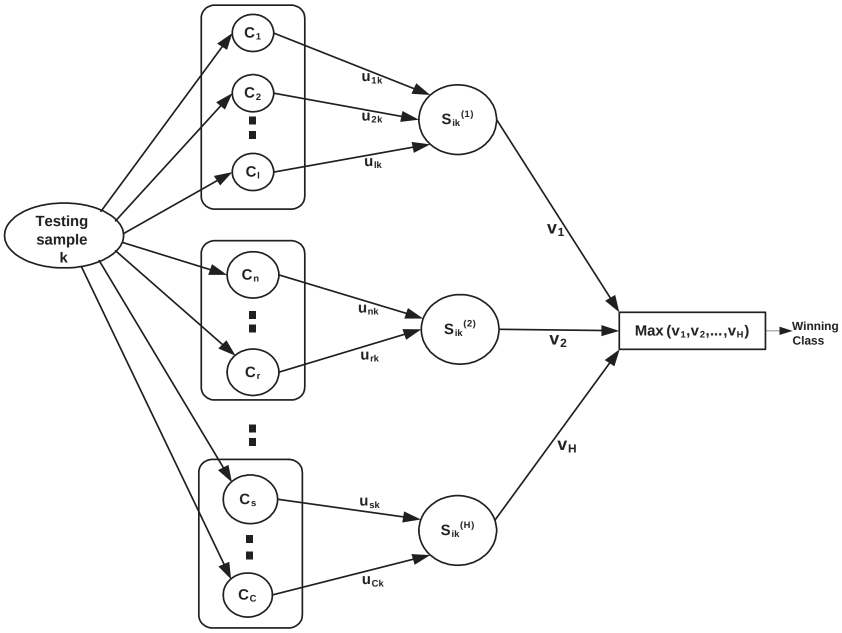


Figure 10. Using the clustering algorithm as a fuzzy classifier ($S_{ik}^{(p)} = \sum_{i \in \pi_p} u_{ik}$)

Algorithm 2 : Using the clustering algorithm as a fuzzy classifier

```
for all data points coming from the testing pool do
    a. Given a data point $k$ whose actual class is $act_k$, compute its
       membership degree $u_{ik}$ to each cluster $i$ of each class $h$.
    b. For each class $h$, compute the amount:
           $\zeta_{hk} = \sum_{i \in \pi_h} u_{ik}$
       that is the sum of the membership degrees of $k$ to class $h$.
    c. To determine the winning class $win_k$, compute:
           $win_k = Argmax(\zeta_{hk}) \quad h = 1, ..., H$
    d. If the winning class $win_k$ is the actual class of the testing data
       point $k$, i.e. $act_k = win_k$, then $S = S + 1$
end for
Compute the performance rate:
    $R = \frac{S}{|\text{testing data set}|}$
```

scale [0,1] and are randomly drawn (from a sampling point of view), hence it is reasonable to admit the normality condition for the means. In our context, the t-test aims at showing whether a given classification algorithm A performs better than another algorithm B. Therefore, the alternative hypothesis is:

$$H_A : \mu_d > 0$$

to be compared against the null hypothesis that is $H_0 : \mu_d \leq 0$. Clearly, we are interested in one direction test that is the right-tailed test. To perform it, we need to compute the statistic T as follows:

$$T = \frac{\overline{d}}{S_d/\sqrt{m}}$$

where S_d is the standard deviation of the differences. This statistic serves to make a decision about the hypothesis acceptance. If T exceeds a certain critical value defined by $t_{\theta,\ m-1}$, the null hypothesis is rejected and the alternative hypothesis, H_A, is satisfied. The amount $t_{\theta,\ m-1}$ is a t-test tabled value parameterized by the number of degrees of freedom (the number of pairs -1) and a given significance level $\theta = 0.05$[1]. In our case, $t_{\theta,\ m-1} = t_{0.05,\ 29} = 1.699$. Rejecting H_0 means that the performance of A being higher than that of B is not due to chance only, but rather it reflects the efficiency of A (Mason et al., 1983; Snedecor and Cochran, 1989).

We run the algorithm using various values of the scaling factor α (i.e. 0.3, 0.5, 0.7, 0.9, and 1) on different combinations of clusters. Table II shows the results obtained. Each cell of Tab. II indicates the success rate based on data points assignment coming from the testing pool to the correct cluster. The results illustrate that the algorithm achieves a very reasonable performance. Lower classification performance occurs when α is set to small values. For such cases, the number of misclassified samples during the training phase is significantly high. In contrast, when the number of misclassified points is small, the performance remains high and depends on the closeness of the testing points to prototypes.

As explained in Sec. 3, π_h is the set of clusters belonging to class h (see Eq. 8). Based on the labels of data points in each cluster, π_h is determined. Impure clusters will be part of more than one class. This will influence the computation of ζ_h, part of step (b) of Alg. 2. If, for instance, a cluster i is shared by two classes h and h', then both ζ_h and $\zeta_{h'}$ of a given data point k will involve the membership of k to the cluster i. Numerically speaking, for instance in the case of synthetic

[1] Due to name conflict, θ is used instead of α that is referred to as the significance level in the related literature.

Table II. Classification performance of the algorithm on the synthetic data

α		0.3	0.5	0.7	0.9	1
Combinations	(1, 2)	0.66	0.66	0.66	0.66	0.66
	(2, 2)	0.33	0.33	0.91	0.92	0.93
	(2, 4)	0.86	0.94	0.84	0.85	0.96
	(3, 2)	0.87	0.87	0.87	0.87	0.95
	(3, 3)	0.86	0.87	0.87	0.87	0.87
	(3, 4)	0.85	0.86	0.86	0.86	0.86
	(4, 2)	0.91	0.91	0.91	0.96	0.96

Table III. Classification performance of the algorithm on the medical data

α		0.3	0.5	0.7	0.9	1
Combinations	(2, 1, 2)	0.66	0.41	0.69	0.69	0.69
	(1, 1, 3)	0.37	0.37	0.37	0.50	0.51
	(2, 1, 1)	0.50	0.61	0.61	0.61	0.61
	(2, 2, 2)	0.38	0.51	0.68	0.69	0.69
	(2, 2, 3)	0.38	0.51	0.53	0.68	0.69
	(3, 2, 2)	0.61	0.61	0.61	0.61	0.63
	(4, 2, 2)	0.61	0.61	0.61	0.61	0.61
	(4, 2, 3)	0.50	0.61	0.63	0.68	0.68

data with the combination (1, 2) and $\alpha = 0.3$, we have $\pi_1 = \{1\}$ and $\pi_2 = \{1, 2, 3\}$. Clusters 2 and 3 are pure clusters of class 2, but cluster 1, while dominated by points from class 1, contains 7 data points from class 2. Now, for a given data point k from class 1, ζ_{1k} is the membership of k to cluster 1 (since class 1 consists of only cluster 1), while for the same point k, ζ_{2k} will be the sum of the membership degrees of k to clusters 1, 2, and 3. Therefore, executing step (c) of Alg. 2 will lead to a misclassification. Consider the case of data point 3 which has a label '1', its membership to the clusters can be represented as a fuzzy set as follows: $\{0.63/C_1, 0.25/C_2, 0.12/C_3\}$. This results in $\zeta_{13} = 0.63$ and $\zeta_{23} = 0.63 + 0.25 + 0.12 = 1$. Therefore, the data point 3 from class 1 is assigned to class 2 although its highest membership degree (0.63) is to cluster 1 which is dominated by class 1. As a conclusion, lower classification performance results from impure clusters which contain data points from more than one class.

To tackle this problem, we suggest to find the dominant class for each cluster first. Then, given a data point k, we consider the winning

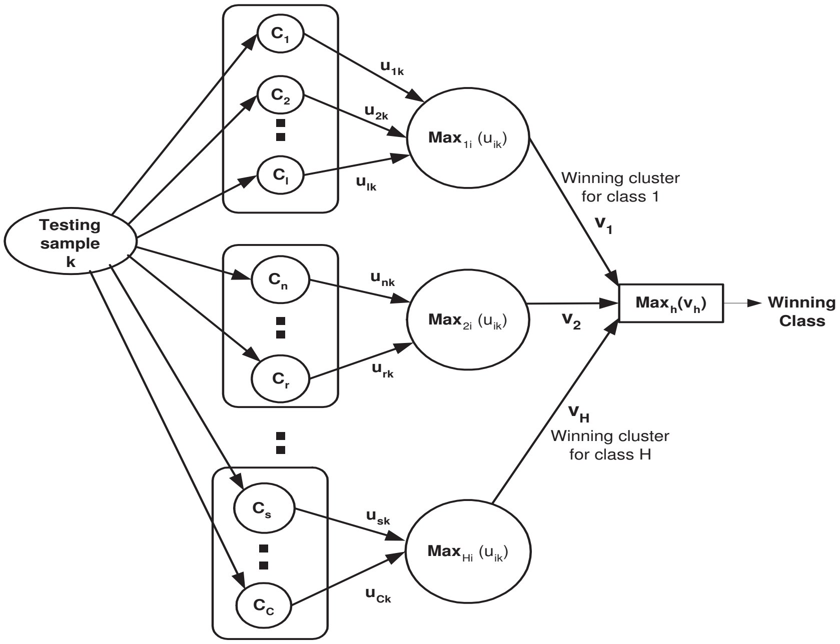


Figure 11. Second version of the classifier

cluster, i.e. the one for which k has the highest membership degree for each class. By comparing the resulting degrees, the winning class will be the one corresponding to the highest degree (see Fig. 11). For instance, in the case of $k = 3$ for $\alpha = 0.3$ and a combination (1, 2), we will have in the first (dominance) stage $\pi_1 = \{1\}$ and $\pi_2 = \{2, 3\}$. The winning cluster from class 1 is cluster 1 (u_{13} =0.63) and the winning cluster from class 2 is 2 (u_{23} =0.25). Comparing these membership values, class 1 is the winner and therefore, the data point 3 will be correctly assigned.

The solution suggested here is a two-stage competition process: intra-class competition and inter-class competition. The first level of competition allows us to determine the winning cluster for each class, while the second competition level allows us to find the winning class.

After running the new version of the fuzzy classifier on the same synthetic and medical data with the same combinations, we obtained the results displayed in Tabs. IV and V. By comparing these results with those obtained with the first version of the classifier (see Tabs. II and III), it is worth noticing that the latter version improves the results and in the worst case, performs as well as the first version. This can be checked using the t-test. To perform that, we will consider two values for the tuning parameter α: 0.3 and 1. These are the extreme values in both

Table IV. Classification performance of the algorithm on the synthetic data

α		0.3	0.5	0.7	0.9	1
Combinations	(1, 2)	0.88	0.88	0.88	0.88	0.88
	(2, 2)	0.92	0.93	0.93	0.94	0.94
	(2, 4)	0.88	0.97	0.97	0.97	0.97
	(3, 2)	0.98	0.98	0.98	0.98	0.98
	(3, 3)	0.91	0.94	0.94	0.96	0.96
	(3, 4)	0.85	0.86	0.86	0.86	0.86
	(4, 2)	0.97	0.97	0.97	0.97	0.98

Table V. Classification performance of the algorithm on the medical data

α		0.3	0.5	0.7	0.9	1
Combinations	(2, 1, 2)	0.66	0.46	0.75	0.75	0.75
	(1, 1, 3)	0.49	0.49	0.49	0.60	0.61
	(2, 1, 1)	0.50	0.61	0.61	0.61	0.61
	(2, 2, 2)	0.48	0.57	0.68	0.72	0.73
	(2, 2, 3)	0.42	0.53	0.58	0.70	0.70
	(3, 2, 2)	0.63	0.66	0.66	0.66	0.66
	(4, 2, 2)	0.61	0.61	0.61	0.63	0.63
	(4, 2, 3)	0.58	0.61	0.66	0.70	0.70

tables. Recall that lower values of α assign more weight to the clustering component (see eq. 3), while higher values of α allow to set a balance between the clustering and the supervised components of the objective function. With these two values, we will have to consider $7 \times 2 = 14$ classification values (all cluster combinations) for the synthetic data set and $8 \times 2 = 16$ classification values for the medical data set.

To apply the t-test in order to measure the statistical significance of those classification results, cross-validation using 30 runs is performed. In each run, the data is randomly shuffled and the cross-validation is performed using the two versions of the algorithm, v_1 and v_2, on a specific parameter combination, that is (number of clusters per class, α) as shown, for instance, in Tab V. In other words, for each of the $14 + 16 = 30$ classification values, a vector of 30 new realizations is computed (# realizations per classification value = # runs, i.e., 900 realizations for each version of the algorithm). In all, we will compare 14 pairs of realization vectors of length 30 for the synthetic data and 16 pairs of vectors for the medical data set.

Table VI. Statistical evidence: version 2 vs. version 1

α		0.3	1
Data set	Combination	$T_{0.3}$	T_1
Synthetic	(1, 2)	10.6358	10.6921
	(2, 2)	13.5941	4.9399
	(2, 4)	3.7863	1.9153
	(3, 2)	6.8311	3.2596
	(3, 3)	5.7505	7.9706
	(3, 4)	0.4947	0.4012
	(4, 2)	5.1220	4.7883
Medical	(2, 1, 2)	4.7883	3.7503
	(1, 1, 3)	10.9495	10.8384
	(2, 1, 1)	0.1261	1.9326
	(2, 2, 2)	8.1982	4.9203
	(2, 2, 3)	5.1737	1.7034
	(3, 2, 2)	3.8373	4.3746
	(4, 2, 2)	0.5023	2.9555
	(4, 2, 3)	8.6732	3.6740

The results of the t-test are displayed in Tab VI. They show a strong evidence that the performance of version 2 of the algorithm is higher than that of version 1. The statistic T compared with the $t_{0.05,\ 29} = 1.699$ indicates clearly that the null hypothesis is rejected in favor of the alternative hypothesis that is "version 2 is better than version 1". However, there are few cases where the null hypothesis is not rejected, e.g. the combination (3, 4) with the synthetic data and the combinations (2, 1, 1), (4, 2, 2) with the medical data as α is set to 0.3. With these cases changing the way of inducing a classifier did not help. In other words, the evidence about the membership of the unseen (testing) samples gathered from such cluster combinations using both algorithm coincides (winner using sum of membership degrees $\cong$ winner using maximum of the membership degrees). Except these three cases, the test shows statistical evidence that version 2 of the algorithm outperforms version 1.

It is also worthwhile to mention that for some combinations, the scaling factor α does not have any effect on the classification performance (e.g. in the case of (1, 2), Tab. II). This is reflected not only in the fact that the number of misclassified data points is the same for all values of α, but also in the fact that the misclassified data points are the same. This means that the algorithm was not able to classify those points correctly. However, if we look at other cases, we notice that as the purity of clusters increases, the classification rate increases or at least remains the same.

4.3. Comparative Analysis of the Classification Performance

This section is concerned with comparing the proposed fuzzy semi-supervised classifier against two fully supervised algorithms: multi-layer perceptron (MLP) and support vector machines (SVM). The goal of this comparison is to validate the clustering-based classifier proposed in the previous section in order to give solid ground to this classifier before using it in the context of semi-supervised learning as will be presented in the next section.

The fully supervised models used in the comparison are powerful for solving nonlinear classification problems. MLPs (Bishop, 1995) are the most popular neural networks model used to approximate any function. They use soft hyperplanes for discrimination. An MLP consists of a number of layers, each layer consisting of a number of nodes. Each node computes its activation level using various activation functions. In this experiment, we apply the sigmoid function (sigm) and the hyperbolic tangent sigmoid function (Tanh). The number of layers, the number of nodes per layer and the learning rate are parameters.

On the other hand, we will apply least square SVMs which are a reformulation of the standard linear SVMs (Suykens and Vandewalle, 1999). The non-linearity is introduced through the use of kernels allowing non-linear mapping. In addition, the cost function is a regularized least squares function with equality constraints that is optimized by iterative methods. In this work, three kernel functions are used: linear ($K(x, y) = x^t \cdot y$), polynomial with degree d: $K(x, y) = (x \cdot y + 1)^d$, and radial basis function: $K(x, y) = exp\left(-\frac{||x-y||^2}{2\sigma}\right)$.

To compare the classification performance of the semi-supervised algorithm against the fully supervised algorithms in an objective way, we use the medical data set. Table VII illustrates results obtained using some neural architectures. Each cell indicates the performance rate with respect to an architecture (with a given number of layers and a given combination of activation functions). The number of layers, which does not include the input layer, and the number of neurons of the hidden layers are shown in the table. In the second column, the first activation function is used in the hidden layers while the second is used in the output layer. The classification performance of SVMs on the medical data set is illustrated in Tab. VIII.

Table VII. Classification performance using MLP nets

Hidden & Output Layers		2	3	4	5
Number of Nodes		100-3	275-100-3	275-100-50-3	275-200-100-50-3
Activation	tanh+sigm	0.59	0.65	0.75	0.68
	tanh+tanh	0.45	0.47	0.55	0.49
	sigm+tanh	0.47	0.81	0.77	0.68
	sigm+sigm	0.85	0.81	0.78	0.78

Table VIII. Classification performance using SVMs

Kernel function	Classification rate
Linear	0.45
Nonlinear: Polynomial	0.63
Nonlinear: Radial basis function	0.59

Considering the best results from Tabs. III, V, VII, VIII, we find that the highest classification performance, which is 85%, is obtained using a 3-layer architecture MLP (including the input layer). With SVMs, the classification rate is low and the best value is 63%, obtained with a polynomial kernel function. With the fuzzy classifier stemming from the suggested algorithm, the highest value for the first version of the classifier is 69%, while with the second version it is 75%.

These experiments show that the fuzzy semi-supervised algorithm is competitive with fully supervised classification algorithms. In fact, it outperforms SVMs and provides a lower classification performance than that of the MLP. To measure the statistical significance of the obtained results, the t-test is applied on the cross-validation using 30 runs. In each run, the data is randomly shuffled and the cross-validation is performed using the three algorithms. As portrayed in Tab. IX, the test shows that there is statistical evidence that the performance of the proposed algorithm is higher than that of SVM (T=6.9577> $t_{0.05,\ 29}$=1.699) and therefore, the null hypothesis is rejected in favor of H_A. The test also shows that the performance of MLP is better than that of the proposed algorithm (with left-tailed test, T=3.1356).

As a concluding remark, the advantage of using the proposed algorithm is that there are not many tuning factors. Except the scaling factor, the other parameters like the training rate and the number of iterations are not that much crucial. In contrast, with fully supervised

Table IX. Statistical evidence: SSC vs. MLP and non-linear SVM

Algorithms	Statistic (T)
SSC vs. SVM	6.9577
SSC vs. MLP	-3.1356

algorithms, especially MLPs, various parameters have to be adjusted as discussed earlier and shown in Tab. VII.

4.4. Semi-supervised Clustering

After finding the optimal structure of the classifier derived from the proposed clustering algorithm and discussing its classification performance when the whole data is labeled, this section will focus on clustering/classification of partly labeled data.

Each of the data sets is split into two parts: a labeled set and a virtually unlabeled set. In our experimental setup, the labeled set is randomly chosen and varies from 10%, 30%, 50% to 70% of the whole data. The remaining part of the data is dynamically and randomly divided into two sets: “used” and “temporarily not used”. Because a 5-cross validation split is applied, the labeled and the unlabeled data to be used serve to train and test the classifier. We will also vary the amount of virtually unlabeled (to be effectively used) from 0%, 10%, ... to 100%. It is worth mentioning that in our data split strategy, we have preserved the uniformity, so that no class is omitted during the training phase. As to the parameter setting, each class for both data sets will be represented by 2 clusters, the scaling parameter α is set to 1, the learning rate β is set to $= 0.06$, and $MaxIter = 20$. Because we aim at combining labeled and unlabeled data to train the algorithm, in this experiment (but also in all experiments in the rest of this paper) α is set to 1, which means that the provided labels of the labeled data are completely trustful and accurate (see Sec. 4.1).

Varying the ratio of labeled and unlabeled data, we can observe the effect of increasing both labeled and unlabeled data on the accuracy of the classifier. The results obtained for different ratios of the labeled and the virtually unlabeled samples from the synthetic and the medical data sets are shown in Figs. 12 and 13. Each of the figures shows the evolution of the classification accuracy as the ratio of both labeled and unlabeled data increases. Clearly, the classification performance increases as the size of unlabeled data increases. However, the improvement of the accuracy depends on the size of the involved labeled data.

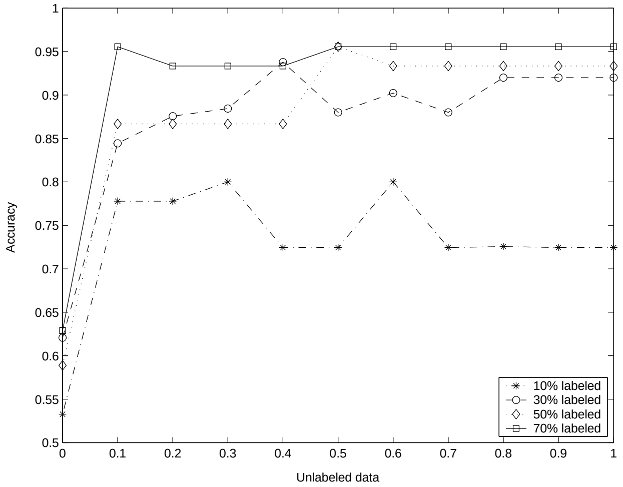


Figure 12. Learning with partially labeled data - Synthetic data

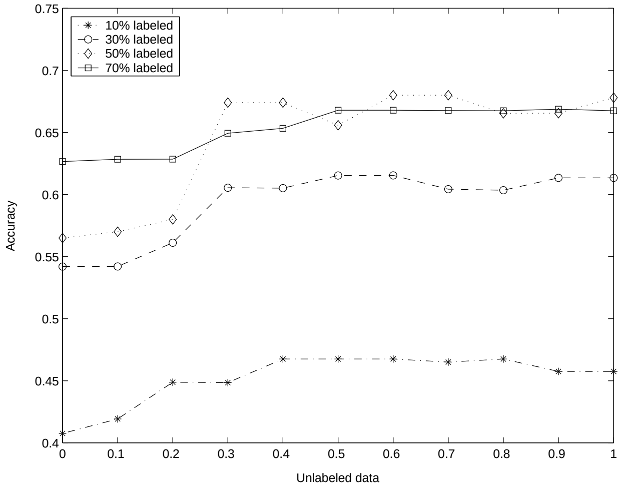


Figure 13. Learning with partially labeled data - Medical data

In fact, for the synthetic data set, the classification accuracy of the proposed algorithm is 53.27% when 10% of the data is labeled and no unlabeled samples are used. This performance increases as more labeled data is involved achieving a level of 62.89% (with 70% labeled data). Similar effect is observed with the medical data where the accuracy goes up from 40% to 62.66% as more labeled data is used. Thus, increasing the size of labeled data does improve the accuracy of the classifier.

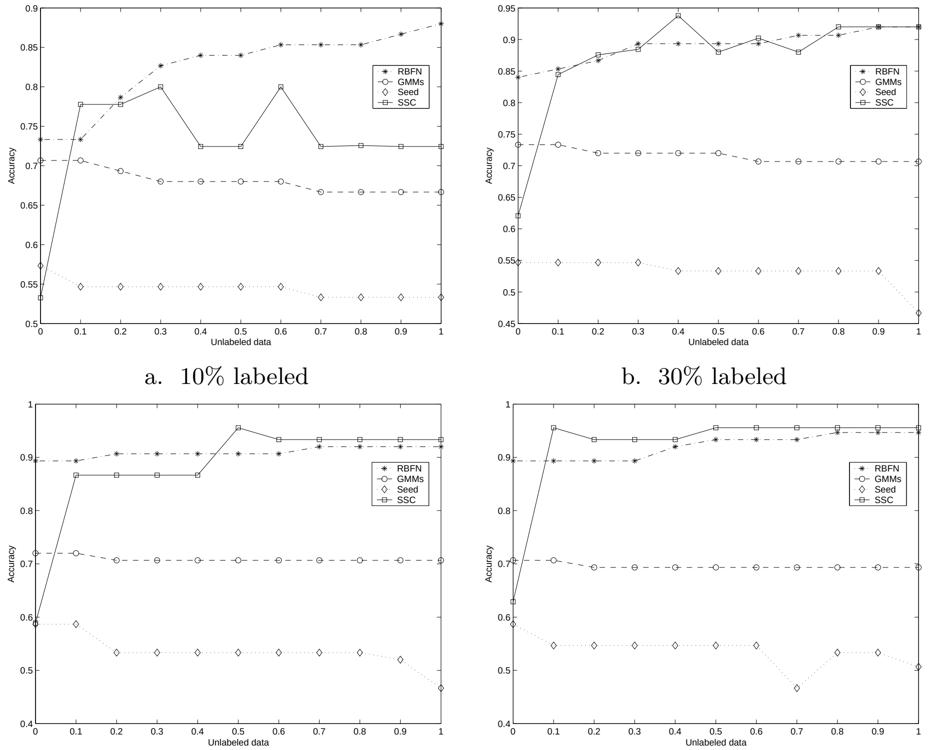
On the other hand, using unlabeled data to train the classifier improves the accuracy too. For instance, for the medical data set an improvement of 5% is achieved (going up from 40.76% to 45.75) when only 10% labeled data is used and the rest is unlabeled. More improvement (11.3%) can be achieved when 70% of the data is labeled. One can easily conclude that not only the labeled data improves the accuracy but also the unlabeled one does. Similar conclusion can be drawn from the application of the algorithm on the synthetic data, where a clear contribution of the unlabeled data is observed (32,67% improvement when 70% of labeled data is used and 19,17% improvement when only 10% of the data is labeled). It is then worth stressing that the classification performance of the proposed algorithm can be boosted by both labeled and unlabeled data.

4.5. Comparative Study of Some Semi-supervised Learning Algorithms

In this section, we will compare the classification performance of the suggested algorithm against three methods (Bouchachia, 2005b): Radial basis function network (RBF) based on active learning proposed by Bouchachia (2005a), seed-based fuzzy clustering which is a variation of the method proposed by Basu et al. (2002), and the Gaussian mixture models based (GMMs) on expectation maximization (EM) which is an extension of the method proposed by Nigam et al. (2000).

The method *RBF neural network based on active learning* relies on active selection of unlabeled samples to be used together with the originally labeled ones to induce a neural classifier. Because, RBF nets are trained using a supervised learning rule, it is necessary to devise a mechanism to allow using unlabeled data to train these networks. In a preprocessing step (which corresponds to sample selection), the label of some unlabeled samples is estimated. To do that, a supervised clustering method has been proposed to generate labeled clusters. The centers of these clusters are used as seeds to initialize the FCM algorithm. The unlabeled samples are then clustered around these seeds. Once, the new clusters are generated, only prototypical samples are selected and are assigned the label of the cluster to which they belong. They are then used together with the originally labeled samples to train the RBF classifier, while the centers of the clusters are used as radial basis functions of the RBF classifier (Bouchachia, 2005a).

Similar to the preprocessing step of the previous method, the *seed-based clustering* method uses the labeled data to generate a seed for each class (Bouchachia, 2005b). The unlabeled samples are then assigned to clusters based on their similarity to the clusters' prototypes. These prototypes are then updated accordingly. In other words, these seeds are used as initial prototypes to guide the process of clustering unlabeled samples. To test this algorithm, the distance of the test-



a. 10% labeled

b. 30% labeled

c. 50% labeled

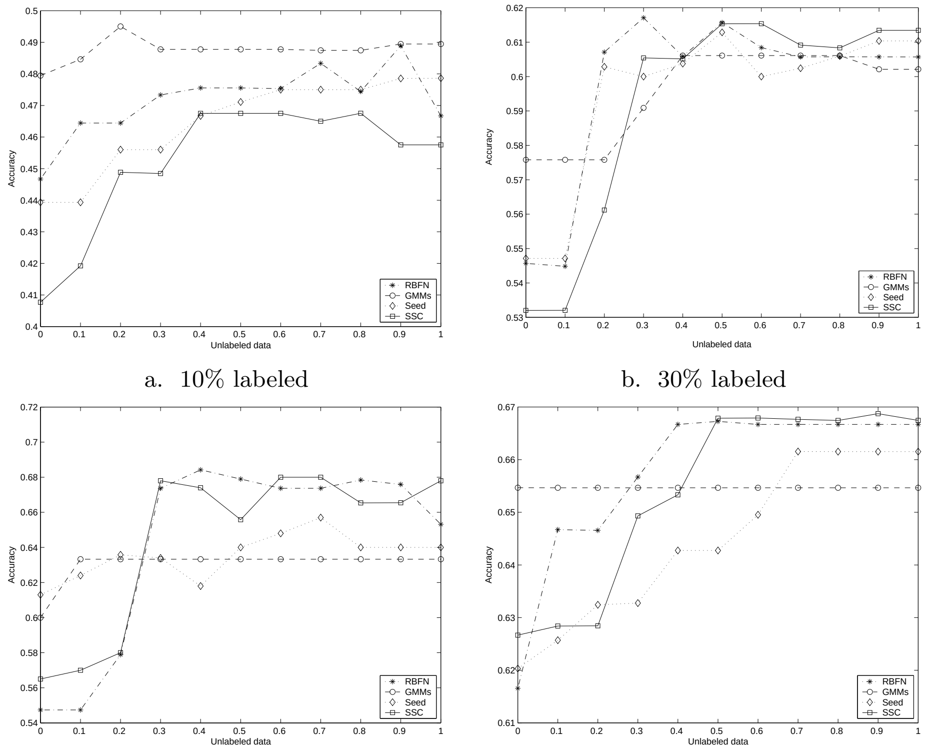
d. 70% labeled

Figure 14. Comparative study: partially labeled synthetic data

ing samples to the generated prototypes is computed and the winning cluster (class) is then determined (i.e., that with the min distance).

The third algorithm is an *Expectation Maximization (EM)-based classifier*. The basic assumption is that the data can be represented as a mixture of Gaussians, where each Gaussian represents a class. Initially, the labeled data is used to estimate the characteristics - mean and variance - of the Gaussians. In a second stage, the unlabeled data is used to retrain the classifier using the EM method which consists of two steps as follows. In the E-step, the labels of the unlabeled data are probabilistically estimated. Then, in the M-step, both the originally labeled data and the pre-labeled data are used to re-adjust the class characteristics. This process is repeated until convergence. During testing, the membership probability of the testing samples to each of the Gaussians is computed and the winner one is found.

Figures 14 and 15 portray the results of the algorithms: the proposed semi-supervised clustering algorithm (SSC), RBF networks based on active learning, seed-based clustering, and GMMs based on EM. It is



a. 10% labeled

b. 30% labeled

c. 50% labeled

d. 70% labeled

Figure 15. Comparative study: partially labeled medical data

worthwhile to mention once again that the motivation of using the synthetic data is to show complicated situations where the structure and the distribution of samples from the same class do not lie contiguously in the same region of the space (see Fig. 3), hence the difficulty of obtaining good results on this data. This difficulty is first experienced with the seed clustering algorithm whose performance accuracy deteriorates when unlabeled data is added to the training pool (see Figs. 14a, 14b, 14c, 14d). With 10% to 70% labeled data, negative improvement has been achieved (-4%, -8%, -12%, and -8%). Similar results but no that much worse have been obtained using the EM-based GMMs approach. Indeed, the improvement is -4%, -2.66%, -2.67%, and -1.34%. These results are worse because with both approaches, the assumption is that each class is represented by one component (cluster). Here class two, H_2, is split into two clusters separated by samples of the first class, H_1 (see Fig. 3). The initial center (i.e., called seed in the seed-based approach and mean in the GMM approach) of H_2 falls in the region occupied by H_1. Because unlabeled samples may emanate from both clusters of H_2 (i.e. sides of class H_1), the seed remains in that "unacceptable" region. Hence, the accuracy of both classifiers cannot increase. In summary, representing a class with one center does not help when the samples of the same class are spread over different regions.

However, the active learning-based RBFNs approach is able to cope with the synthetic data. Indeed, the accuracy increases as more unlabeled data is involved. An improvement of 14.67%, 8%, 2.67%, and 5.34% has been achieved for different ratios of labeled data (i.e. 10% to 70%). The active learning-based RBFNs algorithm is capable of dealing with partially labeled data because each class can be represented by more than one center. These centers are actually radial basis functions of the network (Note that radial basis functions are Gaussian functions defined by a center and a spread). By tuning the spread of each Gaussian, we can cover the whole space. Hence, active selection of samples around these centers can be very useful for the accuracy of the classifier.

Better results are obtained using the proposed semi-supervised algorithm. Indeed, the improvement of the accuracy when unlabeled data is added to the training set is 19.17%, 29.93%, 34.44%, and 32,67% respectively. However when only the labeled data is used, the level of accuracy is lower than that of the active learning-based RBFNs algorithm. The best results on the synthetic data set has been achieved by the semi-supervised clustering algorithm with 95.56%. As a conclusion, the performance of the first two approaches, seed-based and EM-based, depend more on the structure of the data and the unlabeled data can worsen the accuracy. The last two approaches, active learning-based RBFNs and semi-supervised clustering, have shown their ability to improve the accuracy independently of the structure of the data.

To check the statistical significance of these results, the t-test is applied on the cross-validation means using 30 runs. In each run, the data is randomly shuffled and the cross-validation is performed using the four algorithms. But, given the large number of combinations we used (different ratios of labeled and unlabeled data), we will consider three representative cases: (1) the amount of labeled data is smaller than that of unlabeled (30% labeled and 70% unlabeled), (2) the amount of labeled and unlabeled data is the same (50% labeled and 50% unlabeled), and (3) the amount of labeled data is larger than that of unlabeled data (70% labeled and 30% unlabeled). As portrayed in Tab. X related to the synthetic data, the test shows that there is a strong evidence that seed-based, EM-based GMMs, and active learning-based RBFNs methods perform worse than the proposed algorithm (SSC). The T statistics clearly justify that there is only one situation (30% of the data is labeled) where there is no evidence that SSC performs better

Table X. Statistical evidence (synthetic data): SSC vs. other methods

Lab-Unlab	30%-70%	50%-50%	70%-30%
SSC vs. RBFNs	-1.8284	1.7319	1.7485
SSC vs. GMMs	7.3074	14.9539	18.3172
SSC vs. Seed	18.2784	19.4877	18.5365

than RBFNs (T= $-1.8284 < t_{0.05,\ 29}$=1.699). Therefore, the obtained classification results are statistically significant.

On the other hand, the evaluation of these approaches on the medical MR data has shown that by increasing the amount of labeled data, the accuracy of these approaches increases (see Fig. 15a, 15b, 15c, and 15d). Furthermore, unlabeled data does also improve the accuracy. Of course, the amount of improvement depends on the method used. Indeed, the highest improvement is achieved by the SSC algorithm when 10% and 30% of the data is labeled. Close improvement levels are achieved by the same algorithm and by the active learning-based RBFNs algorithm when 50% and 70% of the data is labeled. However, seed-based and GMMs approaches have performed better on the MR data compared with the synthetic data though the accuracy of GMMs in presence of unlabeled data has not been improved when 70% of the data is labeled.

As to the statistical significance of the obtained results on the medical data, the t-test is again applied using 30 runs with randomly shuffled data on three combinations of labeled and unlabeled ratios of data (30%-70%, 30%-70%, 30%-70%). Table XI illustrates that in the case of 30% labeled, the t-test does not provide evidence that SSC outperforms GMMs and RBFN since the statistic $T < t_{0.05,\ 29}$=1.699. But, SSC does outperform the seed-based clustering algorithm ($T > t_{0.05,\ 29}$). With 50% labeled, SSC is better than the other algorithms (higher values of T in favor of the alternative hypothesis H_A). With 70% labeled, the test shows that SSC still outperforms GMMs (T=2.7798), seed-based clustering (T=2.7649) but not RBFNs (T=0.7469). Therefore, the obtained classification results shown in Figs. 15b, 15c, and 15d are statistically verified. In addition, in some cases (e.g., 50% labeled data), the test has shown the superiority of the SSC algorithm which was not clear from Fig. 15c.

In summary, the proposed semi-supervised clustering algorithm is able to cope with the problem of learning from labeled and unlabeled data even when the structure of data is difficult to handle. Furthermore, the experiments have shown that both labeled and unlabeled data have an impact on the accuracy of the proposed classifier.

Table XI. Statistical evidence (medical data): SSC vs. other methods

Lab-Unlab	30%-70%	50%-50%	70%-30%
SSC vs. RBFNs	-0.6621	4.7926	0.7469
SSC vs. GMMs	-1.0158	4.6599	2.7798
SSC vs. Seed	4.0153	5.7778	2.7649

4.6. Behavior of the algorithm

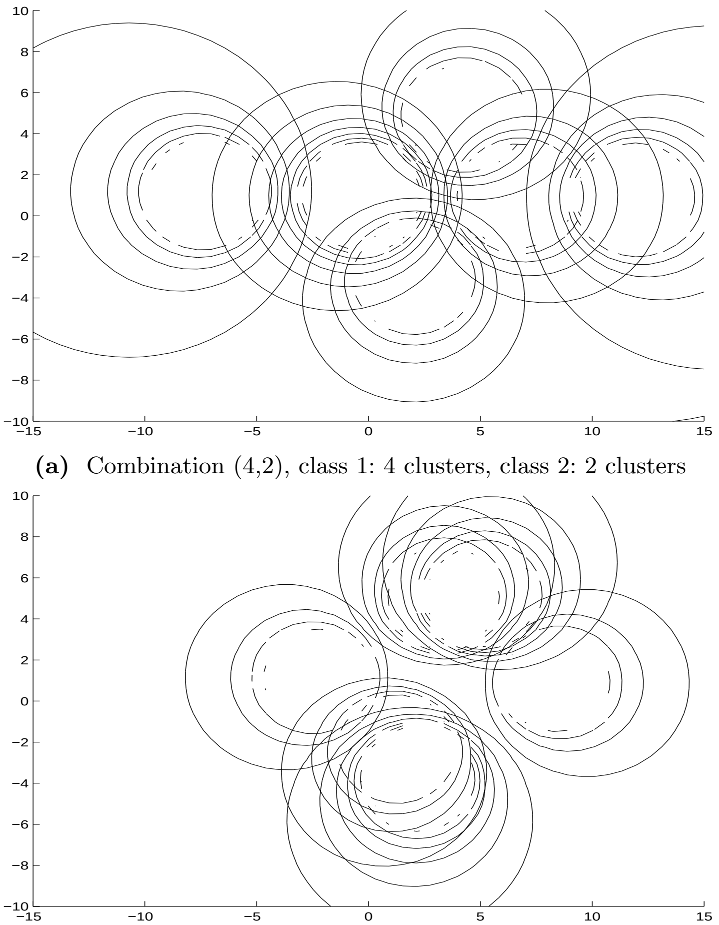
It is usually of interest to find the optimal structure of the data (i.e. the optimal number of clusters). This is generally done using some validity measures like entropy, partition coefficient, separability index and many other indices. In our context, we seek to cluster each class into a certain number of clusters in such a way that the content of each cluster becomes pure. This is done by finding the combination of clusters that yields the smallest value of the objective Q. It is also worth stressing that the current approach is preferable to one where classes are clustered separately because it addresses the problem of noisy data. If we cluster each class separately, we have no opportunity to deal with the aspect of detecting noisy points (mislabeled and misclassified) especially when the data has a hidden cluster structure and some clusters possibly contain data points with different labels (this aspect of noisy points detection is not discussed here).

Moreover, the algorithm is able to follow a pre-specified cluster combination. To show that, we rely on the synthetic data because it helps visualize the behavior of the algorithm. For example, Fig. 16b shows the contour plot for the combination (2, 4) (i.e. 2 clusters for H_1 and 4 for H_2). Such a combination cannot be addressed using FCM due to the structure of this data. There are a few exceptions where the algorithm does not follow the pre-specified cluster combination, i.e., when too many misclassified points result from the clustering process. By giving the data analyst the possibility to specify the number of clusters per class, we aim not only at involving more control in the behavior of the algorithm but also at making the algorithm more adaptive.

4.7. Relationship between classes and clusters

We can also use a linear regression model to estimate the strength of the relationship between classes and clusters. The linear regression model can be expressed as:

$$F = A\tilde{U} \tag{16}$$



(a) Combination (4,2), class 1: 4 clusters, class 2: 2 clusters

(b) Combination (2,4), class 1: 2 clusters, class 2: 4 clusters

Figure 16. The evolution of clusters

where $F = [f_{hk}]$ is defined as explained earlier. Equation 16 is a formulation of Eq. 9, part of the Q expression in Eq. 8. The aim is then to determine the matrix $A(H, C)$ that represents the regression parameters, where H is the number of classes (rows) and C is the number of clusters (columns). The form of the residuals is given as:

$$e_k = F_k - A\tilde{u}_k \tag{17}$$

The goal is to find A having e_k minimized; the residual sum of squares can be written as:

$$Q = \sum_{k=1}^{N} e_k^t e_k = \sum_{k=1}^{N} (F_k - A\tilde{u}_k)^t (F_k - A\tilde{u}_k) \tag{18}$$

Table XII. Regression parameters for the combination (2, 3)

		C_1	C_2	C_3	C_4	C_5
$\alpha = 0.3$	H_1	1.0015	1.0014	-0.0021	-0.0010	-0.0041
	H_2	-0.0128	-0.0153	1.0005	1.0006	0.9962
$\alpha = 0.5$	H_1	1.0012	1.0012	-0.0021	-0.0009	-0.0042
	H_2	-0.0109	-0.0127	1.0003	1.0004	0.9973
$\alpha = 0.9$	H_1	1.0009	1.0009	-0.0019	-0.0007	-0.0036
	H_2	-0.0081	-0.0093	1.0002	1.0002	0.9987

A can be found by setting $\frac{\partial Q}{\partial A} = 0$, hence:

$$\frac{\partial Q}{\partial A} = -\sum_{k=1}^{N} \tilde{u}_k (F_k - A\tilde{u}_k)^t = 0$$

$$\Rightarrow -\sum_{k=1}^{N} \tilde{u}_k F_k^t + \sum_{k=1}^{N} \tilde{u}_k (A\tilde{u}_k)^t = 0$$

$$\Rightarrow A^t = \left(\sum_{k=1}^{N} \tilde{u}_k \tilde{u}_k^t\right)^{-1} \left(\sum_{k=1}^{N} \tilde{u}_k F_k^t\right)$$

hence:

$$A = \left(\sum_{k=1}^{N} F_k \tilde{u}_k^t\right) \left(\sum_{k=1}^{N} \tilde{u}_k \tilde{u}_k^t\right)^{-1} \tag{19}$$

As an illustration, let us use the the combination (2, 3) for clustering the synthetic data and (2, 2, 1) for clustering the medical data. Three values of the scaling factor α are applied: 0.3, 0.5, and 0.9. The results obtained are displayed in Tabs. XII and XIII. For the synthetic data, the first class has a strong relationship with the first two clusters for all values of α. This relationship is reflected by positive regression parameters and at the same time it is negatively related to the last two clusters. To understand which cluster strongly relates to which class for a given α, the regression tables are to be read column-wise. For a given cluster, the largest coefficient in the corresponding column indicates the class (row) to which the cluster belongs. This reasoning can also be applied on the medical data. Clearly, the coefficients of the first two clusters are the largest in the first row corresponding to class 1. The next two are strongly related to class 2 while the last one is strongly related to class 3. To conclude this section, it is worth pointing out that the regression parameters explain the relationship between clusters and classes according to the requested combination of clusters.

Table XIII. Regression parameters for the combination (2, 2, 1)

		C_1	C_2	C_3	C_4	C_5
$\alpha = 0.3$	H_1	1.0836	0.9203	-0.0004	-0.0008	-0.0015
	H_2	-0.6378	0.6244	0.9960	1.0131	-0.0072
	H_3	0.2406	-0.2625	-0.0020	-0.0101	1.0514
$\alpha = 0.5$	H_1	1.1735	0.8298	-0.0003	-0.0004	-0.0010
	H_2	-1.5019	1.4921	0.9954	1.0121	-0.0054
	H_3	0.8811	-0.8999	-0.0044	-0.0041	1.0445
$\alpha = 0.9$	H_1	1.1120	0.8906	-0.0010	0.0016	-0.0008
	H_2	1.0260	-1.0387	0.9755	1.0793	-0.0472
	H_3	0.7932	-0.8076	-0.0121	0.0184	1.0342

4.8. A Note on the Complexity of the Algorithm

Algorithm 1 consists of a two-level loop: *(i)* an outer loop of length $L = MaxIt$ and *(ii)* two nested loops: the first of length L_1 for optimizing Q (i.e., for computing $\tilde{U}$) and the second of length L_2 for optimizing J (i.e., for computing U and V). The computation of $\tilde{U}$ is of time complexity $Max(O(L_1CN), O(L_1HN))$ where C is the number of clusters, N is the number of samples, and H is the number of classes (Eq. 13). Knowing that $H < C$, the time complexity will be $O(L_1CN)$. On the other hand, the computation of each of U and V is of complexity $O(L_2CN)$. Since these are computed sequentially, the complexity is $Max(O(L_2CN), O(L_2CN))$ which yields $O(L_2CN)$. Note that the formulas are split into sub-formulas and then implemented sequentially. Taking the outer loop into account, the whole time complexity is: $Max(O(LL_1CN), O(LL_2CN))$. Let L' be $Max(L_1, L_2)$, then the overall complexity will be $O(LL'CN)$. This means that the size of the data, the number of iterations, and the number of clusters have an impact on the complexity of the algorithm. For each of these parameters, the algorithm has linear time complexity.

For the sake of illustration, let us consider a synthetic data set consisting of 10 features, 1000 samples distributed over 10 clusters, each cluster represents a class. Fig. 17(a) shows that the number of iterations in the outer loop ($L = MaxIter$) affects the computational time (the other parameters are frozen). The amount of time varies at different rates over increases of the number of iterations. It is clear from the plot that the algorithm has linear time complexity in the number of iterations. Similar results are obtained when varying the number of iterations in the nested loops of the algorithm (Fig. 17(b)), varying the number of clusters (Fig. 17(c), where a class consists of several clusters), and increasing the number of samples (Fig. 17(d)).

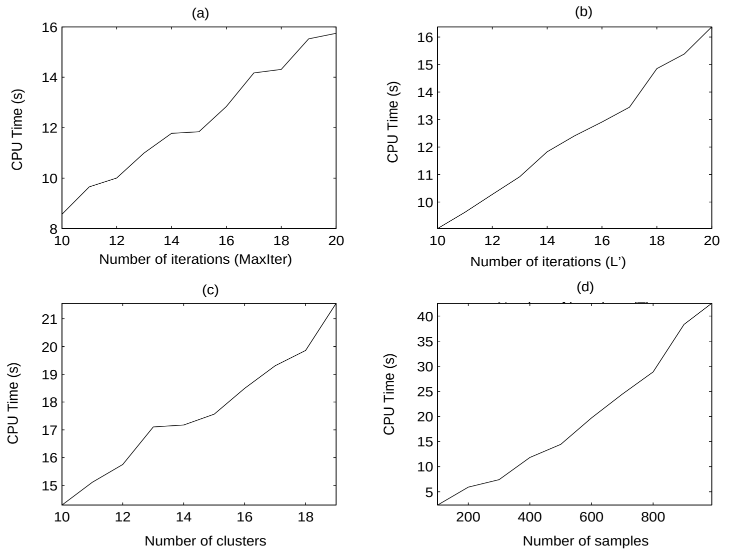


Figure 17. The execution time of the algorithm after varying the key parameters

5. Conclusion and Future Work

The paper discusses a new approach to performing fuzzy clustering with partial supervision. This approach exploits available knowledge about data to supervise the clustering process. The experimental evaluation has shown that the approach performs very well on different aspects like the discrimination of classes, the ability to control the structure of clusters, and the classification performance.

In some cases, e.g., when the data supplier is unable to classify samples into clear partitions or when labels come from another clustering source, the matrix F will have values from the interval [0,1]. Furthermore, so far the Euclidean distance is applied allowing us to generate clusters with only a spherical shape. A further development is to devise a mechanism so that hyperspherical clusters can be generated. To achieve that, the algorithm should be equipped with a more adaptive distance. Therefore, it would be interesting to investigate the algorithm with respect to these aspects.

References

Amini, M. and P. Gallinari. Semi-supervised learning with explicit misclassification modeling. *Proceedings of the* 18^{th} *International Joint Conference on Artificial Intelligence*, pages 555-561, 2003.

Basu, S., A. Banerjee, and R. Mooney. Semi-supervised clustering by seeding. *Proceedings of the Int. Conference on Machine Learning*, pages 19–26, 2002.

Bennett, K. and A. Demiriz. Semi-supervised support vector machines. *Advances in Neural Information Processing Systems*, vol.11, pp:368-374, 1999.

Bezdek, J. C. Pattern recognition with fuzzy objective function algorithms. *Plenum, New York*, 1981.

Bishop, C. Neural networks for pattern recognition. *Oxford press, New York*, 1995.

Blum, A. and T. Mitchell. Combining labeled and unlabaled data with co-training. *Proceedings of the* 11^{th} *Annual Conference on Computatioonal Learning Theory*, pages 92-100, 1998.

Blum, A., J. Lafferty, M. Rwebangira, and R. Reddy. Cluster kernels for semi-supervised learning. *Proceedings of the* 21^{th} *International Conference on Machine Learning*, pages 92–100, 2004.

Bouchachia, A. RBF networks for learning from partially labeled data. *Proceedings of the workshop on learning with partially classified training data at the* 22^{nd} *international conference on machine learning*, pages 10–18, Bonn, 2005.

Bouchachia, A. Learning with hybrid data. *Proceedings of the* 5^{th} *International IEEE Conference on Intelligent Hybrid Systems*, Rio de Janeiro, 2005.

Chapelle, O., J. Weston, and B. Schölkopf. Semi-supervised learning using randomized mincuts. *Advances in Neural Information Processing Systems*, vol.15, pp:585–592, 2002.

Demiriz, A., K. Bennett, and M. Embrechts. Semi-supervised clustering using genetic algorithms. *Intelligent Engineering Systems*, pages 809–814, 1999.

Guyon, I., N. Matic, and V. Vapnik. Discovering information patterns and data cleaning. *Advances in Knowledge Discovery and Data Mining. U. Fayyad et al. eds. AAAI Press*, pp:181–203, 1996.

Hathaway, R.J., J. Bezdek, and Y. Hu. Generalized fuzzy C-Means clustering strategies using L_p-norm distances. *IEEE Transaction on Fuzzy Systems*, Vol.8, No.5, pp:576–582, 2000.

Jeon, B. and D. Landgrebe. Partially supervised classification using weighted unsupervised clustering. *IEEE Transactions on Geoscience and Remote Sensing*, 37(2):1073–1079, 1999.

Klinkenberg, R. Using labeled and unlabeled data to learn drifting concepts. *Proceedings of the Workshop on Learning from Temporal and Spatial Data*, pages 16–24, 2001.

Mason, R., D. Lind, and W. Marchal Statistics: An Introduction. *Harcourt Brace Jovanovich, Inc.*, 1983.

Nigam, K., A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using Expectation-Maximization. *Machine Learning*, 39(2/3):103–134, 2000.

Pedrycz, W. and J. Waletzky. Fuzzy clustering with partial supervision. *IEEE Transactions on Systems Man and Cybernetics*, B 27(5):787–795, 1997.

Pizzi, N. Fuzzy pre-processing of gold standards as applied to biomedical spectra classification. *Artificial Intelligence in Medicine*, Vol.16, pp:171–182, 1999.

Snedecor, G. and W. Cochran Statistical Methods. *8th edition, Iowa State University Press*, 1989.

Suykens, J. and J. Vandewalle, Least squares support vector machine classifiers. *Neural Processing Letters*, Vol.9(3), pp:293–300, 1999.

Zhu, X., J. Kandola, Z. Ghahramani, and J. Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. *Advances in Neural Information Processing Systems*, vol.17, pp:1641–1648, 2005.