Possibilistic fuzzy c-means with partial supervision

Violaine Antoine[a,*], Jose A. Guerrero[b], Gerardo Romero[c]

[a] ISIMA, Clermont Auvergne University, UMR 6158 CNRS, Limos, F-63000, Clermont-Ferrand, France

[b] Institut Pascal, Clermont Auvergne University, UMR 6602 CNRS, F-63000, Clermont-Ferrand, France

[c] Electronic Department at University Autonomous of Tamaulipas, UAM Reynosa-Rodhe, Reynosa, México

Abstract

Clustering is a data analysis method that creates groups of objects according to a similarity notion. Amongst the existing clustering algorithms, the possibilistic fuzzy c-means (PFCM) is a well-known algorithm since it generates a possibilistic partition. Such possibilistic partition is helpful in the presence of a noisy environment and allows to express various types of uncertainty and imprecision. In recent years, the performance of clustering methods has been improved by incorporating partial information. The approach, called semi-supervised clustering, introduces instance-level information such as labeled patterns in the clustering process. In this work, we propose to extend PFCM to combine labeled patterns with the possibilistic framework. To provide more flexibility to the new method, in addition to the Euclidean distance, an adaptive distance measure is considered. Experimental results show the interest of our new semi-supervised possibilistic fuzzy c-means algorithm on various data sets.

Keywords: Partially supervised clustering, possibility, labeled patterns, Mahalanobis distance

1. Introduction

Clustering is a research field of data mining considered as an essential tool to analyze data. It aims at grouping objects into clusters following a similarity notion. There exist two families of clustering methods: the hierarchical and the partition-based methods [1]. Partitional clustering methods can be divided into distinct categories, ranging from crisp (hard) partitional clustering to various types of soft partition-based methods such as fuzzy, possibilistic, etc.

A crisp partitional clustering algorithm organizes data into clusters by assigning each object unambiguously to a single cluster. A large number of clustering methods generating crisp partition have been proposed over the last few

[*] Corresponding author
Email address: violaine.antoine@uca.fr (Violaine Antoine)

decades. A complete review of these methods can be found in [2]. Among them, the *k-means* algorithm has attracted a lot of attention since it is a simple and fast optimization-based method. However, as any crisp partitional algorithm, it does not allow to represent uncertainty in the case for instance of overlapping clusters. Thus, variants of the *k-means* algorithm generating soft partitions have been developed ever since.

The most known extensions of *k-means* are the *fuzzy c-means* (FCM) algorithm [3] and its version with an adaptive distance called FCM-GK [4]. The goal of an adaptive distance is to adjust a specific distance to a clustering problem. In FCM-GK, it corresponds to Mahalanobis distances where covariance matrices are adjusted for each cluster. The FCM and FCM-GK methods, which have been used in many applications [5, 6, 7, 8], are based on the fuzzy sets theory [9]. They produce a fuzzy partition that allows to express a degree of uncertainty for the membership of each object to each cluster. More precisely, an object has a probability to belong to a cluster and the sum of the probabilities to each existing cluster is equal to 1. Thus, a fuzzy partition is also referred to as a probabilistic partition. Although such partition provides richer information than a crisp one, it does not allow to describe imprecision. Hence, an object may be assigned to a cluster even if it is far from this cluster. This can lead to anomalous clustering results. To overcome such weakness, the *possibilistic c-mean* (PCM) algorithm introduced in [10] relaxes the probabilistic constraints of FCM to obtain a possibilistic partition. Similarly to *k-means* and FCM, the PCM algorithm is based on an objective function minimized using a heuristic method. However in PCM, the heuristic highly depends on initial parameter settings and often generates coincident clusters [11]. To avoid such shortcoming, several algorithms have been proposed [12, 13, 14, 15]. In [12], the possibilistic partition obtained with the objective function of PCM is altered in order to create a core for each cluster. Although the algorithm presents a guideline for parameter settings and is robust to coincident clusters, it does not follow a strict optimization. In [13], a repulsive term has been added in the PCM objective function in order to force the separation of the clusters. Nonetheless, this method, called RPCM, is sensitive to the parameter settings. In particular, adjusting the weight given to the repulsive term is a complex task. Another way to overcome the problem of coincident clusters is to linearly combine the objective function of FCM and PCM [14]. The algorithm, referred to as PFCM, is an improved version of FPCM [16]. It corresponds to the most famous extension of PCM since it obtains good results with reasonable settings.

It is well known that clustering is a complex task and an ill-defined problem. The little information given to a clustering method is often not enough to obtain a relevant solution. Nonetheless, in many domains where clustering methods are applied, there exists some background information that can help to improve the extracted knowledge from data analysis [17, 18, 19]. This has led to the development of a new category of clustering algorithms that can be organized in *semi-supervised clustering* and *supervised clustering*.

Supervised clustering has been defined as the problem of learning a dissimilarity function from labeled data that will be used by a clustering algorithm to partition unlabeled data [20, 21]. This problem has also been described as a prediction problem in [22] similar to that of the classification problem. Thus, supervised clustering aims at extracting the data structure using a two-stage process: model parameter learning and cluster prediction for new unlabeled data.

The essence of semi-supervised clustering methods is the same as for unsupervised clustering methods: to extract the data structure directly from the data set with no learning involved. These methods incorporate some a priori information in the form of a few constraints or labels. Various types of a priori information have been considered in the literature: pairwise constraints [23, 24], labeled patterns [25, 26, 27], and others [28, 29]. Although semi-supervised and supervised clustering seem closely related, the idea of transferring learned knowledge to cluster new unlabeled data in supervised clustering leads to significantly different problem formulations and solutions. A complete discussion on the differences between semi-supervised and supervised learning can be found in [30, 31]. The focus of this work is on semi-supervised methods.

Semi-supervised clustering algorithms with labeled patterns, also referred to as *partially supervised clustering*, have been mainly proposed when labeled patterns are scarce, which is often the case in real applications. As shown in Figure 1, the idea of clustering with partial supervision emerged in the mid-80s. Clustering with partial supervision assumes that some labeled patterns are available and can be used to improve clustering algorithms [32].

A well-known variant of k-means that provides a hard partition is SKMEANS [26]. It uses hard labeled patterns and consequently, it is unable to model uncertainty on the a priori knowledge. Without a doubt, the literature is rich on fuzzy (soft) partition-based clustering methods: SFCM85 [32], SFCM97 [25], SFCM04 [33], 2kSFCM04 [34], SFCM06a [27], SFCM06b [35], ESFCM09 [36], SSFCM09 [36], and SFCM-HPR13 [37]. Fuzzy partition-based methods define degrees of probability on labels. Such soft labels have two advantages: they are more abundant in real applications and an expert can decide to decrease the probabilities when there exists noise in the labels, in order to maintain good performances [38]. Since SFCM is an extension of FCM and both generate fuzzy partitions, it is clear that the probabilistic framework used in SFCM fails to represent imprecision.

Early partially supervised clustering algorithms capable of dealing with imprecision are based on the possibilistic framework discussed in [10] and [13]. [39] extends RPCM [13] to take into account a priori knowledge or labeled patterns in an algorithm called semi-supervised PCM (SPCM). SPCM modified the objective function in [13] to maximize the distance between clusters while it seeks to minimize the inverse of the distance. [39] uses a Euclidean distance and solves the optimization problem using first-order necessary conditions. Thus, since SPCM is an extension of RPCM, it faces the coincident cluster centroids problem. Coincident cluster centroids are handled, during the optimization process, by removing and re-initializing them at another location. This problem is discussed in [40] and [41] where the convexity of the objective function of the SRPCM algorithm is analyzed. A second-order condition optimization method

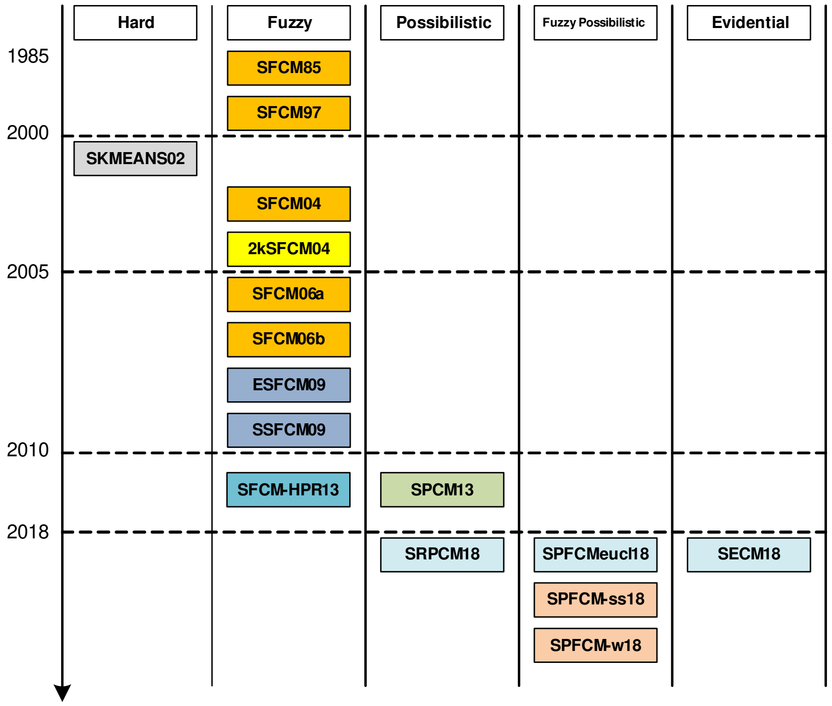


Figure 1: Historical time line of semi-supervised clustering algorithms from 1985 to 2018. Hybrid column refers to the algorithms that provide a possibilistic and probabilistic partition.

is used in [41] that guarantees the convergence of the solution and thus solves the problem of coincident cluster centroids. [41] also introduces the idea of semi-supervised possibilistic fuzzy c-means clustering SPFCM-eucl using Euclidean distance. [42] and [43] present an extension to the PFCM algorithm by using labeled centroids instead of labeled instances or samples. The two methods are dedicated to image segmentation and take into account the neighborhood of the pixels in the objective functions. In [44, 45] SECM, an extension of the evidential c-means algorithm (ECM) [46] to take into account labeled patterns, is presented. SECM allows the extraction of much richer partition information at the price of a much higher computational complexity.

Most existing semi-supervised clustering algorithms have been developed using a Euclidean distance which, in general, may not match the structure of the data set to which they are applied to. Yet, using an adaptive distance in the form of a Mahalanobis distance gives more flexibility to adjust a metric for each cluster, and constraints or labels can help to guide the algorithm towards more accurate metrics. Such adaptive distance has been introduced in semi-supervised clustering algorithms with pairwise constraints [47, 48, 24] and labeled patterns [27, 38]. These methods have shown improvement in the clustering performances on data sets with clusters having non-spherical shapes.

Although there exist several semi-supervised clustering algorithms with an adaptive distance, none ally the advantages of the possibilistic framework and the use of an adaptive distance. In this work, a new algorithm called SPFCM-mah is presented. The new method is an extension PFCM [14] and SPFCM-eucl [41]. It introduces the use of the Mahalanobis distance and incorporates a few labeled patterns with a degree of possibility, it adjusts its proper metric, and generates a possibilistic partition. Such possibilistic framework, which is richer than hard and probabilistic framework, and such adaptive distance make it possible to express labeled patterns in a more flexible way and to obtain more precise solutions. We also present a complete analysis of the performance of SPFCM-mah.

The paper is organized as follows: first, section 2 reviews possibilistic clustering algorithms related to this work. Next, section 3 presents the new semi-supervised clustering algorithm with labeled patterns and an adaptive distance in the form of Mahalanobis distances. This section describes how labeled patterns are expressed and integrated into PFCM. Experimental settings and results are discussed in section 4. It includes an analysis of various parameters and partially supervised clustering comparisons. Finally, conclusions and perspectives of our work are presented in the last section.

2. Preliminaries on soft partitional clustering

Given a collection of n objects represented by a set of feature vectors $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathbb{R}^p$, *k-means* and its extensions are algorithms that search for c clusters such that each cluster k is represented by a prototype (or centroid) $\mathbf{v}_k \in \mathbb{R}^p$. The matrix $\mathbf{V} = [\mathbf{v}_k]$ corresponds to the set of the c centroids.

2.1. Fuzzy c-means

The *fuzzy c-means* algorithm is a method that generates a fuzzy partition, which conveniently can be expressed as a matrix, $\mathbf{U} = [u_{ik}]$ such that a degree of uncertainty u_{ik} is defined for the membership of each object i to each cluster k. The FCM algorithm aims at minimizing an objective function considered as the weighted-within group sum of squared errors:

$$J_{FCM}(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{n} \sum_{k=1}^{c} u_{ik}^m d_{ik}^2, \tag{1}$$

subject to

$$\sum_{k=1}^{c} u_{ik} = 1, \qquad \forall i = \{1, \ldots, n\}, \tag{2}$$

$$\sum_{i=1}^{n} u_{ik} > 0, \qquad \forall k = \{1, \ldots, c\}, \tag{3}$$

$$1 \geq u_{ik} \geq 0, \quad \forall i = \{1, \ldots, n\},\ k = \{1, \ldots, c\}. \tag{4}$$

The parameter $m > 0$, called *fuzzifier* and usually set between 1.5 and 3 [3], is an exponent controlling the fuzziness of the partition. The distance d_{ik} represents the Euclidean distance of the object $\mathbf{x}_i$ to the cluster k.

The minimization of FCM is carried out through an iterative optimization until the convergence of the clustering solution. First, the fuzzy partition $\mathbf{U}$ is fixed and J_{FCM} is minimized with the respect to the centroids $\mathbf{V}$. Then $\mathbf{V}$ is fixed and the partition $\mathbf{U}$ is updated. Estimations at a step of the algorithm for the centroid $\mathbf{v}_k$ such that $k \in \{1, \ldots, c\}$ and the degree of membership u_{ik} of an object $i \in \{1, \ldots, n\}$ in cluster k are given by:

$$u_{ik} = \left(\sum_{l=1}^{c} \left(\frac{d_{ik}}{d_{il}} \right)^{\frac{2}{m-1}} \right)^{-1}, \tag{5}$$

$$\mathbf{v}_k = \frac{\sum_{i=1}^{n} u_{ik}^{m} \mathbf{x}_i}{\sum_{i=1}^{n} u_{ik}^{m}}. \tag{6}$$

2.2. Possibilistic c-means

The *possibilistic c-means* is, historically, the first method that proposes to use the possibilistic framework to generate groups and deal with outliers [10]. It produces a possibilistic partition $\mathbf{T} = [t_{ik}]$ via the minimization of the following objective function:

$$J_{PCM}(\mathbf{T}, \mathbf{V}) = \sum_{i=1}^{n} \sum_{k=1}^{c} t_{ik}^{\eta} d_{ik}^{2} + \sum_{k=1}^{c} \gamma_k \sum_{i=1}^{n} (1 - t_{ik})^{\eta}, \tag{7}$$

subject to

$$\sum_{i=1}^{n} t_{ik} > 0, \qquad \forall k = \{1, \ldots, c\}, \tag{8}$$

$$1 \geq t_{ik} \geq 0, \qquad \forall i = \{1, \ldots, n\},\ k = \{1, \ldots, c\}. \tag{9}$$

The first term of J_{PCM} corresponds to the objective function of FCM. However, since PCM is relaxing the constraints (2), a second term has been integrated in order to avoid the trivial solution consisting of a possibilistic partition with only null values. The parameter η corresponds as m to a coefficient that controls the fuzziness of the possibilistic partition. In [10], it is set to 2. The parameters γ_k are positive coefficients that determine the area of influence of the clusters. These fixed parameters can have a great impact on the clustering results. The γ_k parameters are usually computed as follows [10]:

$$\gamma_k = K \frac{\sum_{i=1}^{n} u_{ik}^m d_{ik}^2}{\sum_{i=1}^{n} u_{ik}^m}, \tag{10}$$

where $\mathbf{U} = [u_{ik}]$ is the probabilistic partition obtained by applying the FCM algorithm and K, usually set to 1, is a weighting factor enabling to reduce or increase the overall size of the clusters.

Finally, PCM follows the same iterative procedure as FCM in order to minimize its objective function. Performing the alternate optimization of $\mathbf{T}$ and $\mathbf{V}$ until convergence of the solution leads to the subsequent updating equations:

$$t_{ik} = \left(1 + \left(\frac{d_{ik}^2}{\gamma_k}\right)^{\frac{1}{\eta-1}}\right)^{-1}, \tag{11}$$

$$\mathbf{v}_k = \frac{\sum_{k=1}^{n} t_{ik}^{\eta} \mathbf{x}_i}{\sum_{k=1}^{n} t_{ik}^{\eta}}. \tag{12}$$

Inversely to FCM, coefficients t_{ik} are cluster-independent. As a result, centroids become independent of each other and are free to move in the same dense region. Such behavior, worsen by the centroids initialization, leads to a solution with coincident clusters [11], although satisfying partitions should encompass c distinct clusters. Several algorithms have been proposed to solve this well-known issue of PCM. The algorithm described next is one of the most popular.

2.3. Possibilistic Fuzzy c-Means

The *possibilistic fuzzy c-means* algorithm solves the problem of coincident clusters by combining possibility and probability membership values in its objective function [14]:

$$J_{PFCM}(\mathbf{U}, \mathbf{T}, \mathbf{V}) = \sum_{i=1}^{n} \sum_{k=1}^{c} (a u_{ik}^m + b t_{ik}^{\eta}) d_{ik}^2 + \sum_{k=1}^{c} \gamma_k \sum_{i=1}^{n} (1 - t_{ik})^{\eta}, \tag{13}$$

such that constraints (2), (4), (9) are respected.

The η and γ_k parameter setting procedure remains as in PCM. The new parameters a and b are positive user-defined constants that control the importance given to the probability memberships and the possibility degrees, respectively. In [14], authors suggest setting a to a smaller value than b, while keeping a large enough to avoid coincident clusters.

To minimize the objective function, the PFCM algorithm alternates the optimization of **U**, **T** and **V** until convergence. The update formula of **U** is identical to FCM and is given by equation (5). Possibilistic membership and centroid update expressions, equations (14) and (15), remain similar to PCM:

$$t_{ik} = \left(1 + \left(\frac{b}{\gamma_k} d_{ik}^2 \right)^{\frac{1}{\eta-1}} \right)^{-1}, \tag{14}$$

$$\mathbf{v}_k = \frac{\sum_{i=1}^{n} (a u_{ik}^m + b t_{ik}^\eta) \mathbf{x}_i}{\sum_{i=1}^{n} (a u_{ik}^m + b t_{ik}^\eta)}. \tag{15}$$

3. PFCM with partial supervision and an adaptive distance

3.1. Introduction of labeled patterns in PFCM

Let us assume that some background knowledge is available; then, we can introduce the following definition:

Definition 1. **(*Soft label*)** Given a priori information about object $\mathbf{x}_i$, then the prior membership possibility $f_{ik} \in [0, 1]$ for object $\mathbf{x}_i$ to belong to cluster k is said to be a soft label.

The above definition allows us to express full possibility that $\mathbf{x}_i$ is associated to cluster k when its membership possibility is $f_{ik} = 1$. The opposite case can be expressed by $f_{ik} = 0$: it corresponds to the total certainty that $\mathbf{x}_i$ does not belong to cluster k. Notice that the possibilistic framework allows to express a strong possibility that object $\mathbf{x}_i$ belongs to multiple clusters k, k', etc. as $f_{ik} = 1$, $f_{ik'} = 1$ and so on.

To identify which objects are known to be associated (soft labeled patterns) with which clusters, we define the variable b_{ik} such that:

$$b_{ik} = \begin{cases} 1, & \text{if a possibilistic opinion that } \mathbf{x}_i \text{ belongs to cluster } k \text{ has been given,} \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

As it will be shown in equation (17), b_{ik} is used to define a penalty term in the clustering model that includes all possibilistic information f_{ik}. It is important to mention that only a small amount of objects might be labeled by human experts. Thus, it is necessary to state which possibilistic information is available by using the b_{ik} variable.

Example 1.

Let us consider a data set composed of n patients that have one (and only one) of the three following diseases: di_1, di_2, di_3. For all the patients, phenotype observations have been collected in the form of numerical attributes, leading to the creation of **X**. In this context, which disease which patient suffers from is not known. However, as a piece of extra information, a doctor has given a precise (or imprecise) opinion about the disease suffered by four patients. Indeed, the doctor thinks it is highly possible that patient $\mathbf{x}_1$ has di_1 or di_2, but not di_3, which leads to define $f_{11} = 1$, $f_{12} = 1$ and $f_{13} = 0$. For patient $\mathbf{x}_2$, the doctor has provided specific possibilities that he suffers from di_1 or di_2, i.e. $f_{21} = 0.8$, $f_{22} = 0.6$, but did not give an opinion about the last disease. Consequently, the possibility f_{23} is not defined (it is unknown). Finally, the doctor suspects patients $\mathbf{x}_3$ and $\mathbf{x}_4$ to have di_1. Since he is less sure for $\mathbf{x}_4$, he set a lower possibility than for $\mathbf{x}_3$: $f_{31} = 1$ and $f_{41} = 0.9$. No information has been supplied for di_2 and di_3.

Table 1 resumes the a priori information available for these patients. The variable b_{ik} is set to 1 when the doctor has mentioned a possibility. Note that possibilistic values f_{ik} related to $b_{ik} = 0$ are not provided because they do not influence the penalty term defined in (17). The other patients, from $\mathbf{x}_5$ to $\mathbf{x}_n$, with all b_{ik} $\forall k \in [1, 3]$ set to 0, correspond to patients where no opinion has been provided by any doctor. Since no background knowledge is available, they represent unlabeled patients.

Table 1: Background knowledge formulation.

	di_1		di_2		di_3	
	b_{i1}	f_{i1}	b_{i2}	f_{i2}	b_{i3}	f_{i3}
$\mathbf{x}_1$	1	1	1	1	1	0
$\mathbf{x}_2$	1	0,8	1	0,6	0	-
$\mathbf{x}_3$	1	1	0	-	0	-
$\mathbf{x}_4$	1	0,9	0	-	0	-
$\mathbf{x}_5$	0	-	0	-	0	-
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathbf{x}_n$	0	-	0	-	0	-

Inspired by [25], we propose to add the penalty term presented below in J_{PFCM} in order to take into account the soft labeled patterns:

$$J_{penalty} = \sum_{i=1}^{n} \sum_{k=1}^{c} b_{ik} (t_{ik} - f_{ik})^{\eta} d_{ik}^2, \tag{17}$$

where $\eta > 0$, the parameter that controls the fuzziness of the possibilistic partition, is even. Ideally, the obtained possibilistic value t_{ik} should be identical to its prior degree of possibility f_{ik}. However, such requirement leads to inconsistent solutions when a labeled object i is near a centroid $\mathbf{v}_k$ for which the possibility f_{ik} is low. This situation occurs with noisy labels or with data sets that contain intricate structures. Consequently, as in [25], the penalty value is relaxed for objects closed to centroids by considering d_{ik}^2 as a coefficient.

3.2. Adaptive distance

PFCM generates a partition whose clusters take a spherical shape that often does not represent the real structure of the data. In this work, we propose to use an adaptive distance such that (1) the shape of the resulting clusters best matches the structure of the data and (2) the solution of the partially supervised clustering algorithm is lead towards a desired solution. Let us provide a definition of adaptive distance in the general clustering framework.

Definition 2. (Adaptive distance) Let $\mathbf{X}$ be a data set that can be partitioned into c groups or clusters such that $\bigcup_k \mathbf{X}_k \in \mathbf{X}$ and $\mathbf{X}_k \cap \mathbf{X}_{k'} = \emptyset$. Then, a metric d, under which the closer the distance between $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_k$ and the farther the distance between objects $\mathbf{x}_i \in \mathbf{X}_k, \mathbf{x}_j \in \mathbf{X}_{k'}$ with $k \neq k'$, whose parameters are adjusted from the data is said to be an adaptive distance.

An adaptive measure that follows definition 2 and that has been widely used in the literature is the Mahalanobis distance. Thus, following [4], we define a Mahalanobis distance specific for each cluster. Let $\mathbf{S} = \{\mathbf{S}_1, \ldots, \mathbf{S}_c\}$ denote a set of c matrices, where $\mathbf{S}_k$ corresponds to a $(p \times p)$ positive definite matrix associated to cluster k. The squared Mahalanobis distance between an object $\mathbf{x}_i$ and the cluster centroid k is defined as follows:

$$d_{ik}^2 = \|\mathbf{x}_i - \mathbf{v}_k\|_{\mathbf{S}_k} = (\mathbf{x}_i - \mathbf{v}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{v}_k). \qquad (18)$$

3.3. Objective function

By incorporating the penalty term (17) in J_{PFCM} and by employing adaptive distances (18), the new criterion to be minimized becomes:

$$\begin{aligned} J_{SPFCM}(\mathbf{U},\mathbf{T},\mathbf{V},\mathbf{S}) &= \sum_{i=1}^{n}\sum_{k=1}^{c}(au_{ik}^m + bt_{ik}^{\eta})\|\mathbf{x}_i - \mathbf{v}_k\|_{\mathbf{S}_k} \\ &+ \sum_{k=1}^{c}\gamma_k \sum_{i=1}^{n}(1-t_{ik})^{\eta} \\ &+ \alpha\sum_{i=1}^{n}\sum_{k=1}^{c} b_{ik}(t_{ik} - f_{ik})^{\eta}\|\mathbf{x}_i - \mathbf{v}_k\|_{\mathbf{S}_k}, \qquad (19) \end{aligned}$$

subject to (2), (4), (9) and

$$\det(\mathbf{S}_k) = \rho_k, \qquad \forall k \in \{1, \ldots, c\}. \qquad (20)$$

where $\alpha \geq 0$ is the new parameter introduced by our algorithm. This parameter α is a trade-off coefficient between the inherent structure retrieved without any supervision and the consideration of the labeled patterns. Parameters a, b, m and γ_k are defined as PFCM. In order to facilitate the optimization of the objective function, η is set to 2. Constraints on the volume of the clusters are imposed using equation (20), where $\rho_k > 0$. These constraints allow to avoid degenerate solution consisting to fix all matrices $\mathbf{S}_k$ with zero entries.

3.4. Optimization

The optimization of function (19) under constraints (2), (4), (9) and (20) is an NP-hard problem. It can be solved similarly to FCM by minimizing the objective function with the respect first to $\mathbf{U}$, then $\mathbf{T}$, then $\mathbf{V}$ and finally $\mathbf{S}$ until convergence of the solution.

3.4.1. Update of the probabilistic partition

The optimization of J_{SPFCM} with respect to $\mathbf{U}$ is achieved by fixing $\mathbf{T}$, $\mathbf{V}$ and $\mathbf{S}$ as constants. Since the penalty term (17) incorporated to SPFCM does not depend on the probabilistic partition, the update of the membership degrees $\mathbf{U}$ are identical to PFCM, thus, they are computed using (5).

3.4.2. Update of the possibilistic partition

In order to minimize J_{SPFCM} with respect to $\mathbf{T}$, the variables $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{S}$ are fixed. Since $\mathbf{T}$ is a matrix with independent rows and columns, each value $t_{ik} \in \mathbf{T}$ can be minimized separately. The derivative of the objective function with respect to t_{ik} is:

$$\frac{\partial J_{SPFCM}}{\partial t_{ik}} = \eta b t_{ik}^{\eta-1} d_{ik}^2 - \eta\gamma_k (1-t_{ik})^{\eta-1} + \eta\alpha b_{ik} d_{ik}^2 (t_{ik} - f_{ik})^{\eta-1}. \qquad (21)$$

Finding an expression for t_{ik} using $\eta > 2$ is not straightforward since the derivative of the objective function results in a polynomial of degree $\eta - 1$, as it can be observed in (21), which might be solved using numerical methods. This problem has also been pointed out in [25]. Similarly to [25], in order to provide an expression for t_{ik}, we set $\eta = 2$. The derivative is then:

$$\frac{\partial J_{SPFCM}}{\partial t_{ik}} = 2bt_{ik}d_{ik}^2 - 2\gamma_k(1-t_{ik}) + 2\alpha b_{ik} d_{ik}^2 (t_{ik} - f_{ik}). \qquad (22)$$

From (22), we obtain an expression to update the possibilistic memberships:

$$t_{ik} = \frac{\gamma_k + \alpha b_{ik} d_{ik}^2 f_{ik}}{b d_{ik}^2 + \gamma_k + \alpha b_{ik} d_{ik}^2}. \qquad (23)$$

3.4.3. Update of the centroids

Since $(au_{ik}^m + bt_{ik}^2) > 0$ and $(t_{ik} - f_{ik})^2 > 0$, J_{SPFCM} is positive semi-definite with respect to $\mathbf{V}$. As a consequence, the minimum of the objective function corresponds to the value of $\mathbf{V}$ that makes the derivative vanish. Notice that each centroid $\mathbf{v}_k$ is independent of each other and can then be handled separately.

$$\frac{\partial J_{SPFCM}}{\partial \mathbf{v}_k} = \sum_{i=1}^{n} (au_{ik}^m + bt_{ik}^2)(-2\mathbf{S}_k(\mathbf{x}_i - \mathbf{v}_k)) + \alpha \sum_{i=1}^{n} b_{ik}(t_{ik} - f_{ik})^2 (-2\mathbf{S}_k(\mathbf{x}_i - \mathbf{v}_k)). \qquad (24)$$

Let z_{ik} be a scalar such that $z_{ik} \triangleq (au_{ik}^m + bt_{ik}^2) + \alpha b_{ik}(t_{ik} - f_{ik})^2$. Annulling the derivative leads to the following result:

$$\sum_{i=1}^{n} \left((au_{ik}^m + bt_{ik}^2) + \alpha b_{ik}(t_{ik} - f_{ik})^2\right) \mathbf{S}_k(\mathbf{x}_i - \mathbf{v}_k) = 0, \tag{25}$$

$$\sum_{i=1}^{n} z_{ik}\mathbf{S}_k\mathbf{x}_i = \left(\sum_{i=1}^{n} z_{ik}\right) \mathbf{S}_k\mathbf{v}_k. \tag{26}$$

Hence, the update formula for the centroids is:

$$\mathbf{v}_k = \frac{\sum\limits_{i=1}^{n} z_{ik}\mathbf{x}_i}{\sum\limits_{i=1}^{n} z_{ik}}. \tag{27}$$

3.4.4. Update of the Mahalanobis distance

Matrices $\mathbf{U}$, $\mathbf{T}$, and $\mathbf{V}$ are now fixed and the set of matrices $\mathbf{S}$ should be determined. In order to solve the minimization problem with constraints (20) and with respect to $\mathbf{S}$, we introduce c Lagrange multipliers δ_i and write the Lagrangian:

$$\mathcal{L}(\mathbf{S}, \delta_1, \ldots, \delta_c) = J_{SPFCM} + \sum_{k=1}^{c} \delta_k(det(\mathbf{S}_k) - \rho_k). \tag{28}$$

By deriving $\mathcal{L}$ with respect to the Lagrange multipliers and annulling the derivative, we obtain

$$\rho_k = det(\mathbf{S}_k). \tag{29}$$

It can be seen that the above equation is identical to (20).

The Lagrangian is now minimized with the respect to $\mathbf{S}$. Since matrices are independent of each other, each matrix $\mathbf{S}_k$ can be optimized separately. The derivative of $\mathcal{L}$ with respect to $\mathbf{S}_k$ is:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{S}_k} &= \sum_{i=1}^{n}(au_{ik}^m + bt_{ik}^2)(\mathbf{x}_i - \mathbf{v}_k)(\mathbf{x}_i - \mathbf{v}_k)^T \\ &\quad + \alpha \sum_{i=1}^{n} b_{ik}(t_{ik} - f_{ik})^2(\mathbf{x}_i - \mathbf{v}_k)(\mathbf{x}_i - \mathbf{v}_k)^T \\ &\quad - \delta_k \det(\mathbf{S}_k)\mathbf{S}_k^{-1}. \end{aligned} \tag{30}$$

Let $\mathbf{\Sigma}_k$ be a matrix such that

$$\mathbf{\Sigma}_k \triangleq \sum_{i=1}^{n} \left((au_{ik}^m + bt_{ik}^2) + \alpha b_{ik}(t_{ik} - f_{ik})^2\right) (\mathbf{x}_i - \mathbf{v}_k)(\mathbf{x}_i - \mathbf{v}_k)^T. \tag{31}$$

This matrix can be considered as a partially supervised possibilistic covariance matrix. Notice that each term $(\mathbf{x}_i - \mathbf{v}_k)(\mathbf{x}_i - \mathbf{v}_k)^T$ is symmetric and positive semi-definite matrix, and each scalar $(au_{ik}^m + bt_{ik}^2) + \alpha b_{ik}(t_{ik} - f_{ik})^2$ is ≥ 0. Hence, the weighted sum corresponding to $\mathbf{\Sigma}_k$ is a symmetric positive semi-definite matrix. Annulling the derivative of the Lagrangian gives:

$$\mathbf{\Sigma}_k - \delta_k \det(\mathbf{S}_k)\mathbf{S}_k^{-1} = 0, \tag{32}$$

$$\mathbf{\Sigma}_k = \delta_k \det(\mathbf{S}_k)\mathbf{S}_k^{-1}. \tag{33}$$

Using constraint (20) and multiplying by $\mathbf{S}_k$ leads to:

$$\mathbf{\Sigma}_k \mathbf{S}_k = \delta_k \rho_k \mathbf{I}, \tag{34}$$

where $\mathbf{I} \in \mathbb{R}^{(p \times p)}$ is the identity matrix. Computing the determinant of the above equation and using the properties of a squared matrix, we obtain:

$$\det(\mathbf{\Sigma}_k \mathbf{S}_k) = \delta_k^p \rho_k^p, \tag{35}$$

$$\det(\mathbf{\Sigma}_k)\rho_k = \delta_k^p \rho_k^p, \tag{36}$$

$$\delta_k = \frac{(\rho_k \det(\mathbf{\Sigma}_k))^{\frac{1}{p}}}{\rho_k}. \tag{37}$$

Thus, introducing (37) in (33), the equation to update the matrix $\mathbf{S}_k$ is:

$$\mathbf{S}_k = \rho_k^{\frac{1}{p}} \det(\mathbf{\Sigma}_k)^{\frac{1}{p}} \mathbf{\Sigma}_k^{-1}, \qquad \forall k \in \{1, \ldots, c\}. \tag{38}$$

Since $\mathbf{\Sigma}_k$ is positive semi-definite, it can occur that the matrix is not invertible in rare situations such as linearly correlated points in a cluster. A solution identical to [49] can be employed to overcome the issue, see Appendix A.

Finally, algorithm 1 resumes the overall procedure of the SPFCM algorithm. As any variants of k-means, this algorithm is a heuristic method that allows to find a local minimum for the objective function of SPFCM. The solution depends on the initialization of the centroids matrix $\mathbf{V}_0$. Thus, it is important to run several times the algorithm with different initializations of $\mathbf{V}_0$. The simplest way is to randomly initialize the centroids in the data space range for each run of the algorithm. Alternatively, one of the run can initialize $\mathbf{V}_0$ using the mean of the known labeled patterns [50].

3.5. Complexity analysis

Complexity analysis of the k-Means variants has been largely studied [51]. The FCM and the PFCM algorithms with a Euclidean distance have both the same complexity of $O(tcnp^2)$, where t represents the number of iterations reached before convergence. The complexity of our algorithm is defined by equations (5), (23), (27), (38). The probabilistic partition update equation (5) requires $O(cp)$ operations using the implementation [52]. The possibilistic partition update equation (23) runs in $O(cn)$ time and the centroids update equation (27) in $O(cnp)$ time. The computation of the $\mathbf{\Sigma}_k$ matrix equation (31)

Algorithm 1 SPFCM with an adaptive distance

```
Require: Data collection $\mathbf{X}$, number of desired clusters $c$, set of labeled objects
    (label patterns) $F = \{f_{ik}\}$, convergence threshold $\varepsilon$.
Ensure: Possibilistic partition $\mathbf{T}$, centroid matrix $\mathbf{V}$, set of matrices $\mathbf{S}$.
Randomly initialize $\mathbf{V}_0$.
$t \leftarrow 0$
repeat
    Update probabilistic partition $\mathbf{U}_t$ using equation (5).
    Update possibilistic partition $\mathbf{T}_t$ using equation (23).
    Update centroids $\mathbf{V}_t$ using equation (27).
    Calculate $\mathbf{\Sigma}_k \forall k \in \{1, \ldots, c\}$ using equation (31)
    for each $\mathbf{\Sigma}_k$ do
        if $\mathbf{\Sigma}_k$ is singular then
            Render $\mathbf{\Sigma}_k$ positive definite using Algorithm 2.
        end if
    end for
    Update the set of matrices $\mathbf{S}_t$ using equation (38).
    $t \leftarrow t + 1$
until $\|\mathbf{V}_t - \mathbf{V}_{t-1}\| < \varepsilon$
```

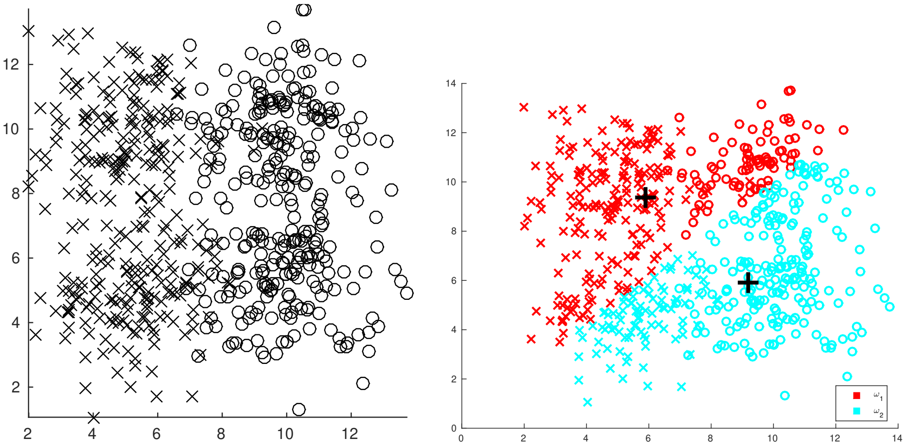
requires $O(cnp^2)$ operations and the update equation of the matrices (38) requires $O(cp^3)$ time. In total, the SPFCM algorithm with a Mahalanobis distance has a complexity of $O(tncp^2 + tcp^3)$, which corresponds to the complexity of FCM with a Mahalanobis distance [4]. The SPFCM with a Euclidean distance has a complexity identical to FCM and PFCM with a Euclidean distance, i.e. $O(tcnp^2)$.

4. Experimental Results

4.1. Interest of adding labeled patterns

To illustrate the interest of the SPFCM algorithm, we created a synthetic data set named gaussK2. As it can be observed in Figure 2(a), gaussK2 consists of four Gaussians generated in a two-dimensional space and divided into two vertical classes. The hard partition obtained after applying PFCM with a Euclidean distance (PFCM-eucl) on this data set, $a = 0.5$, $b = 5$ and after computing the maximal possibilistic rule is shown in Figure 2(b). The Euclidean distance, which only allows the search of spherical classes, is not adapted for the gaussK2 data set. This explains the singular partition obtained, with a diagonal boundary.

In order to take into account the inherent structure of the data, we run PFCM with a Mahalanobis distance (PFCM-mah) and the same parameter settings as for PFCM-eucl. PFCM-mah was run five times on gaussK2 with various centroids initialization randomly chosen. The possibilistic partition providing the lowest objective function value was kept. As illustrated in Figure 3(a),



(a)

(b)

Figure 2: The gaussK2 data set: (a) the true labels, (b) hard partition obtained using PFCM-eucl. Symbols '×' and 'o' represent the real classes. Colors and big crosses in (b) indicate the clusters found by PFCM-eucl and their centroids, respectively.

PFCM-mah obtains a horizontal boundary between the two clusters. However, it is clear that this solution does not correspond to the true partition of the data.

To provide labeled patterns on our synthetic data set, we play the role of the human expert by randomly selecting and labeling 10 objects and execute SPFCM with the same parameters and $\alpha = 1$. Notice that, as in real scenarios, the human expert must make sure that labeled patterns are representative enough of the data. We applied five times SPFCM-mah on gaussK2 with random centroid initialization and we retain the solution with the minimum objective function value. As it can be observed in Figure 3(b), a small number of labeled patterns is enough to lead the algorithm toward the desired representative solution of the data structure.

4.2. Data sets

The performance of the proposed algorithm has been evaluated on various well-known data sets coming from the UCI Machine Learning repository[1]. The experiments were performed on an Intel(R) Xeon(R) CPU E5-2670 v2 @ 2.50 GHz. Table 2 presents the characteristics of each data set, i.e the number of objects n, the number of attributes p, the number of real classes c, and the distribution of objects in the classes.

Drift3PCA9 corresponds to the third batch proposed in the UCI Machine learning repository. It contains originally 128 attributes that can be compressed using a Principal Component Analysis without losing the essence of the data

[1] http://www.ics.uci.edu/~mlearn

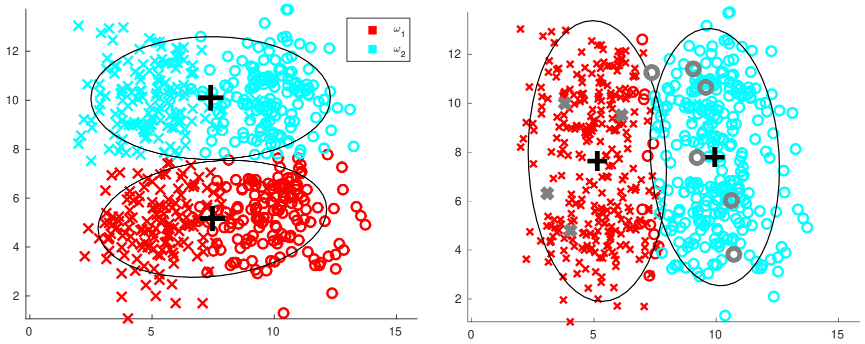


Figure 3: Final hard partition obtained with (a) PFCM-mah and (b) SPFCM-mah. Real clusters are shown using different markers and found clusters are shown using different colors, '+' markers show the cluster centroids, and labeled patterns used by SPFCM are shown using gray markers.

Table 2: Characteristics of the data sets.

	n	p	c
Banknote	1372	4	{762,610}
Drift3PCA9	1586	9	{365,490,216,240,275}
DryBean	6907	16	{2027,1322,1630,1928}
ForestFires	122	10	{63,59}
Iris	150	4	{50,50,50}
LettersVZ	3823	5	{764,752,787,786,734}
Sat	2236	36	{ 1533,703}
Vehicle	846	18	{199,217,218,212}
Wine	178	13	{59,71,48}
ToysC10	5000	2	{500 × 10 }

structure. Finally, 9 components that represent over 99% of the information on the Drift3 data set have been retained. Drybean is a selection of four types of beans from the Dry Bean original data set. The LettersVZ data set is a subset of the Letters data set where only letters from V to Z have been kept. Finally, ToysC10 is a synthetic data set generated with various Gaussians as shown in Figure 4.

4.3. Performance metrics

To compare our method, five partially supervised clustering algorithms have been selected. Table 3 resumes their properties as well as the SPFCM characteristics. The SFCM, ESFCM, and SPFCM methods have the particularity to be able to handle a Euclidean and a Mahalanobis distance.

Probabilistic partition generated by SFCM, ESFCM and possibilistic partition produced by SPCM, SRPCM, SPFCM can be transformed into a crisp partition $\hat{\mathbf{P}}$ using the maximal probabilistic/possibilistic rule. Then, since the actual partition $\mathbf{P}$ is well-known for all data sets, it is possible to compare it with

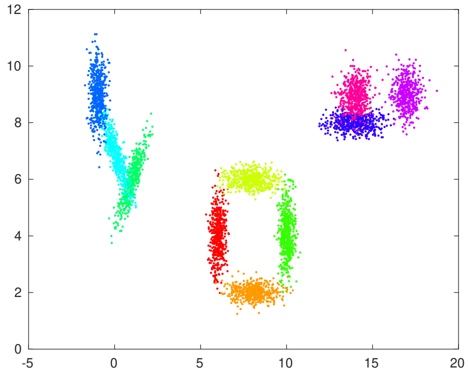


Figure 4: ToysC10 data set.

Table 3: Compared algorithms. Possible partitions are possibilistic (**T**), fuzzy (**U**) and hard (**H**) partitions.

Algorithms	Ref	Distance		Partition		
		Eucl.	Mah.	**T**	**U**	**H**
SKMEANS	[26]	X				X
SFCM	[25, 35]	X	X		X	
ESFCM	[36]	X	X		X	
SPCM	[39]	X		X		
SRPCM	[41]	X		X		
SPFCM		X	X	X	X	

the obtained crisp partition using the ARI measure [53]. A confusion matrix is also computed to deeper analyze the results. Since the data sets have more than two classes, this confusion matrix is split into several binary confusion matrices using the one-vs-all rule. Then, quantitative measures such as the *precision*, *recall* and F_1 score [54] can be calculated for each confusion matrix. The *precision* indicates the ratio between the number of objects correctly identified as positives over the total number of objects retrieved as positive:

$$P = \frac{TP}{TP + FP}, \tag{39}$$

where TP corresponds to the number of true positives and FP to the number of false positives. The *recall* is the ratio between the number of objects correctly identified as positives and the total number of objects that actually belong to the positive class.

$$R = \frac{TP}{TP + FN}, \tag{40}$$

where FN is the number of false negatives. Finally, the F_1-score is the harmonic mean of the precision and recall. In order to obtain general measures of the *precision*, *recall* and the F_1 score, a weighted average is performed using the number of objects of each class.

In order to quantify the degree of uncertainty and imprecision existing in the possibilistic clustering partition, we propose to use the u-certainty measure, also called imprecision measure or non-specificity measure [55, 56]:

$$u(t) = \left(\sum_{k=1}^{c} (\pi_k - \pi_{k+1}) \log_2(k) \right) + (1 - \pi_1) \log_2(c), \tag{41}$$

where t corresponds to the possibilistic distribution of the object i and π denotes the reordered possibilistic distribution t such that $\pi_1 \geq \pi_2 \geq \cdots \geq \pi_c$ and π_{c+1} is a variable set to 0. The measure ranges from 0 in the case of complete knowledge to $log_2(c)$ when there exists a total imprecision or uncertainty. The u-certainty measure is normalized when it is divided by $log_2(c)$. We define $unorm(\mathbf{T})$ as the average of the normalized u-certainty for the possibilistic distribution of each object.

4.4. Experimental protocol

Table 4 presents the parameters, their domains, the values set for the experiments, and the algorithms which employ them.

Table 4: Parameter settings for semi-supervised clustering algorithms.

	domain	used value	algorithms
a	$\mathbb{R}^+$	0.5	PFCM, SPFCM
b	$\mathbb{R}^+$	5	PFCM, SPFCM
η	$\mathbb{R}_{>0}$	2	PCM, PFCM, SPFCM
m	$\mathbb{R}_{>0}$	2	FCM, PFCM, SPFCM
$\{\gamma_1, \ldots, \gamma_c\}$	$\mathbb{R}_{>0}$	eq. (10)	PCM, PFCM, SPFCM
α	$\mathbb{R}^+$	1	SPFCM
$\{\rho_1, \ldots, \rho_c\}$	$\mathbb{R}_{>0}$	$\{1, \ldots, 1\}$	FCM-GK, SPFCM

For SPFCM, γ_k is fixed as explained in [10] with equation (10) and $K = 1$. The influence of parameters b and α on the clustering performance is analyzed in Sections 4.5 and 4.6. The constraints on cluster volumes ρ_k $\forall k \in \{1, \ldots, c\}$ are set to 1. Indeed, with no prior information about the clusters size, we assume that the clusters have identical volume.

Labeled patterns were defined by randomly picking objects in the data set and by assigning them their true labels with the highest possibility, except for experiments implying noisy labels, as it will be explained in section 4.7. Note that in a real application, the labels are provided by an expert. This set of labels must be able to fully cover the data space to ensure that the inherent structure of the data will be found. The role of the expert consists in verifying such property using the background knowledge of its domain.

In order to obtain a fair comparison, all tested algorithms use the same initialization for centroids and the same set of labeled patterns. Experiments consist of 100 trials with a given percentage of labeled patterns. Each trial corresponds to 5 executions of an algorithm with different centroid initializations. The partition with the minimum objective function value is then selected.

4.5. Tuning the possibilistic coefficient b

Similar to PFCM, the SPFCM algorithm needs to set its two parameters a and b. For all the data sets, we fixed $a = 0.5$ to give a relative importance to the probabilistic partition, in order to avoid coincident clusters (cf. section 2.3). Then, different values of b ranging from 0.5 to 15 are tested. High (respectively, low) values of b increase (respectively, decrease) the influence of the possibilistic partition. Tables 5 and 6 show the average ARI of SPFCM with $\alpha = 1$ obtained on each data sets. Columns show the percentage of labeled patterns whereas lines represent the b values.

Table 5: Influence of the b parameter with respect to the percentage of labeled patterns. Average ARI of SPFCM.

		Euclidean						Mahalanobis					
		5	10	15	20	25	30	5	10	15	20	25	30
Banknote	0.5	0.06	0.09	0.13	0.18	0.23	0.27	0.02	0.32	0.56	0.67	0.74	0.79
	1	0.07	0.12	0.17	0.23	0.29	0.35	0.37	0.77	0.84	0.88	0.91	0.93
	2	0.08	0.15	0.21	0.29	0.37	0.44	0.88	0.90	0.91	0.92	0.93	0.94
	5	0.09	0.16	0.25	0.34	0.43	0.49	0.90	0.91	0.92	0.93	**0.94**	0.94
	10	**0.10**	**0.17**	0.26	**0.36**	**0.44**	**0.50**	**0.91**	**0.92**	**0.93**	0.93	**0.94**	**0.95**
	15	**0.10**	**0.17**	**0.27**	**0.36**	**0.44**	**0.50**	**0.91**	**0.92**	**0.93**	**0.94**	**0.94**	**0.95**
Drift3PCA9	0.5	0.44	0.46	0.48	0.51	0.54	0.57	0.40	0.46	0.49	0.51	0.55	0.61
	1	0.44	0.47	0.49	0.53	0.57	0.61	0.44	0.53	0.61	0.66	0.73	0.80
	2	0.43	0.46	0.51	**0.59**	**0.62**	**0.65**	0.52	0.65	0.73	0.77	0.84	0.86
	5	0.42	0.52	0.56	**0.59**	**0.62**	**0.65**	0.84	**0.89**	0.87	0.86	0.85	0.87
	10	0.45	**0.56**	0.57	**0.59**	**0.62**	**0.65**	0.86	0.88	0.89	0.88	0.88	0.89
	15	**0.46**	**0.56**	**0.58**	**0.59**	**0.62**	**0.65**	**0.87**	**0.89**	**0.90**	**0.89**	**0.89**	**0.90**
DryBean	0.5	0.69	0.77	0.79	0.81	0.82	0.83	0.71	0.79	0.84	0.87	0.89	**0.90**
	1	0.76	**0.79**	0.80	**0.82**	**0.83**	0.84	0.83	0.85	**0.87**	0.88	0.88	0.89
	2	0.77	**0.79**	**0.81**	**0.82**	**0.83**	0.84	**0.84**	**0.86**	**0.87**	0.88	0.89	**0.90**
	5	**0.78**	**0.79**	**0.81**	**0.82**	**0.83**	**0.85**	**0.84**	**0.86**	**0.87**	0.88	0.89	**0.90**
	10	**0.78**	**0.79**	**0.81**	**0.82**	**0.83**	**0.85**	**0.84**	**0.86**	**0.87**	**0.89**	**0.90**	**0.90**
	15	0.77	**0.79**	**0.81**	**0.82**	**0.83**	**0.85**	**0.84**	**0.86**	**0.87**	**0.89**	**0.90**	**0.90**
ForestFires	0.5	0.66	**0.68**	**0.69**	**0.72**	**0.72**	**0.74**	0.49	0.55	0.57	0.63	0.65	0.68
	1	**0.67**	**0.68**	**0.69**	0.71	**0.72**	**0.74**	0.60	0.63	0.65	0.68	0.70	0.72
	2	**0.67**	**0.68**	**0.69**	0.70	0.71	0.72	0.63	0.66	0.68	**0.71**	0.72	0.74
	5	0.66	0.67	0.68	0.69	0.70	0.71	**0.65**	0.66	**0.69**	**0.71**	**0.73**	**0.75**
	10	0.65	0.66	0.67	0.68	0.70	0.71	**0.65**	**0.67**	**0.69**	**0.71**	**0.73**	**0.75**
	15	0.64	0.65	0.67	0.68	0.70	0.71	**0.65**	**0.67**	**0.69**	**0.71**	**0.73**	**0.75**
Iris	0.5	**0.79**	**0.80**	**0.81**	**0.82**	**0.82**	**0.83**	0.35	0.44	0.60	0.73	0.78	0.81
	1	0.78	**0.80**	**0.81**	**0.82**	**0.82**	**0.83**	0.65	0.72	0.77	0.83	0.85	0.88
	2	0.75	0.79	**0.81**	**0.82**	**0.82**	**0.83**	0.72	0.79	0.83	0.88	0.90	0.91
	5	0.76	0.79	**0.81**	0.81	0.81	**0.83**	0.76	0.85	0.88	0.90	0.91	**0.93**
	10	0.76	0.79	**0.81**	0.81	0.81	0.82	**0.77**	**0.86**	**0.89**	**0.91**	**0.92**	**0.93**
	15	0.76	0.79	0.80	0.81	0.81	0.82	**0.77**	0.85	**0.89**	**0.91**	**0.92**	**0.93**

Unsurprisingly, the accuracy of the clustering algorithm mostly increases when the percentage of labeled patterns grows. It can also be observed that the effect of the parameter b is most of the time not affected by the selected percentage of labeled patterns. Exceptions appear only when the accuracy is low, i.e. when the distance chosen does not suit. Finally, most of the time a high value of b allows to obtain a better accuracy. Only the Euclidean distance with ForestFire, Iris, and Sat seems to better work with a low value of b.

Table 6: Influence of the b parameter with respect to the percentage of labeled patterns. Average ARI of SPFCM.

		Euclidean						Mahalanobis					
		5	10	15	20	25	30	5	10	15	20	25	30
LettersVZ	0.5	0.37	0.42	0.45	0.48	0.52	0.55	0.50	0.61	0.68	0.72	0.75	0.78
	1	**0.40**	0.44	0.47	0.51	0.55	0.58	0.54	0.65	0.71	0.75	0.78	0.81
	2	**0.40**	0.45	0.49	0.53	0.57	0.60	0.57	0.69	0.75	0.79	0.82	0.84
	5	**0.40**	0.46	0.51	0.56	0.59	0.62	0.61	0.73	0.78	0.82	0.85	0.87
	10	**0.40**	**0.47**	0.52	**0.57**	**0.60**	**0.63**	0.63	**0.75**	0.80	**0.84**	0.86	**0.88**
	15	**0.40**	**0.47**	**0.53**	**0.57**	**0.60**	**0.63**	**0.64**	**0.75**	**0.81**	**0.84**	**0.87**	**0.88**
Sat	0.5	**0.85**	**0.85**	**0.86**	**0.87**	**0.88**	**0.88**	0.80	0.81	0.84	0.85	0.87	0.89
	1	0.83	0.84	0.85	0.86	0.87	**0.88**	0.82	0.84	0.88	0.90	0.92	0.94
	2	0.82	0.83	0.84	0.85	0.86	0.87	0.85	0.89	0.92	0.95	0.96	0.97
	5	0.82	0.83	0.84	0.85	0.86	0.87	0.89	0.93	0.96	0.97	**0.98**	**0.98**
	10	0.82	0.83	0.84	0.85	0.86	0.87	0.91	0.95	**0.97**	**0.98**	**0.98**	**0.98**
	15	0.82	0.83	0.84	0.85	0.86	0.87	**0.92**	**0.96**	**0.97**	**0.98**	**0.98**	**0.98**
Vehicle	0.5	0.06	0.09	0.12	**0.14**	**0.16**	**0.17**	0.03	0.08	0.18	0.26	0.33	0.41
	1	0.07	0.12	**0.13**	**0.14**	0.15	0.16	0.07	0.18	0.31	0.39	0.47	0.54
	2	0.10	**0.13**	**0.13**	0.13	0.14	0.16	0.13	0.29	0.41	0.49	0.55	0.60
	5	0.11	**0.13**	**0.13**	0.13	0.14	0.16	0.21	0.37	0.47	0.54	0.59	0.64
	10	**0.12**	**0.13**	**0.13**	**0.14**	0.15	0.16	0.25	0.39	**0.49**	0.56	0.61	**0.66**
	15	**0.12**	**0.13**	**0.13**	0.13	0.14	0.15	**0.26**	**0.40**	**0.49**	**0.57**	**0.61**	**0.66**
Wine	0.5	0.67	0.75	0.81	0.85	0.88	0.90	0.20	0.34	0.44	0.48	0.57	0.62
	1	0.84	0.88	0.90	0.90	**0.92**	0.92	0.26	0.39	0.48	0.52	0.61	0.67
	2	0.89	**0.90**	**0.91**	0.91	**0.92**	**0.93**	0.30	0.42	0.52	0.56	0.64	0.70
	5	**0.90**	**0.90**	**0.91**	**0.92**	**0.92**	**0.93**	0.35	0.46	0.55	0.60	0.67	0.71
	10	**0.90**	**0.90**	**0.91**	0.91	**0.92**	**0.93**	0.37	0.47	0.56	**0.61**	**0.68**	**0.72**
	15	0.89	**0.90**	**0.91**	0.91	**0.92**	**0.93**	**0.38**	**0.48**	**0.57**	**0.61**	**0.68**	**0.72**

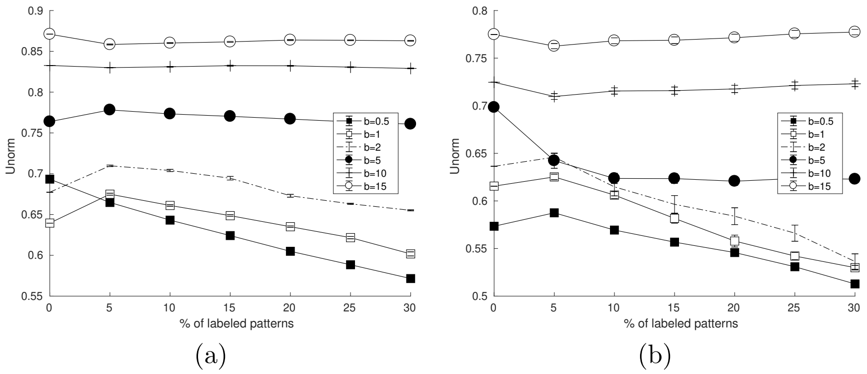
Conversely, we observed for all data sets that the highest is the value of b, the more the clustering solution contains uncertainties and imprecisions. Figure 5 illustrates this behavior on the Drift3PCA9 data set. It can be noticed that the behavior is independent of the integration of labeled patterns.

In order to obtain a good trade-off between the accuracy for the clustering partition and the degree of uncertainties/imprecision available in the final partition, we choose to set $b = 5$.

4.6. Influence of the penalty coefficient α

The penalty coefficient α, which controls the trade-off between the respect of the labeled patterns and the inherent structure of the data, is a difficult parameter to tune. Indeed, too low values make the algorithm neglect labels, and inversely, too high values force the algorithm to respect the labels but do not ensure an overall coherent solution with compact classes. Tables 7 and 8 present the mean ARI obtained with SPFCM such that $a = 0.5$, $b = 1$ and $\alpha \in [0.01\ 1]$. Columns show the percentage of labeled patterns whereas lines represent the α values.

Results show that high penalty coefficients give most of the time better accuracy. Hence, we set $\alpha = 1$ for all experiments.



(a) (b)

Figure 5: U-certainty measure obtained using SPFCM on Drif3PCA9 data set: (a) Euclidean distance and (b) Mahalanobis distance.

Table 7: Influence of the α parameter with respect to the percentage of labeled patterns. Average ARI of SPFCM.

		Euclidean						Mahalanobis					
		5	10	15	20	25	30	5	10	15	20	25	30
Banknote	0.01	0.03	0.03	0.03	0.03	0.03	0.03	0.05	0.05	0.05	0.04	0.04	0.04
	0.05	0.03	0.03	0.04	0.04	0.05	0.05	0.05	0.04	0.04	0.03	0.02	0.01
	0.1	0.03	0.04	0.05	0.06	0.07	0.08	0.04	0.04	0.02	0.01	0.14	0.61
	0.5	0.06	0.09	0.14	0.18	0.23	0.27	0.04	0.65	0.77	0.82	0.86	0.88
	1	**0.07**	**0.12**	**0.17**	**0.23**	**0.29**	**0.35**	**0.37**	**0.77**	**0.84**	**0.88**	**0.91**	**0.93**
Drift3PCA9	0.01	0.40	0.40	0.40	0.41	0.41	0.41	0.31	0.31	0.33	0.32	0.35	0.41
	0.05	0.41	0.41	0.40	0.40	0.40	0.39	0.37	0.41	0.48	0.49	0.51	0.51
	0.1	0.40	0.40	0.39	0.41	0.42	0.42	0.40	0.43	0.45	0.45	0.44	0.44
	0.5	0.41	0.45	0.47	0.49	0.52	0.55	0.42	0.47	0.52	0.59	0.66	0.74
	1	**0.44**	**0.47**	**0.49**	**0.53**	**0.57**	**0.61**	**0.44**	**0.53**	**0.61**	**0.66**	**0.73**	**0.80**
DryBean	0.01	0.62	0.62	0.62	0.62	0.62	0.62	0.61	0.62	0.64	0.66	0.67	0.68
	0.05	0.62	0.63	0.64	0.67	0.71	0.75	0.66	0.73	0.77	0.78	0.79	0.79
	0.1	0.63	0.65	0.71	0.76	0.79	0.81	0.74	0.77	0.79	0.80	0.81	0.81
	0.5	0.72	0.78	**0.80**	0.81	0.82	0.84	0.78	**0.85**	0.86	0.87	**0.88**	0.88
	1	**0.76**	**0.79**	**0.80**	**0.82**	**0.83**	**0.84**	**0.83**	**0.85**	**0.87**	**0.88**	**0.88**	**0.89**
Iris	0.01	0.74	0.74	0.75	0.75	0.75	0.75	0.65	0.66	0.66	0.66	0.66	0.66
	0.05	0.74	0.75	0.75	0.75	0.74	0.74	**0.66**	0.67	0.68	0.70	0.72	0.73
	0.1	0.74	0.74	0.74	0.75	0.76	0.76	**0.66**	0.68	0.70	0.73	0.75	0.77
	0.5	0.76	0.78	0.79	0.81	**0.82**	**0.83**	0.65	0.70	0.75	0.80	0.83	0.85
	1	**0.78**	**0.80**	**0.81**	**0.82**	**0.82**	**0.83**	0.65	**0.72**	**0.77**	**0.83**	**0.85**	**0.88**
ForestFires	0.01	0.64	0.64	0.64	0.64	0.64	0.64	0.57	0.57	0.57	0.57	0.57	0.57
	0.05	0.65	0.65	0.66	0.66	0.66	0.67	0.57	0.58	0.58	0.59	0.59	0.60
	0.1	0.65	0.66	0.66	0.67	0.67	0.68	0.57	0.58	0.59	0.60	0.61	0.62
	0.5	0.66	0.67	0.68	0.70	0.70	0.71	0.59	0.62	0.63	0.66	0.68	0.70
	1	**0.67**	**0.68**	**0.69**	**0.71**	**0.72**	**0.74**	**0.60**	**0.63**	**0.65**	**0.68**	**0.70**	**0.72**

4.7. Interest of uncertain labels

One major advantage of the SPFCM algorithm is its capacity to integrate labels expressed with degrees of possibility on classes. The following experiment shows the interest on data sets containing noisy labels.

Figure 6 presents the experimental procedure followed to add noise on la-

Table 8: Influence of the α parameter with respect to the percentage of labeled patterns. Average ARI of SPFCM.

		Euclidean						Mahalanobis					
		5	10	15	20	25	30	5	10	15	20	25	30
LettersVZ	0.01	0.28	0.29	0.32	0.35	0.35	0.35	0.42	0.48	0.51	0.54	0.57	0.59
	0.05	0.31	0.34	0.37	0.41	0.44	0.47	0.43	0.52	0.58	0.63	0.66	0.68
	0.1	0.32	0.37	0.41	0.45	0.48	0.50	0.45	0.54	0.60	0.65	0.69	0.72
	0.5	0.39	0.43	0.46	0.49	0.53	0.56	0.51	0.63	0.69	0.73	0.76	0.78
	1	**0.40**	**0.44**	**0.47**	**0.51**	**0.55**	**0.58**	**0.54**	**0.65**	**0.71**	**0.75**	**0.78**	**0.81**
Sat	0.01	0.82	0.82	0.82	0.82	0.82	0.82	0.60	0.67	0.72	0.75	0.78	0.80
	0.05	0.82	0.82	0.83	0.83	0.83	0.83	0.73	0.77	0.79	0.81	0.82	0.84
	0.1	0.82	0.83	0.83	0.84	0.84	0.85	0.77	0.78	0.80	0.82	0.83	0.85
	0.5	**0.83**	**0.84**	**0.85**	**0.86**	**0.87**	0.87	0.80	0.82	0.85	0.87	0.89	0.91
	1	**0.83**	**0.84**	**0.85**	**0.86**	**0.87**	**0.88**	**0.82**	**0.84**	**0.88**	**0.90**	**0.92**	**0.94**
Vehicle	0.01	0.06	0.06	0.06	0.06	0.06	0.06	0.00	0.00	0.00	0.00	0.00	0.00
	0.05	0.06	0.07	0.06	0.07	0.07	0.07	0.00	0.01	0.01	0.01	0.02	0.04
	0.1	0.06	0.06	0.07	0.08	0.08	0.08	0.00	0.01	0.02	0.06	0.10	0.16
	0.5	**0.07**	0.08	0.12	0.13	0.13	0.13	0.04	0.11	0.23	0.31	0.38	0.45
	1	**0.07**	**0.12**	**0.13**	**0.14**	**0.15**	**0.16**	**0.07**	**0.18**	**0.31**	**0.39**	**0.47**	**0.54**
Wine	0.01	0.72	0.72	0.72	0.73	0.73	0.73	0.06	0.10	0.16	0.21	0.28	0.33
	0.05	0.74	0.75	0.76	0.80	0.83	0.85	0.14	0.24	0.36	0.42	0.51	0.57
	0.1	0.76	0.81	0.84	0.86	0.88	0.88	0.16	0.30	0.41	0.46	0.55	0.60
	0.5	0.82	0.86	0.89	**0.90**	0.91	**0.92**	0.25	0.38	0.47	0.51	0.59	0.65
	1	**0.84**	**0.88**	**0.90**	**0.90**	**0.92**	**0.92**	**0.26**	**0.39**	**0.48**	**0.52**	**0.61**	**0.67**

bels. First, a percentage of objects is selected to create a priori knowledge (AK percentage) in the form of labels. Then, from the set of objects dedicated to being labeled, a percentage is retrieved to create noisy labels (NL percentage) and the rest remains with their true labels (TL percentage). Finally, depending on the set of labels, a different possibility value is set: TLP for true labels and NLP for noisy labels.

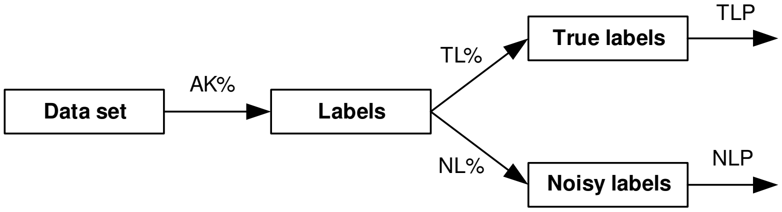


Figure 6: Noisy labels set generation.

In the experiments, AK is set either to 0, 15, or 30%, NL and TL are both fixed to 50%, TLP is equal to 1, and values tested for NLP are 1, 0.5, and 0.2. Average ARI is presented Table 9. As expected, noisy labels make decrease most of the time the accuracy of the clustering solution when NLP is set to 1. However, with a lower value of the NLP, the SPFCM algorithm manages to improve the performances compared to PFCM. Thus, NLP=0.2 gives the best results. Such an experiment shows the importance to set a low possibility value on labels when the a priori knowledge is not certain.

Table 9: Average ARI obtained with different NLP values (in lines) varying with the percentage of labeled patterns (in columns).

		Euclidean			Mahalanobis		
		(% labeled patterns)			(% labeled patterns)		
	NLP	0%	15%	30%	0%	15%	30%
Banknote	1	0.03	0.02	0.03	0.43	0.38	0.09
	0.5	0.03	0.09	0.15	0.43	0.76	0.65
	0.2	0.03	**0.11**	**0.20**	0.43	**0.87**	**0.85**
Drybean	1	0.67	0.67	0.53	0.80	0.67	0.53
	0.5	0.67	0.73	0.68	0.80	0.81	0.73
	0.2	0.67	**0.76**	**0.76**	0.80	**0.86**	**0.86**
Drift3PCA9	1	0.38	0.42	0.39	0.65	0.47	0.37
	0.5	0.38	0.42	0.49	0.65	0.72	0.68
	0.2	0.38	**0.46**	**0.53**	0.65	**0.81**	**0.83**
Forestfires	1	0.29	**0.23**	0.18	0.65	0.30	0.11
	0.5	0.29	0.21	0.21	0.65	0.40	0.33
	0.2	0.29	0.21	**0.22**	0.65	**0.58**	**0.59**
Iris	1	0.75	0.64	0.44	0.67	0.32	0.19
	0.5	0.75	0.74	0.70	0.67	0.53	0.45
	0.2	0.75	**0.78**	**0.77**	0.67	**0.75**	**0.77**
LettersVZ	1	0.28	0.34	0.31	0.17	0.41	0.43
	0.5	0.28	0.36	0.34	0.17	0.55	0.52
	0.2	0.28	**0.39**	**0.42**	0.17	**0.62**	**0.65**
Pen	1	0.35	0.50	0.46	0.33	0.66	0.59
	0.5	0.35	0.57	0.52	0.33	0.74	0.67
	0.2	0.35	**0.62**	**0.61**	0.33	**0.81**	**0.82**
Sat	1	0.83	0.68	0.40	0.69	0.06	0.02
	0.5	0.83	0.78	0.72	0.69	0.65	0.46
	0.2	0.83	**0.81**	**0.81**	0.69	**0.72**	**0.71**
Vehicle	1	0.11	0.09	0.09	0.01	0.10	0.12
	0.5	0.11	0.11	0.10	0.01	0.19	0.25
	0.2	0.11	**0.12**	**0.12**	0.01	**0.24**	**0.33**
Wine	1	0.87	0.52	0.40	0.16	0.13	0.09
	0.5	0.87	0.76	0.65	0.16	0.24	0.24
	0.2	0.87	**0.86**	**0.84**	0.16	**0.34**	**0.41**

4.8. Performance analysis

The results of the performance analysis for SPFCM carried out on each data set are shown in Tables 10 and 11. Confidence intervals are not reported since they vary between 0 and 0.01. The *precision* (P_u), *recall* (R_u) and F_1 (F_{1_u}) measures are computed using only the unlabeled objects while the Adjusted Random Index is computed with and without labeled patterns (ARI and ARI_u).

Notice that as the percentage of labeled patterns increases, the ARI_u and the F_{1_u} score also increase. Then, it is clear that labeled objects influence unlabeled objects and lead them to a solution that better fits the inherent structure of the data. Finally, results reported in Table 10 and 11 show an improvement on *precision* and *recall* measures when labeled patterns are integrated into the clustering algorithm.

Table 10: Performances of SPFCM using $a = 0.5$, $b = 5$ and $\alpha = 1$ with the respect to the percentage of labeled patterns.

		Euclidean					Mahalanobis				
		P_u	R_u	$F1_u$	ARI_u	ARI	P_u	R_u	$F1_u$	ARI_u	ARI
Banknote	0	0.60	0.59	0.59	0.03	0.03	0.81	0.80	0.80	0.48	0.48
	5	0.65	0.65	0.65	0.07	0.09	0.98	0.97	0.97	0.90	0.90
	10	0.70	0.70	0.70	0.12	0.16	0.98	0.98	0.98	0.90	0.91
	15	0.75	0.75	0.75	0.18	0.25	0.98	0.98	0.98	0.91	0.92
	20	0.79	0.79	0.79	0.25	0.34	0.98	0.98	0.98	0.92	0.93
	25	0.82	0.82	0.82	0.30	0.43	0.98	0.98	0.98	0.92	0.94
	30	0.85	0.85	0.85	0.34	0.49	0.98	0.98	0.98	0.92	0.94
Drift3PCA9	0	0.60	0.54	0.55	0.39	0.37	0.67	0.69	0.66	0.75	0.74
	5	0.82	0.72	0.74	0.46	0.46	0.89	0.87	0.87	0.84	0.83
	10	0.88	0.78	0.80	0.52	0.55	0.91	0.90	0.90	0.87	0.85
	15	0.86	0.79	0.80	0.54	0.56	0.92	0.91	0.90	0.87	0.86
	20	0.87	0.81	0.82	0.55	0.59	0.92	0.90	0.90	0.86	0.86
	25	0.88	0.82	0.84	0.56	0.62	0.92	0.91	0.91	0.87	0.87
	30	0.89	0.84	0.85	0.56	0.65	0.93	0.91	0.91	0.87	0.86
DryBean	0	0.73	0.73	0.68	0.75	0.67	0.83	0.83	0.83	0.80	0.80
	5	0.92	0.90	0.90	0.84	0.78	0.93	0.92	0.92	0.84	0.84
	10	0.92	0.91	0.91	0.85	0.79	0.94	0.93	0.93	0.85	0.86
	15	0.93	0.92	0.92	0.85	0.81	0.94	0.94	0.94	0.86	0.87
	20	0.93	0.93	0.93	0.86	0.82	0.95	0.95	0.95	0.87	0.88
	25	0.94	0.93	0.93	0.86	0.83	0.95	0.95	0.95	0.87	0.89
	30	0.94	0.94	0.94	0.86	0.85	0.96	0.95	0.95	0.88	0.90
ForestFire	0	0.90	0.89	0.89	0.66	0.62	0.92	0.90	0.90	0.61	0.64
	5	0.92	0.91	0.91	0.64	0.66	0.92	0.91	0.91	0.65	0.65
	10	0.92	0.91	0.91	0.64	0.67	0.93	0.91	0.91	0.66	0.66
	15	0.92	0.92	0.92	0.65	0.68	0.93	0.92	0.92	0.68	0.69
	20	0.93	0.92	0.92	0.63	0.69	0.93	0.93	0.93	0.68	0.71
	25	0.92	0.91	0.91	0.64	0.70	0.93	0.91	0.91	0.69	0.73
	30	0.94	0.93	0.93	0.64	0.71	0.93	0.92	0.92	0.70	0.75
Iris	0	0.89	0.89	0.89	0.73	0.75	0.86	0.85	0.85	0.73	0.66
	5	0.91	0.90	0.90	0.76	0.76	0.92	0.91	0.91	0.82	0.76
	10	0.93	0.92	0.92	0.77	0.79	0.95	0.95	0.95	0.86	0.85
	15	0.94	0.94	0.94	0.77	0.81	0.96	0.96	0.96	0.88	0.88
	20	0.93	0.92	0.92	0.77	0.81	0.97	0.97	0.97	0.90	0.90
	25	0.94	0.93	0.93	0.77	0.81	0.98	0.98	0.98	0.91	0.91
	30	0.95	0.94	0.94	0.77	0.83	0.97	0.97	0.97	0.92	0.93

4.9. Performance comparison

The SPFCM algorithm has been compared to SKMEANS [26], SFCM [25], ESFCM [36], SPCM [39], and SRPCM [41]. Mahalanobis and Euclidean distances were employed for SPFCM, SFCM, and ESFCM, giving six distinct algorithms respectively named SPFCM-mah, SFCM-mah, SFCM-eucl and SPFCM-eucl. SKMEANS, SRPCM, and SPCM are only carried out with a Euclidean distance since no extension using a Mahalanobis distance exists.

Possibilistic and fuzzy partitions were transformed into hard ones in order to compute the ARI. Figures 7 to 9 present the evolution of the mean ARI obtained according to the percentage of labeled patterns.

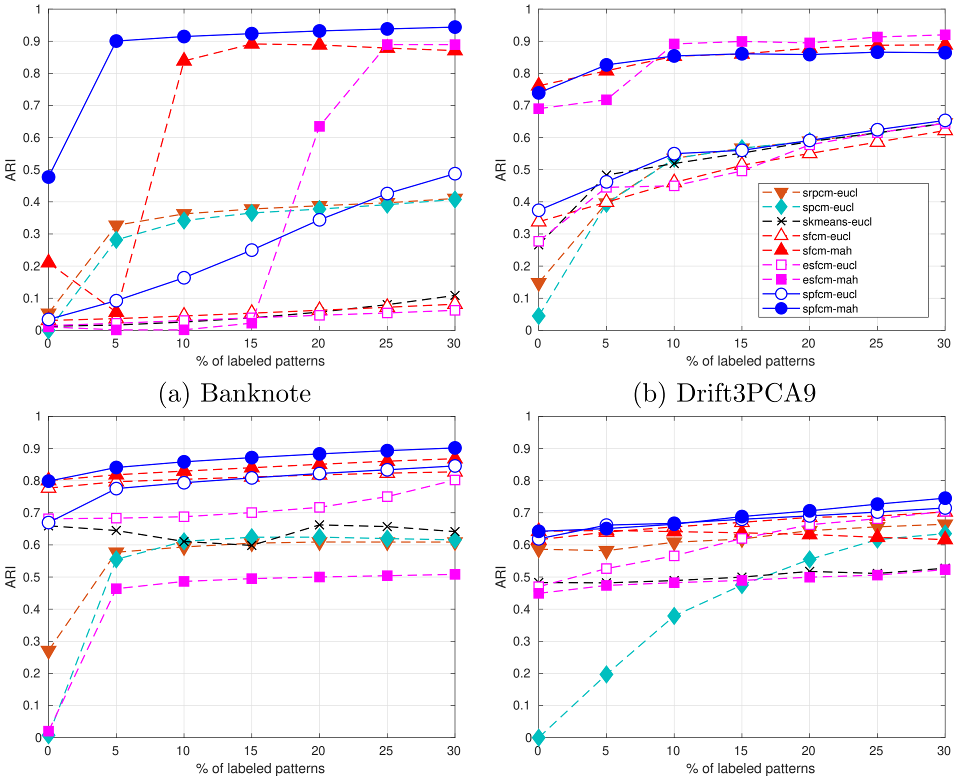
Results show that a Mahalanobis distance gives better accuracy than a Euclidean distance on most of the data sets considered in this work. This behavior

Table 11: Performances of SPFCM using $a = 0.5$, $b = 5$ and $\alpha = 1$ with the respect to the percentage of labeled patterns.

		Euclidean					Mahalanobis				
		P_u	R_u	$F1_u$	ARI_u	ARI	P_u	R_u	$F1_u$	ARI_u	ARI
LettersVZ	0	0.34	0.42	0.35	0.29	0.28	0.26	0.30	0.25	0.05	0.17
	5	0.62	0.62	0.61	0.38	0.40	0.82	0.82	0.81	0.61	0.61
	10	0.73	0.72	0.71	0.43	0.46	0.88	0.88	0.88	0.72	0.73
	15	0.76	0.76	0.75	0.46	0.51	0.91	0.91	0.91	0.77	0.78
	20	0.79	0.78	0.78	0.48	0.56	0.93	0.92	0.92	0.80	0.82
	25	0.81	0.80	0.80	0.49	0.59	0.94	0.94	0.94	0.82	0.85
	30	0.83	0.82	0.82	0.51	0.62	0.95	0.95	0.95	0.84	0.87
Sat	0	0.96	0.96	0.95	0.83	0.83	0.88	0.87	0.86	0.70	0.69
	5	0.96	0.96	0.95	0.84	0.82	0.97	0.97	0.97	0.89	0.89
	10	0.96	0.96	0.96	0.84	0.83	0.98	0.98	0.98	0.93	0.93
	15	0.96	0.96	0.96	0.84	0.84	0.99	0.99	0.99	0.96	0.96
	20	0.96	0.96	0.96	0.84	0.85	0.99	0.99	0.99	0.97	0.97
	25	0.97	0.97	0.97	0.84	0.86	0.99	0.99	0.99	0.97	0.98
	30	0.97	0.97	0.97	0.84	0.87	1.00	0.99	1.00	0.97	0.98
ToysC10	0	0.64	0.64	0.62	0.76	0.75	0.69	0.70	0.69	0.84	0.83
	5	0.92	0.92	0.91	0.87	0.86	0.95	0.95	0.95	0.92	0.92
	10	0.85	0.85	0.85	0.87	0.87	0.96	0.96	0.96	0.92	0.92
	15	0.94	0.93	0.93	0.87	0.87	0.97	0.97	0.97	0.92	0.93
	20	0.93	0.92	0.92	0.87	0.88	0.97	0.97	0.97	0.92	0.93
	25	0.91	0.90	0.90	0.88	0.88	0.97	0.97	0.97	0.92	0.93
	30	0.89	0.89	0.89	0.88	0.89	0.97	0.97	0.97	0.92	0.94
Vehicle	0	0.38	0.38	0.36	0.12	0.11	0.16	0.25	0.13	0.02	0.01
	5	0.32	0.33	0.31	0.11	0.11	0.42	0.43	0.40	0.27	0.21
	10	0.35	0.36	0.35	0.12	0.13	0.60	0.61	0.60	0.36	0.37
	15	0.36	0.37	0.35	0.11	0.13	0.71	0.71	0.70	0.43	0.47
	20	0.32	0.33	0.31	0.10	0.13	0.77	0.77	0.77	0.50	0.54
	25	0.27	0.29	0.27	0.10	0.14	0.80	0.80	0.80	0.55	0.59
	30	0.31	0.32	0.30	0.10	0.16	0.83	0.83	0.83	0.59	0.64
Wine	0	0.96	0.96	0.96	0.88	0.87	0.38	0.43	0.32	0.34	0.14
	5	0.97	0.96	0.96	0.87	0.90	0.66	0.66	0.65	0.35	0.35
	10	0.97	0.97	0.97	0.88	0.90	0.77	0.77	0.76	0.46	0.46
	15	0.97	0.97	0.97	0.88	0.91	0.83	0.82	0.82	0.55	0.55
	20	0.97	0.97	0.97	0.89	0.92	0.87	0.85	0.85	0.60	0.60
	25	0.98	0.98	0.98	0.89	0.92	0.90	0.89	0.88	0.67	0.67
	30	0.97	0.97	0.97	0.90	0.93	0.91	0.89	0.89	0.71	0.71

is simple to explain: an adaptive distance can discover clusters of different sizes and shapes whereas a Euclidean distance only makes the assumptions of spherical shapes. However, as illustrated with Wine, a Euclidean distance can sometimes outperform a Mahalanobis distance with a small number of labeled patterns. Such a situation happens with intricate data sets, for example with classes having non-geometrical shapes or highly overlapped clusters with spherical shapes.

Experiences also show that although SPFCM without labeled patterns gives mixed results, it outperforms SKMEANS and SFCM with few labeled patterns. Such behavior is clearly explained by the use of the possibilistic framework. Indeed, conversely to hard or probabilistic clustering algorithms, a possibilistic algorithm can obtain a solution with closed centroids. Without background



(a) Banknote

(b) Drift3PCA9

(c) Drybean

(d) ForestFire

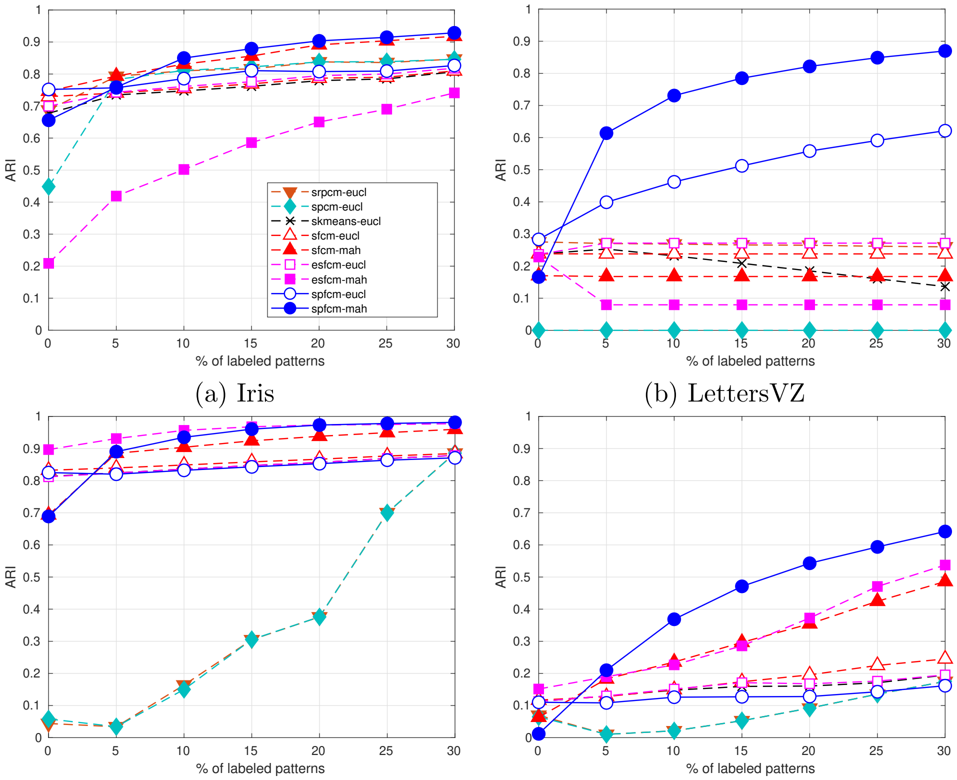
Figure 7: Average ARI and confidence interval obtained by different algorithms for (a) Banknote, (b) Drift3PCA9, (c) Drybean, and (d) ForestFire.

knowledge, such freedom might be too important and lead to the undesired merge of some clusters. However, with very few labeled patterns, this effect vanishes and allows SPFCM to find an accurate solution.

4.10. Complexity analysis

The complexity of the SPFCM algorithm has been studied by computing the CPU time and the number of iterations needed to reach convergence of the solution. Results obtained for SPFCM with parameters set to $a = 0.5$, $b = 5$, $\alpha = 1$ on a server with an Intel(R) Xeon(R) CPU E5-2670 v2 2.50 GHz processor are shown in Table 12.

As explained in section 3, the complexity of our algorithm with a Euclidean distance is identical to FCM with a Euclidean distance. When using a Mahalanobis distance, the complexity of SPFCM is equivalent to FCM with a Mahalanobis distance. However, it can be seen that the CPU time and the iterations number decrease as the number of labeled objects increases, whatever the distance used. The variability of the number of iterations also drastically



(a) Iris

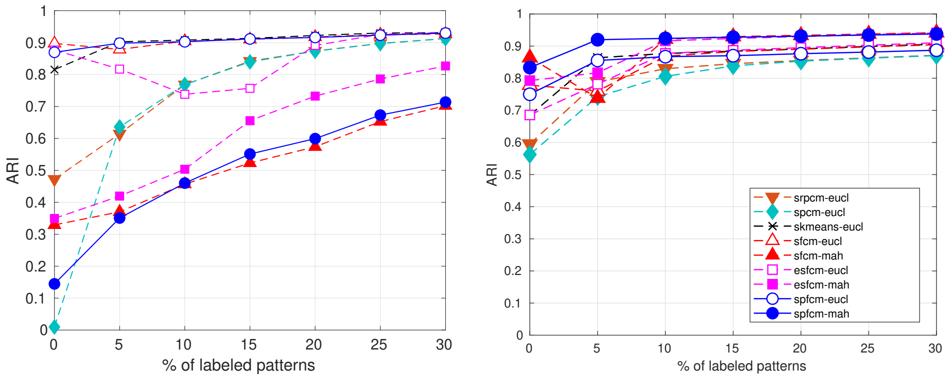
(b) LettersVZ

(c) Sat

(d) Vehicle

Figure 8: Average ARI and confidence interval obtained by different algorithms for (a) Iris, (b) LettersVZ, (c) Sat, and (d) Vehicle.

reduces with labeled patterns. Thus, the labeled patterns allow the algorithm to converge faster towards a solution.



(a) Wine

(b) ToysC10

Figure 9: Average ARI and confidence interval obtained by different algorithms for (a) Wine and (b) ToysC10.

5. Conclusions and Future Work

In this paper, a new partially supervised clustering algorithm called SPFCM has been introduced. The algorithm is an extension of PFCM that (1) incorporates partial knowledge in the form of labeled patterns in the clustering process and (2) introduces Mahalanobis distances in order to characterize clusters with ellipsoidal shapes. The SPFCM method is based on the possibilistic framework that allows expressing various types of uncertainty. Thus, we proposed to represent labeled patterns in the form of a possibility to belong to clusters in such a way as to allow an expert to include partial information into the clustering algorithm.

Conducted experiments show that adding both labeled patterns and an adaptive distance improve the performance of the algorithm. Indeed, labeled patterns guide the algorithm towards a desired solution or simply improve an already good solution while the adaptive distance estimates better the shapes of the clusters. The influence of the various parameters of SPFCM has also been studied. It turns out that most of the time labeled patterns have a low impact on the parameters coming from PFCM and unsurprisingly, a trade-off should be carefully chosen between the respect of the labels given in entry and the structure of the data. Our SPFCM algorithm has then been compared with other soft clustering handling labeled patterns and has presented good results.

This research may be extended in several directions. First, an experimental study on labeled objects having possibility to belong to several classes can be conducted. Second, developing a method to automatically set the parameters of the algorithm can be explored. Finally, in some applications, labeled patterns are not available a priori, but an expert is available to provide some labels. In this context, an active learning scheme can be developed in order to automatically select the labeled objects that will have the better impact on the clustering performance.

Table 12: Average number of iterations and CPU time ± interval confidence for SPFCM with the respect to the percentage of labeled patterns.

		Euclidean		Mahalanobis	
		CPU time(sec)	# iterations	CPU time(sec)	# iterations
Drift3PCA9	0	0.40 ± 0.05	104.82 ± 14.30	0.58 ± 0.06	120.67 ± 12.04
	5	0.33 ± 0.02	89.72 ± 6.84	0.43 ± 0.03	90.65 ± 6.44
	10	0.16 ± 0.01	45.03 ± 2.54	0.31 ± 0.02	65.15 ± 5.18
	15	0.13 ± 0.01	36.28 ± 3.32	0.27 ± 0.03	55.77 ± 5.76
	20	0.11 ± 0.01	29.97 ± 1.76	0.27 ± 0.02	56.11 ± 4.34
	25	0.09 ± 0.00	26.16 ± 1.16	0.25 ± 0.03	51.40 ± 5.54
	30	0.09 ± 0.00	23.30 ± 0.71	0.23 ± 0.02	48.50 ± 3.98
Iris	0	0.07 ± 0.01	27.48 ± 1.10	0.07 ± 0.00	35.19 ± 1.57
	5	0.04 ± 0.00	26.52 ± 1.04	0.07 ± 0.00	35.44 ± 2.29
	10	0.04 ± 0.00	20.08 ± 0.61	0.05 ± 0.00	25.93 ± 1.25
	15	0.03 ± 0.00	16.74 ± 0.36	0.04 ± 0.00	20.39 ± 0.84
	20	0.03 ± 0.00	14.90 ± 0.34	0.03 ± 0.00	16.71 ± 0.63
	25	0.02 ± 0.00	14.61 ± 0.29	0.03 ± 0.00	15.11 ± 0.50
	30	0.02 ± 0.00	12.90 ± 0.30	0.03 ± 0.00	14.02 ± 0.37
LettersVZ	0	1.10 ± 0.07	166.74 ± 11.34	0.82 ± 0.02	74.30 ± 1.43
	5	0.16 ± 0.00	23.39 ± 0.64	0.35 ± 0.00	31.75 ± 0.33
	10	0.11 ± 0.00	15.34 ± 0.24	0.28 ± 0.00	24.89 ± 0.24
	15	0.08 ± 0.00	12.12 ± 0.15	0.23 ± 0.00	20.69 ± 0.20
	20	0.07 ± 0.00	10.68 ± 0.13	0.20 ± 0.00	18.00 ± 0.18
	25	0.07 ± 0.00	9.35 ± 0.09	0.18 ± 0.00	15.74 ± 0.15
	30	0.06 ± 0.00	8.56 ± 0.09	0.16 ± 0.00	14.40 ± 0.14
Vehicle	0	0.24 ± 0.02	87.93 ± 5.57	1.11 ± 0.10	294.09 ± 26.12
	5	0.18 ± 0.03	69.32 ± 10.62	0.33 ± 0.03	86.46 ± 8.79
	10	0.13 ± 0.01	52.99 ± 3.55	0.23 ± 0.02	60.04 ± 4.93
	15	0.17 ± 0.02	69.16 ± 7.51	0.18 ± 0.01	47.59 ± 2.91
	20	0.18 ± 0.02	72.29 ± 8.77	0.16 ± 0.01	42.72 ± 3.59
	25	0.12 ± 0.01	50.01 ± 5.50	0.16 ± 0.02	41.15 ± 5.21
	30	0.14 ± 0.02	56.26 ± 10.03	0.13 ± 0.02	32.80 ± 4.61
Wine	0	0.06 ± 0.00	24.91 ± 0.90	0.08 ± 0.00	36.86 ± 1.58
	5	0.04 ± 0.00	20.94 ± 1.02	0.07 ± 0.00	34.16 ± 0.87
	10	0.03 ± 0.00	13.79 ± 0.36	0.06 ± 0.00	28.93 ± 0.52
	15	0.02 ± 0.00	11.51 ± 0.22	0.06 ± 0.00	26.58 ± 0.63
	20	0.02 ± 0.00	10.39 ± 0.18	0.05 ± 0.00	25.44 ± 0.57
	25	0.02 ± 0.00	9.59 ± 0.16	0.05 ± 0.00	24.85 ± 0.67
	30	0.02 ± 0.00	8.81 ± 0.15	0.05 ± 0.00	23.30 ± 0.73

Appendix A. Covariance matrix singularity correction

The solution to render the $\mathbf{\Sigma}_k$ invertible is detailed in Algorithm 2. It consists in transforming a positive semi-definite matrix into a positive definite matrix by adding small values to the null eigenvalues (cf. Algorithm 2, lines 3 to 7). The parameter $\mathcal{E}$, usually set to 10^5, allows specifying when an eigenvalue should be increased. Although this correction avoids the numerical problem (inversion and determinant are now calculable), the new ellipsoid might still be extremely elongated and ultimately not match with the cluster data. This occurs mainly when there is a small amount of data in the cluster. To overcome this problem, a scaled identity matrix is added to the partially supervised possibilistic covariance matrix $\mathbf{\Sigma}_k$ (cf. Algorithm 2, line 1). Finally, note that the shape tuning parameter $\kappa \in [0, 1]$ depends on the data set and should be manually adjusted [49].

Algorithm 2 Covariance singularity correction [49]

```
Require: Possibilistic covariance matrix positive semi-definite $\mathbf{\Sigma}_k$, $\mathbf{\Sigma}_0$ covariance matrix of the whole data set, shape tuning parameter $\kappa$, threshold $\mathcal{E}$.
Ensure: Possibilistic covariance matrix positive definite $\mathbf{\Sigma}_k$.
$\mathbf{\Sigma}_k \leftarrow (1-\kappa)\mathbf{\Sigma}_k + \kappa \det(\mathbf{\Sigma}_0)^{1/p}\mathbf{I}$.
Extract eigenvalues $\lambda_k = \{\lambda_{k1}, \dots, \lambda_{kp}\}$ and eigenvectors $\Phi_k = [\Phi_{k1} \dots \Phi_{kp}]$ from $\mathbf{\Sigma}_k$.
for each eigenvalue $\lambda_{k\ell}$ do
    if $\lambda_{k\ell} < \frac{\max(\lambda_k)}{\mathcal{E}}$ then
        $\lambda_{k\ell} \leftarrow \frac{\max(\lambda_k)}{\mathcal{E}}$.
    end if
end for
$\mathbf{\Sigma}_k \leftarrow [\Phi_{k1} \dots \Phi_{kp}] diag(\lambda_{k1}, \dots, \lambda_{kp})[\Phi_{k1} \dots \Phi_{kp}]$.
```

References

[1] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. Patel, A. Tiwari, M. Er, W. Ding, C. Lin, A review of clustering techniques and developments, Neurocomputing 267 (2017) 664–681.

[2] G. Gan, C. Ma, J. Wu, Data clustering: theory, algorithms, and applications, volume 20, Siam, 2007.

[3] J. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm, Computers & Geosciences 10 (1984) 191 – 203.

[4] D. Gustafson, W. Kessel, Fuzzy clustering with a fuzzy covariance matrix, in: IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes, 1979, pp. 761–766.

[5] F. Bu, An efficient fuzzy c-means approach based on canonical polyadic decomposition for clustering big data in IoT, Future Generation Computer Systems (2018).

[6] M. Xue, L. Zhou, N. Kojima, L. dos Muchangos, T. Machimura, A. Tokai, Application of fuzzy c-means clustering to PRTR chemicals uncovering their release and toxicity characteristics, Science of the Total Environment 622 (2018) 861–868.

[7] A. Zotin, K. Simonov, M. Kurako, Y. Hamad, S. Kirillova, Edge detection in mri brain tumor images based on fuzzy c-means clustering, Procedia Computer Science 126 (2018) 1261–1270.

[8] M. Horáček, P. Samec, J. Minár, The mapping of soil taxonomic units via fuzzy clustering–a case study from the outer carpathians, czechia, Geoderma 326 (2018) 111–122.

[9] L. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.

[10] R. Krishnapuram, J. Keller, A possibilistic approach to clustering, IEEE transactions on fuzzy systems 1 (1993) 98–110.

[11] M. Barni, V. Cappellini, A. Mecocci, Comments on a possibilistic approach to clustering, IEEE Transactions on Fuzzy Systems 4 (1996) 393–396.

[12] H. Yu, J. Fan, Cutset-type possibilistic c-means clustering algorithm, Applied Soft Computing 64 (2018) 401–422.

[13] H. Timm, C. Borgelt, C. Döring, R. Kruse, An extension to possibilistic fuzzy cluster analysis, Fuzzy Sets and systems 147 (2004) 3–16.

[14] N. Pal, K. Pal, J. Keller, J. Bezdek, A possibilistic fuzzy c-means clustering algorithm, IEEE Transactions on Fuzzy Systems 13 (2005) 517–530.

[15] Y. Namkoong, G. Heo, Y. Woo, An extension of possibilistic fuzzy c-means with regularization, in: IEEE International Conference on Fuzzy Systems, IEEE, 2010, pp. 1–6.

[16] N. Pal, K. Pal, J. Bezdek, A mixed c-means clustering model, in: FUZZ IEEE Intl. Conference, 1997, pp. 11–21.

[17] Q. Zheng, Y. Wang, The application of semi-supervised clustering in web services composition, in: Advances in Computer Science and Information Engineering, Springer, 2012, pp. 683–688.

[18] X. Wang, B. Qian, I. Davidson, On constrained spectral clustering and its applications, Data Mining and Knowledge Discovery 28 (2014) 1–30.

[19] A. Esmaeilian, M. Kezunovic, Prevention of power grid blackouts using intentional islanding scheme, IEEE transactions on industry applications 53 (2017) 622–629.

[20] T. Finley, T. Joachims, Supervised clustering with support vector machines, in: Intl. Conf. on Machine Learning, 2005, pp. 217–224.

[21] N. Yadav, A. Kobren, N. Monath, N. McAllum, Supervised hierarchical clustering with exponential linkage, in: Intl. Conf. on Machine Learning, 2019.

[22] H. Daume III, D. Marcu, A bayesian model for supervised clustering with the dirichlet process prior, Journal of Machine Learning Research 6 (2005) 1551–1577.

[23] V. Antoine, B. Quost, M. Masson, T. Denoeux, CEVCLUS: evidential clustering with instance-level constraints for relational data, Soft Computing 18 (2014) 1321–1335.

[24] A. Abin, H. Beigy, Active constrained fuzzy clustering: A multiple kernels learning approach, Pattern Recognition 48 (2015) 953–967.

[25] W. Pedrycz, J. Waletzky, Fuzzy clustering with partial supervision, IEEE Transactions on Systems, Man and Cybernetics 27 (1997) 787–795.

[26] S. Basu, A. Banerjee, R. Mooney, Semi-supervised clustering by seeding, in: Proceedings of 19th International Conference on Machine Learning (ICML), 2002, pp. 19–26.

[27] A. Bouchachia, W. Pedrycz, Enhancement of fuzzy clustering by mechanisms of partial supervision, Fuzzy Sets and Systems 157 (2006) 1733–1759.

[28] S. Zhong, J. Ghosh, Scalable, balanced model-based clustering, in: Proceedings of the 2003 SIAM International Conference on Data Mining, SIAM, 2003, pp. 71–82.

[29] D. Gondek, T. Hofmann, Non-redundant data clustering, Knowledge and Information Systems 12 (2007) 1–24.

[30] T. Finley, T. Joachims, Supervised k-Means Clustering, Technical Report, Cornell Computing and Information Science Technical Report, 2008.

[31] T. Finley, Supervised Clustering with Structural SVMs, Ph.D. thesis, 2008.

[32] W. Pedrycz, Algorithms of fuzzy clustering with partial supervision, Pattern Recognition Letters 3 (1985) 13–20.

[33] W. Pedrycz, G. Vukovich, Fuzzy clustering with supervision, Pattern Recognition 37 (2004) 1339–1349.

[34] D. Zhang, K. Tan, S. Chen, Semi-supervised kernel-based fuzzy c-means, in: International Conference on Neural Information Processing, SIAM, 2004, pp. 1229–1234.

[35] A. Bouchachia, W. Pedrycz, Data clustering with partial supervision, Data Mining and Knowledge Discovery 12 (2006) 47–78.

[36] Y. Endo, Y. Hamasuna, M. Yamashiro, S. Miyamoto, On semi-supervised fuzzy c-means clustering, in: IEEE Intl. Conf. FUZZ IEEE, 2004, pp. 1119–1124.

[37] S. Zeng, X. Tong, H. Nong, A study on semi-supervised fcm algorithm, Knowledge and Information Systems 35 (2013) 585–612.

[38] V. Antoine, N. Labroche, Semi-supervised fuzzy c-means variants: A study on noisy label supervision, in: International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer, 2018, pp. 51–62.

[39] L. Liu, X. Wu, Semi-supervised possibilistic fuzzy c-means clustering algorithm on maximized central distance, in: International Conference on Computer Science and Electronics Engineering, 2013, pp. 1–6.

[40] M. B. Ferraro, P. Giordani, On possibilistic clustering with repulsion constraints for imprecise data, Information Sciences 53 (2013) 63–75.

[41] V. Antoine, J. Guerrero, T. Boone, G. Romero, Possibilistic clustering with seeds, in: FUZZ IEEE Intl. Conference, Rio de Janeiro, Brazil, 2018, pp. 220–226.

[42] D. Mai, L. Ngo, L. Trinh, Advanced semi-supervised possibilistic fuzzy c-means clustering using spatial-spectral distance for land-cover classification, IEEE, 2018, pp. 4375–4380.

[43] D. Mai, L. Ngo, Semi-supervised method with spatial weights based possibilistic fuzzy c-means clustering for land-cover classification, IEEE, 2018, pp. 406–411.

[44] V. Antoine, N. Labroche, V. Vu, Evidential seed-based semi-supervised clustering, in: International Symposium on Soft Computing & Intelligent Systems, Kitakyushu, Japan, 2014, pp. 706–711.

[45] V. Antoine, J. A. Guerrero, J. Xie, Fast semi-supervised evidential clustering, International Journal of Approximate Reasoning 133 (2021) 116–132.

[46] M.-H. Masson, T. Denœux, ECM: An evidential version of the fuzzy c-means algorithm, Pattern Recognition 41 (2008) 1384–1397.

[47] N. Grira, M. Crucianu, N. Boujemaa, Active semi-supervised fuzzy clustering, Pattern Recognition 41 (2008) 1834–1844.

[48] V. Antoine, B. Quost, M.-H. Masson, T. Denoeux, CECM: Constrained evidential c-means algorithm, Computational Statistics & Data Analysis 56 (2012) 894–914.

[49] R. Babuka, P. Van der Veen, U. Kaymak, Improved covariance estimation for gustafson-kessel clustering, in: Proceedings of the IEEE International Conference on Fuzzy Systems, volume 2, 2002, pp. 1081–1085.

[50] A. Bensaid, J. Bezdek, Partial supervision based on point-prototype clustering algorithms, in: Proceedings of the 4th European congress on intelligent techniques and soft computing, 1996, pp. 1402–1406.

[51] L. Vendramin, M. Naldi, R. Campello, Fuzzy clustering algorithms and validity indices for distributed data, in: Partitional Clustering Algorithms, Springer, 2015, pp. 147–192.

[52] J. F. Kolen, T. Hutcheson, Reducing the time complexity of the fuzzy c-means algorithm, IEEE Transactions on Fuzzy Systems 10 (2002) 263–267.

[53] L. Hubert, P. Arabie, Comparing partitions, Journal of classification 2 (1985) 193–218.

[54] C. Goutte, E. Gaussier, A probabilistic interpretation of precision, recall and f-score, with implication for evaluation, in: European conference on information retrieval, Springer, 2005, pp. 345–359.

[55] M. Higashi, G. Klir, Measures of uncertainty and information based on possibility distributions, International Journal of General Systems 9 (1982) 43–58.

[56] M. Higashi, G. Klir, On the notion of distance representing information closeness: Possibility and probability distributions, International Journal of General Systems 9 (1983) 103–115.