Advances in Possibilistic Clustering with Application to Hyperspectral Image Processing

S.D. Xenaki*

Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, GR-15236, Athens, Greece
Institute for Astronomy, Astrophysics, Space Applications and Remote Sensing, National Observatory of Athens, GR-15236, Penteli, Greece
ixenaki@di.uoa.gr or ixenaki@noa.gr

Abstract. Clustering is a well established data analysis methodology that has been extensively used in various fields of applications during the last decades. The main focus of the present thesis is on a well-known cost-function optimization-based family of clustering algorithms, called Possibilistic C-Means (PCM) algorithms. Specifically, the shortcomings of PCM algorithms are exposed and novel batch and online PCM schemes are proposed to cope with them. These schemes rely on (i) the adaptation of certain parameters which remain fixed during the execution of the original PCMs and (ii) the adoption of sparsity. The incorporation of these two characteristics renders the proposed schemes: (a) capable, in principle, to reveal the true number of physical clusters formed by the data, (b) capable to uncover the underlying clustering structure even in demanding cases, where the physical clusters are closely located to each other and/or have significant differences in their variances and/or densities, and (c) immune to the presence of noise and outliers. Moreover, theoretical results concerning the convergence of the proposed algorithms, also applicable to the classical PCMs, are provided. The potential of the proposed methods is demonstrated via extensive experimentation on both synthetic and real data sets. In addition, they have been successfully applied on the challenging problem of clustering in HyperSpectral Images (HSIs). Finally, a feature selection technique suitable for HSIs has also been developed.

1 Introduction and Related Work

Clustering is a well established data analysis methodology that lie in the framework of pattern recognition and it has been extensively used in various fields of applications during the last decades. Given a set of *objects*, the aim of clustering is the identification of groups (*clusters*) formed by "similar" objects. A great amount of work reported in the clustering literature has been devoted to the identification of compact and hyperellipsoidally shaped clusters. Each such

* Dissertation Advisor: Sergios Theodoridis, Professor

cluster is represented by a vector called *cluster representative* or simply *representative*, which lies in the same l-dimensional space with the data and (ideally) is located at the *center* of the cluster.

The most well-known algorithms that deal with this problem, belong to the family of cost optimization clustering algorithms and are the *k-means* (hard clustering), e.g. [1], the *fuzzy c-means* (FCM - fuzzy clustering), e.g. [2], [3] and the *possibilistic c-means* (PCM - possibilistic clustering), e.g. [4], [5], [6], [7], [8]. The main goal of all these algorithms is to move iteratively the representatives towards the centers of the regions that are *dense in data points* (dense regions), that is, to regions where significant aggregations of data points (clusters) exist. Under this perspective, we say that each such vector *represents* a cluster, while their movement towards the centers of the clusters is carried out via the minimization of appropriately defined cost functions.

Let us consider first the k-means and FCM, which share some significant features. First of all, they both require prior knowledge of the exact number of clusters m underlying in the data set (which, of course, is rarely known in practice). In addition, in both schemes the updating equations of the representatives are interrelated. As a result, these algorithms *impose* a specific clustering structure on the data set (rather than uncovering the underlying one), in the sense that they will return m clusters *irrespective* of the actual number of physical clusters existing in the data set. Specifically, if m is less than the actual number of clusters, at least some representatives will fail to move to dense regions, while in the opposite case, some naturally formed clusters will split into more than one pieces[1]. A common method for estimating m is via the use of suitable validity indices (e.g., [8]). Finally, as shown in [6], [7], k-means and FCM are vulnerable to noisy data and outliers.

As far as the PCM algorithms are concerned, the cluster representatives are updated, based on the *degrees of compatibility* of the data vectors with the clusters. In contrast to FCM and k-means, in PCM algorithms, the degrees of compatibility of a data vector with the various clusters are mutually independent. A direct consequence of this fact is that even if the number of clusters is overestimated, in principle, all representatives will be driven to dense regions, making thus feasible the uncovering of the actual clusters. However, in this case, the scenario where two or more cluster representatives are led to the same dense in data region, may arise, which, however, can be faced after the termination of the algorithm by seeking for (almost) coincident representatives. In addition, PCM deals well with noisy data points and outliers, compared to k-means and FCM. However, it involves additional parameters, usually denoted by γ. Each of these parameters is associated with a single cluster, while their accurate estimation is of crucial importance. Since, once they have been estimated they are kept fixed during the execution of the PCM algorithm, it is clear that poor initial estimates are likely to lead to poor clustering performance, especially in more

[1] Of course, if the value of m corresponds to the actual number of physical clusters, the algorithms have the ability to recover the physical clusters; that is, in this case "imposition" coincides with "uncovering".

demanding data sets (e.g. where clusters with significantly different variances are encountered in the data set).

2 Dissertation Summary

The present thesis focuses on the Possibilistic C-Means (PCM) algorithms. Specifically, exposing first their shortcomings, they are extended next, in order to overcome them. These extensions rely on the adoption of the parameter adaptivity and the sparsity concepts. In the sequel, the main contributions of the present thesis are briefly exposed.

First, a novel approach in the context of possibilistic clustering algorithms, named *Adaptive Possibilistic C-Means* (APCM) has been developed [9], [10]. APCM addresses several of the weaknesses of original PCM, by allowing the *adaptation* of some parameters that are characteristic to all PCM algorithms, during its execution. This is in contrast to classical PCM algorithms where these parameters, once they are set, they remain fixed. This characteristic of APCM gives rise to two new features that are not met in classical PCM algorithms. The first one is that APCM is capable, in principle, to reveal the true number of physical clusters, provided that it starts with a reasonable overestimate of it, thus overcoming a long-standing issue in the clustering literature. This is carried out by removing the clusters that gradually become obsolete (i.e., the clusters whose characteristic parameter diminishes towards zero as the algorithm evolves). The other feature resulting from the adaptation of the characteristic parameters of APCM is the increase of its flexibility in following the variations in the formation of the clusters during the algorithm execution. This makes APCM able to uncover the underlying clustering structure, even in demanding cases, where the physical clusters are closely located to each other and/or have significant differences in their variances. APCM is compared against several related state-of-the-art algorithms through extensive simulations on both synthetic and real data and the provided results show that APCM exhibits superior performance in almost all the considered data sets. Moreover, theoretical results that are indicative of the convergence behavior of the algorithm are also provided.

Next, we extended PCM by introducing the concept of *sparsity*. The rationale behind this extension is that, in practice, a data point is most compatible with at most one, a few or even none cluster (outlier). Thus, taking into account the data points that are most compatible with a given cluster and excluding those that are not compatible with it, leads to more accurate estimations of the clusters' parameters. The resulting algorithm, called *Sparse Possibilistic C-Means* (SPCM) [11] can deal well with closely located clusters that may also be of significantly different densities, while at the same time it exhibits immunity to noise and outliers. Finally, a non-trivial convergence proof for the SPCM algorithm is conducted [12]. The main source of difficulty in the provided convergence analysis, compared to those given for previous possibilistic algorithms, relies on the fact that one of its updating parameter equations is not given in closed form but is computed via a two-branch expression, which defines a non-continuous

mapping. In this thesis, it is shown that SPCM will converge to one of the local minima of its associated cost function. As a side effect, it is shown that similar convergence results can be derived for the PCM algorithm, viewed as a special case of SPCM, which are stronger than those established in previous works.

In the sequel, the main features of the proposed APCM and SPCM algorithms are combined giving rise to the *Sparse Adaptive Possibilistic C-Means* (SAPCM) algorithm [13], [11], which, inheriting all the advantages of its ancestors, has the ability to (a) cope well with demanding data sets with closely located physical clusters with possibly different densities and/or variances, (b) determine the number of physical clusters and (c) improve even more the estimates of the clusters' parameters, compared to APCM and SPCM. Extensive experimentation verified the overall advantages of SAPCM compared to other related algorithms. Moreover, two variants of SAPCM, which use the above original SAPCM algorithm as a building block, have been devised. The first one is an iterative bottom-up version, called *Sequential SAPCM* (SeqSAPCM) [14], which, at each iteration, determines a single new cluster by employing SAPCM. Thus, it unravels sequentially the underlying clustering structure. The basic advantage of SeqSAPCM is that it does not require knowledge of the number of physical clusters (not even a crude overestimate, as is the case with APCM, SPCM and SAPCM). The second variant of SAPCM is called *Layered SAPCM* (L-SAPCM) [15] and works in layers. Specifically, the SAPCM algorithm is initially applied in the whole data set and then it is recursively applied individually on each resulting cluster, in order to reveal possible clustering structure within it, working in a tree structure basis. L-SAPCM terminates when none of the clusters resulting so far has further clustering structure within it. As is verified by the experimental results, L-SAPCM can provide accurate clustering even in cases where the data form closely located clusters at various "resolutions", i.e. the variances of the clusters may differ orders of magnitude from each other.

Also, a considerable contribution of this thesis is the development of an online version of the APCM algorithm, called *Online APCM* (O-APCM) [16], which processes data points one by one and memorizes their impact to suitably defined accumulating variables. O-APCM embodies three new procedures for (a) generating, (b) merging or (c) deleting clusters dynamically and it is a good candidate for clustering of big data sets, whose size and dimensionality are prohibitive for batch algorithms. Finally, it is highlighted that O-APCM may be utilized for applications in both stationary, as well as dynamically varying environments, where the physical clusters may change their location in data space over time. Specifically, O-APCM has the ability to weight more heavily the most recent data, compared to older data, in the estimation of its parameters. Experimental results show that O-APCM offers high discrimination ability at a very low computational cost for data sets in stationary conditions and, additionally, it is able to track with high accuracy the physical clusters at a non-stationary environment. Finally, the application of O-APCM to a real video data set, in order to identify and track moving objects, highlights its great potential in monitoring the evolution of dynamically varying phenomena.

The potential of the proposed methods is also demonstrated via experimentation on the basis of three case studies, concerning real hyperspectral images (HSIs). The images have been collected from different hyperspectral sensors and depict various land cover cases. The proposed algorithms gave, in general, superior performance compared to other related algorithms.

Finally, a sparsity-aware feature selection technique suitable for HSIs has been developed in the frame of the current thesis [17]. The proposed method is based on the optimization of a sparsity promoting cost-function, in order to identify the bands with the most significant ability in discriminating the various homogeneous regions in the HSI under study. Experimental results on real HSI data have shown remarkable quality of the clustering considering only the selected bands that result from the above technique.

3 Results and Discussion

In the sequel, we describe in detail one of the proposed possibilistic clustering algorithms, that is the Sparse Adaptive Possibilistic C-Means (SAPCM), that incorporates the idea of adaptivity and sparsity.

3.1 Sparse Adaptive Possibilistic C-Means Algorithm

The SAPCM algorithm stems from the optimization of the cost function

$$J(\Theta, U) = \sum_{j=1}^{m} \left[\sum_{i=1}^{N} u_{ij} \|\mathbf{x}_i - \boldsymbol{\theta}_j\|^2 + \gamma_j \sum_{i=1}^{N} (u_{ij} \ln u_{ij} - u_{ij}) \right] + \lambda \sum_{i=1}^{N} \|\mathbf{u}_i\|_p^p \quad (1)$$

where $u_{ij} > 0$, $i = 1, ..., N$, $j = 1, ..., m$, the parameter γ_j is related to the "size" of jth cluster, C_j, and it could be described as a measure of its variance around its $\boldsymbol{\theta}_j$, and λ is a parameter that controls the degree of the imposed sparsity.

In SAPCM, the parameters γ, after their initialization, are properly adapted as the algorithm evolves. In particular, the parameter γ of each specific cluster is updated based only on those data vectors that are "*most compatible*" with this cluster. The proposed SAPCM algorithm stems from the optimization of the cost function of eq. (1), by setting

$$\gamma_j = \frac{\hat{\eta}}{\alpha} \eta_j \quad (2)$$

with η_j being a measure of the mean absolute deviation of C_j as it has been formed in the current iteration, $\hat{\eta}$ is a constant defined as the minimum among all initial η_j's, i.e., $\hat{\eta} = \min_{j=1,...,m_{ini}} \eta_j$, where m_{ini} is the initial number of clusters, and α is a user-defined positive parameter, so that the ration $\hat{\eta}/\alpha$ approximates the mean absolute deviation of the smallest physical cluster. Note that although the latter quantity is fixed for a given data set, it is unknown in practice.

Initialization in SAPCM: First, we make an overestimation, denoted by m_{ini}, of the true number of natural clusters m, formed by the data points; that is, we

begin with m_{ini} $\boldsymbol{\theta}_j$'s and their corresponding η_j's. Regarding $\boldsymbol{\theta}_j$'s and η_j's, their initialization drastically affects the final clustering result in SAPCM. Recalling that SAPCM is a possibilistic-type algorithm and these algorithms move the cluster representatives towards "dense in data points" regions (physical clusters), care should be taken so that at least one representative lies "close" to each physical cluster with its associated η_j being initialized suitably. Thus, a good starting point for them is of crucial importance. To this end, the initialization of $\boldsymbol{\theta}_j$'s is carried out using the final cluster representatives obtained from the FCM algorithm, when the latter is run with m_{ini} clusters. Taking into account that FCM is very likely to drive the representatives to dense in data regions (since $m_{ini} > m$), the probability that at least one of the initial $\boldsymbol{\theta}_j$'s is placed in each dense region (cluster) of the data set, increases with m_{ini}.

After the initialization of $\boldsymbol{\theta}_j$'s, η_j's are initialized as follows:

$$\eta_j = \frac{\sum_{i=1}^{N} u_{ij}^{FCM} \|\mathbf{x}_i - \boldsymbol{\theta}_j\|}{\sum_{i=1}^{N} u_{ij}^{FCM}}, \quad j = 1, \ldots, m_{ini}, \tag{3}$$

where $\boldsymbol{\theta}_j$'s and u_{ij}^{FCM}'s in eq. (3) are the final parameter estimates obtained by FCM. Combining eqs. (2), (3), the initialization of γ_j's is completely defined.

Parameter adaptation in SAPCM: In SAPCM algorithm, all parameters are adapted during its execution. More specifically, this refers to, (a) the parameters $\boldsymbol{\theta}_j$'s, (b) the parameters u_{ij}'s, (c) the number of clusters, m, and (d) the parameters γ_j's, with (c) and (d) being achieved through two interrelated processes. Minimization of $J(\Theta, U)$ with respect to $\boldsymbol{\theta}_j$ leads to the same updating equation as in the original PCM scheme, that is

$$\boldsymbol{\theta}_j = \frac{\sum_{i=1}^{N} u_{ij} \mathbf{x}_i}{\sum_{i=1}^{N} u_{ij}} \tag{4}$$

Taking the derivative of $J(\Theta, U)$ with respect to u_{ij}, we obtain

$$\frac{\partial J(\Theta, U)}{\partial u_{ij}} \equiv f(u_{ij}) = d_{ij} + \gamma_j \ln u_{ij} + \lambda p u_{ij}^{p-1}, \tag{5}$$

where $d_{ij} = \|\mathbf{x}_i - \boldsymbol{\theta}_j\|^2$. Obviously, $\frac{\partial J(\Theta, U)}{\partial u_{ij}} = 0$ is equivalent to $f(u_{ij}) = 0$, the solution of which will give the requested u_{ij}. Clearly, this equation cannot be solved analytically. However, it can be efficiently solved arithmetically based on the following propositions.

Proposition 1 *$f(u_{ij})$ does not become zero for $u_{ij} \in (-\infty, 0) \cup (1, +\infty)$* [2].

Proposition 2 *The stationary points of $f(u_{ij})$ are $\hat{u}_{ij} = \left[\frac{\lambda}{\gamma_j} p(1-p)\right]^{\frac{1}{1-p}}$ and $\tilde{u}_{ij} = +\infty$.*

[2] The proofs of Propositions 1 to 6 are given in the dissertation.

Proposition 3 *The unique minimum of* $f(u_{ij})$ *appears at* $\hat{u}_{ij} = \left[\frac{\lambda}{\gamma_j} p(1-p)\right]^{\frac{1}{1-p}}$.

Proposition 4 *If* $f(\hat{u}_{ij}) < 0$ *then* $f(u_{ij}) = 0$ *has exactly two solutions* $u_{ij}^{\{1\}}$, $u_{ij}^{\{2\}} \in (0,1)$ *with* $u_{ij}^{\{1\}} < u_{ij}^{\{2\}}$.

Proposition 5 *If* $f(u_{ij}) = 0$ *has two solutions* $u_{ij}^{\{1\}}$, $u_{ij}^{\{2\}}$ *(with* $u_{ij}^{\{1\}} < u_{ij}^{\{2\}}$*),* $J_{SPCM}(\Theta, U)$ *exhibits a local minimum at the largest of them* ($u_{ij}^{\{2\}}$).

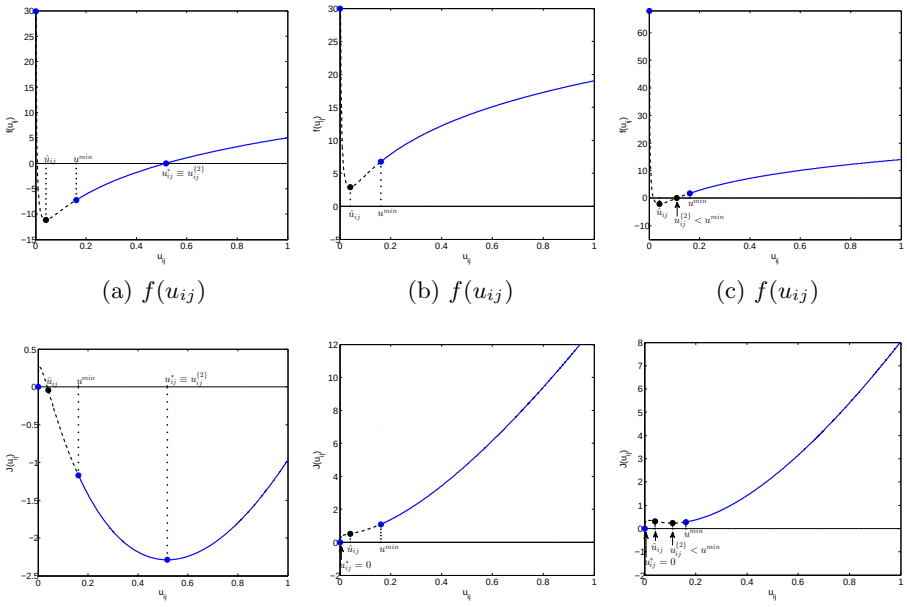
Proposition 6 $J_{SPCM}(\Theta, U)$ *exhibits its global minimum (with respect to* u_{ij}*) at* u_{ij}^**, where:*

$$u_{ij}^* = \begin{cases} u_{ij}^{\{2\}}, \text{ if } f(\hat{u}_{ij}) < 0 \text{ and } u_{ij}^{\{2\}} > \left(\frac{\lambda(1-p)}{\gamma_j}\right)^{\frac{1}{1-p}} \ (\equiv u_{min}) \\ 0, \qquad\qquad \text{otherwise} \end{cases} \tag{6}$$

Based on the above propositions, to determine u_{ij}^*, we solve $f(u_{ij}) = 0$ as follows. First, we determine $\hat{u}_{ij}$ and check whether $f(\hat{u}_{ij}) > 0$. If this is the case, then $f(u_{ij})$ has no roots in $[0,1]$. Note that, in this case, it is $f(u_{ij}) > 0$ for all $u_{ij} \in (0,1]$, since $f(\hat{u}_{ij}) > 0$ (Fig. 1b). Thus, J_{SPCM} is increasing with respect to u_{ij} in $(0,1]$ (Fig. 1e). Consequently, in this case we set $u_{ij}^* = 0$, *imposing sparsity*. In the rare case, where $f(\hat{u}_{ij}) = 0$, we set $u_{ij}^* = 0$, as $\hat{u}_{ij}$ is the unique root of $f(u_{ij}) = 0$ and $f(u_{ij}) > 0$ for $u_{ij} \in (0, \hat{u}_{ij}) \cup (\hat{u}_{ij}, 1]$. If $f(\hat{u}_{ij}) < 0$, then $f(u_{ij}) = 0$ has exactly two solutions that both lie in $[0,1]$ (see Figs. 1a, 1c). In order to determine the largest of the solutions ($u_{ij}^{\{2\}}$), we apply the bisection method (e.g. [18]) in the range $(\hat{u}_{ij}, 1]$, as $u_{ij}^{\{2\}}$ is greater than $\hat{u}_{ij}$. The bisection method is known to converge very rapidly to the optimum u_{ij}, that is, in our case, to the largest of the two solutions of $f(u_{ij}) = 0$. If the obtained solution $u_{ij}^{\{2\}}$ satisfies the rightmost condition in the first branch of eq. (6), then we set $u_{ij}^* = u_{ij}^{\{2\}}$ (Fig. 1d) . Otherwise, u_{ij}^* is set to 0 (see Fig. 1f).

Concerning the adjustment of the number of clusters $m(t)$ at the tth iteration, we proceed as follows. Let *label* be a N-dimensional vector, whose ith element is the index of the cluster which is *most compatible* with $\mathbf{x}_i$, that is the index j for which $u_{ij}(t) = \max_{r=1,\ldots,m(t)} u_{ir}(t)$. At each iteration of the algorithm, the adjustment (reduction) of the number of clusters $m(t)$ is achieved by examining, for each cluster C_j, if its index j appears at least once in the vector *label* (i.e. if there exists at least one vector $\mathbf{x}_i$ that is most compatible with C_j). If this is the case, C_j is preserved. Otherwise, C_j is eliminated and, thus, U and Θ are updated accordingly. As a result, the current number of clusters $m(t)$ is reduced.

Finally, concerning γ_j's and in contrast to the classical PCM where they are kept fixed, in SAPCM they are given by eq. (2) and are *adapted* at each iteration of the algorithm through the adaptation of the corresponding η_j's. More specifically, we propose to compute the parameter η_j of a cluster C_j at



(a) $f(u_{ij})$ (b) $f(u_{ij})$ (c) $f(u_{ij})$

(d) $J(u_{ij})$ (e) $J(u_{ij})$ (f) $J(u_{ij})$

Fig. 1: In all plots the dashed parts of the graphs correspond to the interval $(0, u_{min})$, which is not accessible by the algorithm (see eq. (6)). (a) The shape of function $f(u_{ij})$, when $f(\hat{u}_{ij}) < 0$ and the right-most condition of eq. (6) is satisfied and (d) the corresponding shape of the cost function $J(u_{ij})$. (b) The shape of function $f(u_{ij})$, when $f(\hat{u}_{ij}) > 0$ and (e) the corresponding shape of $J(u_{ij})$. (c) The shape of function $f(u_{ij})$, when $f(\hat{u}_{ij}) < 0$ and the right-most condition of eq. (6) is not satisfied and (f) the corresponding shape of $J(u_{ij})$.

each iteration, as the *mean absolute deviation* of the most compatible to cluster C_j data vectors, i.e.,

$$\eta_j(t+1) = \frac{1}{n_j(t)} \sum\nolimits_{\mathbf{x}_i : u_{ij}(t) = \max_{r=1,\ldots,m(t+1)} u_{ir}(t)} \|\mathbf{x}_i - \boldsymbol{\mu}_j(t)\|, \tag{7}$$

where $n_j(t)$ denotes the number of the data points $\mathbf{x}_i$ that are most compatible with C_j at iteration t and $\boldsymbol{\mu}_j(t)$ the mean vector of these data points.

Selection of parameter λ: As it follows from the previous analysis, considering a specific data point $\mathbf{x}_i$ and a cluster C_j, a necessary condition in order for the equation $f(u_{ij}) = 0$ to have a solution is $f(\hat{u}_{ij}) < 0$, which, taking into account eq. (5) and solving with respect to λ gives $\lambda < \frac{\gamma_j}{p(1-p)} \exp\left(-1 - \frac{d_{ij}(1-p)}{\gamma_j}\right)$. Consequently, selecting

$$\lambda \geq \frac{\gamma_j}{p(1-p)} \exp\left(-1 - \frac{d_{ij}(1-p)}{\gamma_j}\right), \tag{8}$$

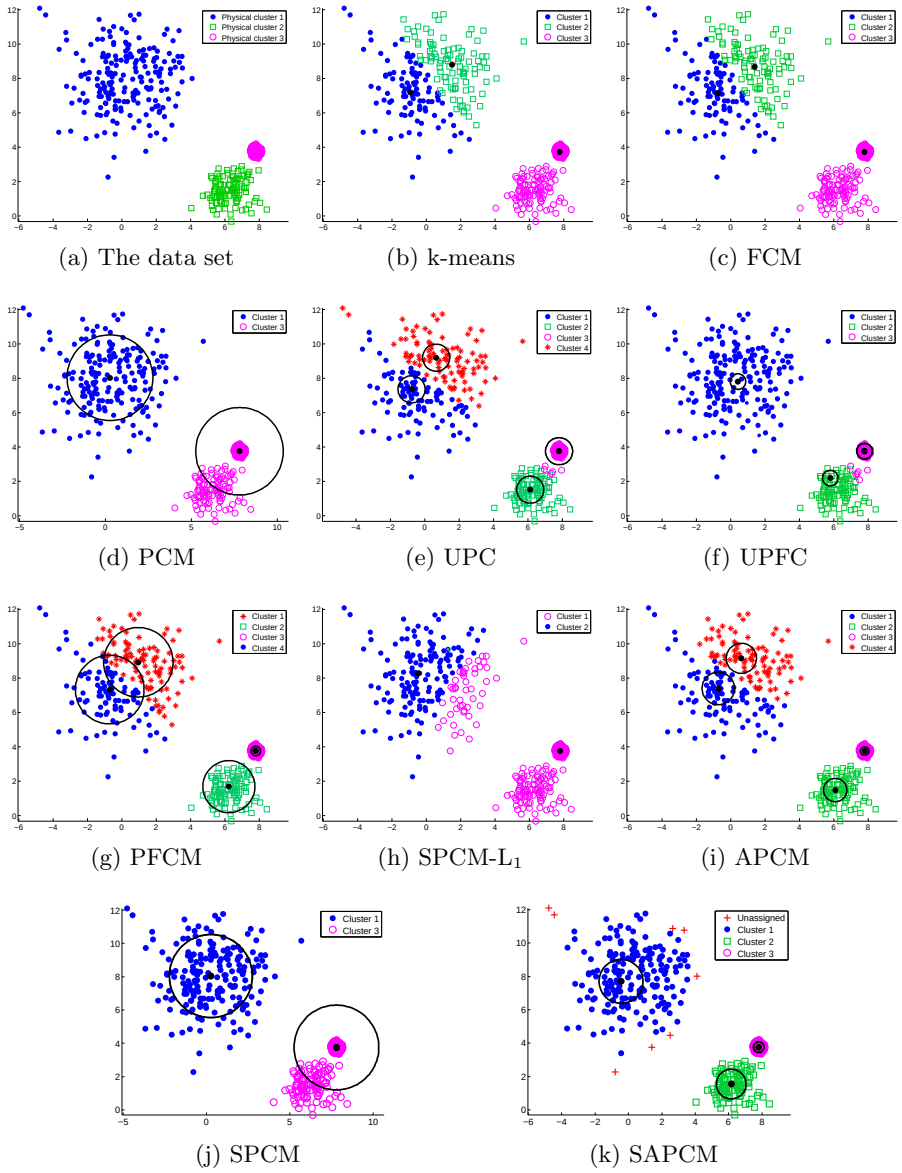
the degree of compatibility u_{ij} of a data point $\mathbf{x}_i$ with a cluster C_j is set to 0, promoting sparsity. Aiming at retaining the smallest sized cluster, say C_q (i.e., the cluster with $\gamma_q = \min_{j=1,\ldots,m} \gamma_j$) until the termination of the algorithm (provided of course that at least one representative has been initially placed in it), a reasonable choice for λ would be the one for which u_{ij} becomes 0 for points $\mathbf{x}_i$ that lie at distance d_{iq} greater than γ_q from the representative $\boldsymbol{\theta}_q$. In this way, $\boldsymbol{\theta}_q$ will be less likely to be "attracted" by nearby larger clusters, aiding it to remain in the region of the physical cluster where it was first placed. This is so because the cluster representative will be affected only by the data points that are very close to it (i.e., points with $d_{iq} < \gamma_q = \min_{j=1,\ldots,m} \gamma_j$).

To this end, applying inequality (8) for d_{ij} and γ_j equal to $\gamma_q = \min_{j=1,\ldots,m} \gamma_j$, we end up with $\lambda \geq \frac{\gamma_q}{p(1-p)\mathrm{e}^{2-p}}$, where e is the base of natural logarithm. In practice, we select λ as

$$\lambda = K \frac{\min_{j=1,\ldots,m} \gamma_j}{p(1-p)\mathrm{e}^{2-p}}, \tag{9}$$

where if we set $K = 1$, we allow non-zero u_{ij}'s for points that lie at distance around γ_q from $\boldsymbol{\theta}_q$. In most of the experiments of SAPCM, we take $K = 0.1$.

Comparison of APCM with state-of-the-art clustering algorithms In this section, we compare the clustering performance of SAPCM with that of the k-means, the FCM, the PCM [5], the UPC [8], the UPFC [19], the PFCM [7], the SPCM-L_1 [20], the APCM [10] and the SPCM [11] algorithms, which all result from cost optimization schemes. For a fair comparison, the representatives $\boldsymbol{\theta}_j$'s of all algorithms (except for SPCM-L_1) are initialized based on the FCM scheme and the parameters of each algorithm are first fine tuned. Moreover, in PCM, UPC, UPFC, PFCM and SPCM, duplicate clusters are removed after their termination. In order to compare a clustering with the true data label information, we utilize (a) the Success Rate (SR) of each physical cluster ($\mathrm{SR}_{c_j}, j = 1, ..., m$), which measures the percentage of the points of each physical cluster that have been correctly labeled by each algorithm, (b) the mean of the Euclidean distances (MD) between the true mean of each physical cluster and its closest cluster representative obtained by each algorithm, (c) the number of iterations (Iter) and (d) the total time required (Time) for the convergence of each algorithm. **Experiment**: Consider a two-dimensional data set consisting of $N = 5300$ points, where three clusters C_1, C_2 and C_3 are formed. Each cluster is modelled by a normal distribution. The means of the distributions are $\mathbf{c}_1 = [0.27, 7.99]^T$, $\mathbf{c}_2 = [6.28, 1.49]^T$ and $\mathbf{c}_3 = [7.81, 3.76]^T$, respectively, while their covariance matrices are set to $3 \cdot I_2$, $0.5 \cdot I_2$ and $0.01 \cdot I_2$, respectively. A number of 200 points are generated by the first distribution, 100 points are generated by the second one and 5000 points are generated by the third one. Note that C_2 and C_3 clusters are very close to each other and they have a big difference in their variances (see Fig. 2a). Also, note the difference in the density among the three clusters.



(a) The data set (b) k-means (c) FCM

(d) PCM (e) UPC (f) UPFC

(g) PFCM (h) SPCM-L_1 (i) APCM

(j) SPCM (k) SAPCM

Fig. 2: (a) The data set of Experiment. Clustering results for (b) k-means, $m_{ini} = 3$, (c) FCM, $m_{ini} = 3$, (d) PCM, $m_{ini} = 5$, (e) UPC, $m_{ini} = 5$, $q = 1.5$, (f) UPFC, $m_{ini} = 10$, $\alpha = 5$, $\beta = 1$, $q = 2.2$, $n = 3$, (g) PFCM, $m_{ini} = 5$, $K = 1$, $\alpha = 1$, $\beta = 5$, $q = 1.5$, $n = 1.5$, (h) SPCM-L_1, $\lambda = 15$, $q = 2$ (i) APCM, $m_{ini} = 5$, $\alpha = 0.3$, (j) SPCM, $m_{ini} = 5$, and (k) SAPCM, $m_{ini} = 10$ and $\alpha = 0.15$.

Table 1 shows the results of all algorithms for Experiment. Fig. 2b and Fig. 2c show the clustering obtained using the k-means and FCM algorithms, respec-

Table 1: Performance of clustering algorithms for the data set of Experiment.

	m_{ini}	m_{final}	SR_{c_1}	SR_{c_2}	SR_{c_3}	MD	Iter	Time
k-means	3	3	51	0	100	3.4066	2	0.265
k-means	5	5	51	94	51.48	0.5369	20	0.202
FCM	3	3	51	0	100	3.3432	110	0.140
FCM	5	5	50.50	93	51.62	0.5537	86	0.218
PCM	5	2	100	0	100	0.9242	15	0.514
PCM	10	2	100	0	100	0.9254	18	1.185
UPC ($q = 1.5$)	5	4	50	95	100	0.4589	65	0.390
UPC ($q = 1.2$)	10	4	50	95	100	0.4480	89	0.910
UPFC ($a = 5$, $b = 1$, $q = 2$, $n = 1.5$)	5	4	50.50	96	100	0.4170	41	0.390
UPFC ($a = 5$, $b = 1$, $q = 2.2$, $n = 3$)	10	3	100	94	100	0.3601	190	2.940
PFCM ($K = 1$, $a = 1$, $b = 5$, $q = 1.5$, $n = 1.5$)	5	4	51.50	100	100	0.4573	38	0.380
PFCM ($K = 1$, $a = 2$, $b = 1$, $q = 2$, $n = 1.2$)	10	5	44	97	100	0.4011	60	0.880
SPCM-L_1 ($\lambda = 15$, $q = 2$)	-	2	76	0	100	1.1831	6	0.031
APCM ($\alpha = 0.3$)	5	4	53	100	100	0.4469	73	0.390
APCM ($\alpha = 0.3$)	10	4	52.50	100	100	0.4748	90	0.889
SPCM ($K = 0.9$)	5	2	100	0	100	0.9256	15	3.276
SPCM ($K = 0.9$)	10	2	100	0	100	0.9263	19	7.769
SAPCM ($\alpha = 0.18$)	5	3	**100**	**100**	**100**	**0.3222**	91	13.40
SAPCM ($\alpha = 0.15$)	10	3	**100**	**100**	**100**	**0.3020**	100	18.94

tively, both for $m_{ini} = 3$. Figs. 2d, 2e, 2f, 2g, 2h, 2i and 2j, depict the performance of PCM, UPC, UPFC, PFCM, SPCM-L_1, APCM and SPCM, respectively, with their parameters chosen (after fine-tuning) as stated in the caption. In addition, the circles, centered at each $\boldsymbol{\theta}_j$ and having radius $\sqrt{\gamma_j}$ (as they have been computed after the convergence of the algorithms), are also drawn.

As it can be deduced from Fig. 2 and Table. 1, even when the k-means and the FCM are initialized with the (unknown in practice) true number of clusters ($m = 3$), they fail to unravel the underlying clustering structure mainly due to the big difference in the variances and densities between clusters. The classical PCM also fails to detect the physical cluster 2, because it is located very close to the densest physical cluster. The UPC algorithm has been fine tuned so that the parameters γ_j's, which remain fixed during its execution and are the same for all clusters, get small enough values, in order to identify cluster C_2. However, it splits the high variance/low density cluster C_1 in two clusters. The same seems to hold for the PFCM algorithm, after fine tuning of its several parameters. The UPFC algorithm produces 3 clusters, at the cost of a computationally demanding fine tuning of the (several) parameters it involves. The APCM algorithm also splits the big variance cluster in two subclusters, failing to detect the underlying clustering structure. On the other hand, SPCM identifies two clusters with high accuracy with respect to the center of the actual clusters, but misses the third one. Finally, as it is deduced from Table 1, the SAPCM algorithm manages to identify all clusters, achieving the best SR and MD results and estimating very accurately the true centers of the clusters, since it exhibits the minimum MD among all algorithms.

References

1. J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-Means Clustering Algorithm", *Journal of the Royal Statistical Society*, vol. 28, pp. 100-108, 1979.
2. J. C. Bezdek, "A Convergence Theorem for the Fuzzy Isodata Clustering Algorithms", *IEEE Trans. on Pattern Analysis and Mach. Intel.*, vol. 2, pp. 1-8, 1980
3. J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", *Plenum*, 1981
4. R. Krishnapuram and J. M. Keller, "A Possibilistic Approach to Clustering", *IEEE Transactions on Fuzzy Systems*, vol. 1, pp. 98-110, 1993.
5. R. Krishnapuram and J. M. Keller, "The Possibilistic C-Means Algorithm: Insights and Recommendations", *IEEE Trans. on Fuzzy Systems*, vol. 4, pp. 385-393, 1996.
6. S. Theodoridis and K. Koutroumbas, "Pattern Recognition", *Academic Press*, 2009.
7. N. R. Pal and K. Pal and J. M. Keller and J. C. Bezdek, "A Possibilistic Fuzzy C-Means Clustering Algorithm", *IEEE Trans. on Fuzzy Systems*, pp. 517-530, 2005
8. M. S. Yang and K. L. Wu, "Unsupervised Possibilistic Clustering", *Pattern Recognition*, vol. 39, pp. 5-21, 2006.
9. S. D. Xenaki and K. D. Koutroumbas and A. A. Rontogiannis, "Adaptive Possibilistic Clustering", *IEEE International Symposium on Signal Processing and Information Technology*, pp. 422-427, Dec 2013.
10. S. D. Xenaki and K. D. Koutroumbas and A. A. Rontogiannis, "A Novel Adaptive Possibilistic Clustering Algorithm", *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 4, pp. 791-810, 2016.
11. S. D. Xenaki and K. D. Koutroumbas and A. A. Rontogiannis, "Sparsity-aware Possibilistic Clustering Algorithms", *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 6, pp. 1611-1626, 2016.
12. K. D. Koutroumbas and S. D. Xenaki and A. A. Rontogiannis, "On the Convergence of the Sparse Possibilistic C-Means Algorithm", *IEEE Transactions on Fuzzy Systems*, 2017, to appear. DOI: 10.1109/TFUZZ.2017.2659739
13. S. D. Xenaki and K. D. Koutroumbas and A. A. Rontogiannis, "Sparse Adaptive Possibilistic Clustering", *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 3072-3076, Florence 2014.
14. S. D. Xenaki and K. D. Koutroumbas and A. A. Rontogiannis, "Sequential Sparse Adaptive Possibilistic Clustering", *SETN 2014: Artificial Intelligence: Methods and Applications*, pp. 29-42, 2014.
15. S. D. Xenaki and K. D. Koutroumbas and A. A. Rontogiannis and O. A. Sykioti, "A Layered Sparse Adaptive Possibilistic Approach for Hyperspectral Image Clustering", *IEEE Intern.Geo. and Rem.Sens.Sympos(IGARSS)*, pp. 2890-2893, 2014.
16. S. D. Xenaki and K. D. Koutroumbas and A. A. Rontogiannis, "Hyperspectral Image Clustering Using a Novel Efficient Online Possibilistic Algorithm", *24th European Signal Processing Conference (EUSIPCO)*, pp. 2020-2024, 2016.
17. S. D. Xenaki and K. D. Koutroumbas and A. A. Rontogiannis and O. A. Sykioti, "A New Sparsity-Aware Feature Selection Method for Hyperspectral Image Clustering", *IEEE Intern. Geoscience and Rem. Sens. Symposium (IGARSS)*, pp. 445-448, 2015.
18. G. Corliss, "Which Root Does the Bisection Algorithm Find?", *Siam Review*, vol. 19, pp. 325-327, 1977.
19. X. Wu, B. Wu, J. Sun, H. Fu, "Unsupervised Possibilistic Fuzzy Clustering", *Journal of Information & Computational Science*, vol. 5, pp. 1075-1080, 2010.
20. Y. Hamasuna and Y. Endo, "On Sparse Possibilistic Clustering with Crispness Classification Function and Sequential Extraction", *Soft Computing and Intel. Syst. (SCIS) and 13th Intern. Symp. on Advanced Intel. Syst. (ISIS)*, pp. 1801-1806, 2012.