SPARSE ADAPTIVE POSSIBILISTIC CLUSTERING

Spyridoula D. Xenaki†*, Konstantinos D. Koutroumbas*†*, Athanasios A. Rontogiannis*†

†IAASARS, National Observatory of Athens, GR-152 36, Penteli, Greece
{ixenaki,koutroum,tronto}@noa.gr

ABSTRACT

In this paper a new sparse adaptive possibilistic clustering algorithm is presented. The algorithm exhibits high immunity to outliers and provides improved estimates of the cluster representatives by adjusting dynamically certain critical parameters. In addition, the proposed scheme manages - in principle - to estimate the actual number of clusters and by properly imposing sparsity, it becomes capable to deal well with closely located clusters of different densities. Extensive experimental results verify the previous statements.

Index Terms— possibilistic clustering, adaptivity, sparsity

1. INTRODUCTION

Clustering a set of objects into groups has been a well established data analysis method in unsupervised pattern recognition and it has been frequently used in a vast range of applications during the last decades (e.g. [1]). Most of the work in this field has been focused on compact and hyperellipsoidally shaped clusters. Among the various clustering methods that have attracted considerable attention in recent years are the cost function optimization ones, where a set of cluster representatives is iteratively adjusted in order to be moved to regions where the clusters live. Distinguished members of this family of methods are the crisp (e.g. [1]), the fuzzy [2] and the possibilistic clustering methods [3]. The celebrated k-means algorithm (e.g. [4]) is the most well-known representative of crisp algorithms, while the fuzzy c-means (FCM) algorithm is the most commonly used fuzzy clustering algorithm. However, they both require knowledge of the true number of clusters and are vulnerable to noisy data and outliers [1], [5]. On the other hand, the possibilistic c-means (PCM) algorithms are more robust to noise and outliers and even if the number of representatives is overestimated, they tend to move all representatives to "dense in data regions" in the data space. However, more than one representative may be moved to the same dense region [6], [7]. Some PCM variants that try to address the problems of the conventional PCM have been reported in [8],[9],[7]. Also in [10], [11] two variations of possibilistic clustering that impose sparsity constraints, adopting the l_1 norm, are proposed. In [11] the clusters are recovered in a sequential manner, in contrast to [10] (and all previous algorithms), where clusters are recovered simultaneously. Other such methods are given in [12], [13].

In the present paper, we deal with the PCM algorithm, which is extended and improved along two directions. Firstly, the number of clusters and the η parameters that it involves (see below) are not kept fixed (as in conventional PCM) but, rather, they are adapted as the algorithm evolves, leading to the so-called Adaptive PCM algorithm (APCM) ([14]). Note that by setting the (initial) number of clusters to a value greater - but not *much* greater - than the true number of the actual clusters, APCM (potentially) reduces this number to the number of *natural* clusters, by driving the cluster representatives towards dense regions. Secondly, a suitable sparsity constraint is imposed on the *degrees of compatibility* of each data vector with the clusters, giving rise to the Sparse APCM (SAPCM) algorithm. SAPCM exhibits increased immunity to data points that may be considered as noise or outliers by not allowing them (in principle) to contribute to the estimation of the cluster representatives. A consequence of this fact is that SAPCM estimates better the "centers" of the "dense regions". Moreover, SAPCM has, in principle, the ability to recover low-density clusters, located close to higher density clusters.

The rest of the paper is organized as follows. In Section 2, the new SAPCM clustering algorithm is presented. In Section 3, the performance of SAPCM is tested against various known clustering methods, using extensive computer simulations. Finally, concluding remarks are provided in Section 4.

2. THE SPARSE ADAPTIVE PCM (SAPCM)

Let $X = \{\mathbf{x}_i \in \Re^\ell, i = 1, ..., N\}$ be a set of N, l-dimensional data vectors. Let also $\Theta = \{\boldsymbol{\theta}_j \in \Re^\ell, j = 1, ..., m\}$ be a set of m vectors that will be used for the representation of the clusters formed in X. In what follows, $||\cdot||$ denotes the Euclidean norm. Let $U = [u_{ij}]$ be an $N \times m$ matrix whose (i, j) element stands for the so called *degree of compatibility* of $\mathbf{x}_i$ to the jth cluster, denoted by C_j, and represented by the vector $\boldsymbol{\theta}_j$. Let also ${\mathbf{u}_i}^T = [u_{i1}, ..., u_{im}]$ be the vector containing the elements of the ith row of U. In contrast to the original PCM, where u_{ij}'s should satisfy the conditions, (a) $u_{ij} \in [0, 1]$, (b) $\max_{j=1,...,m} u_{ij} > 0$ and (c) $0 < \sum_{i=1}^{N} u_{ij} < N$ ([3]), here, only the first one (a) is considered. Based on the minimization of a classical PCM cost function (see below and [15]), the u_{ij}'s and the representatives $\boldsymbol{\theta}_j$'s are (respectively) computed as

$$u_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \boldsymbol{\theta}_j\|^2}{\eta_j}\right) \quad (1) \qquad \boldsymbol{\theta}_j = \frac{\sum_{i=1}^{N} u_{ij}\mathbf{x}_i}{\sum_{i=1}^{N} u_{ij}} \quad (2)$$

Loosely speaking, the η_j parameter is a measure of how much the *influence* of C_j is spread around $\boldsymbol{\theta}_j$, i.e., a measure of its variance.

2.1. Initialization and adaptivity issues

Initialization: In the proposed clustering scheme, the initialization of $\boldsymbol{\theta}_j$'s is carried out using a fast approximate variation of the Max-Min algorithm proposed in [16] (see also [14]), in order to increase the probability of each $\boldsymbol{\theta}_j$ to be placed initially nearby a "dense in data" region. Denoting by X_{re} the set of the initial cluster represen-

This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: ARISTEIA - HSI-MARS - 1413.

tatives, we propose the following initialization of η_j's. First, the distance of each $\boldsymbol{\theta}_j \in X_{re}$ from its closest $\boldsymbol{\theta}_s \in X_{re} - \{\boldsymbol{\theta}_j\}$, denoted by $d_{\min}(\boldsymbol{\theta}_j)$, is determined and then η_j is set to $\eta_j = \frac{d_{\min}(\boldsymbol{\theta}_j)/2}{-\log\beta}$, where $\beta \in (0,1)$ is an appropriately chosen parameter (see *Initialization of η_j's* part in **Alg. 1**). As it has been verified experimentally, typical values for β that lead to good initializations are in the range $[0.1, 0.5]$. The experiments showed also that β depends on how densely the natural clusters are located; smaller values of β are more appropriate for sparsely located clusters, while larger values of β are more appropriate for more densely located clusters.

Adaptation: In the proposed clustering algorithm, this part refers to, (a) the adjustment of the number of clusters and (b) the adaptation of η_j's, which are two interrelated processes. Let $label$ be a N-dimensional vector, whose ith component contains the index of the cluster which is most *compatible* with $\mathbf{x}_i$, that is the cluster C_j for which $u_{ij} = \max_{r=1,\ldots,m} u_{ir}$. Let also n_j denote the number of the data points $\mathbf{x}_i$, that are most compatible with the jth cluster and $\boldsymbol{\mu}_j$ be the mean vector of these data points. The adjustment (reduction) of the number of clusters is achieved by examining if the index j of a cluster C_j appears in the vector $label$. If this is the case, C_j is preserved. Otherwise, C_j is eliminated (see *Possible cluster elimination* part in **Alg. 1**). Moreover, the η_j parameter of a cluster C_j is estimated as a measure of its variance, i.e., as the mean value of the distances of the most compatible to C_j data vectors from their mean vector $\boldsymbol{\mu}_j$ and *not* from the representative $\boldsymbol{\theta}_j$, as in previous works (e.g. [3], [7]) (see *Adaptation of η_j's* part in **Alg. 1**). It is also noted that, in the case where two or more clusters are equally compatible with a specific $\mathbf{x}_i$, then $\mathbf{x}_i$ will contribute to the determination of the η parameter of *only* one of them, which is chosen arbitrarily.

2.2. Sparsity issue

Sparsity is imposed on the vectors $\mathbf{u}_i$ via penalization of the PCM cost function with the l_p norm with $0 < p < 1$, i.e.,

$$J(\Theta, U) = J_{PCM}(\Theta, U) + \lambda \sum_{i=1}^{N}\sum_{j=1}^{m} u_{ij}^p \tag{3}$$

where the conventional PCM cost function (e.g. [3]) is given by

$$J_{PCM}(\Theta, U) = \sum_{i=1}^{N}[\sum_{j=1}^{m} u_{ij}\|\mathbf{x}_i - \boldsymbol{\theta}_j\|^2 + \sum_{j=1}^{m}\eta_j(u_{ij}\ln u_{ij} - u_{ij})] \tag{4}$$

and λ is a user-defined penalizing factor that controls sparsity. The SAPCM is derived via the minimization of $J(\Theta, U)$. Clearly the updating equation of $\boldsymbol{\theta}_j$'s will be the same as in eq. 2. It is only the updating of u_{ij}'s that will be modified, in the light of $J(\Theta, U)$.

Taking the derivative of J with respect to u_{ij} we get

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

where $d_{ij} = \|\mathbf{x}_i - \boldsymbol{\theta}_j\|^2$. Obviously, $\frac{\partial J(\Theta,U)}{\partial u_{ij}}$ becomes zero if and only if $f(u_{ij})$ becomes zero. Clearly, at a first glance, solving $f(u_{ij}) = 0$ is not a trivial task. However, it can be shown (the proof is omitted due to space limitations) that the roots of $f(u_{ij})$ - if they exist - are at most two and lie definitely in $[0, 1]$. Based on this, we first determine the minimum $\hat{u}_{ij}$ of $f(u_{ij})$, which is proved to be

$$\hat{u}_{ij} = \left(\frac{\lambda}{\eta_j} p(1-p)\right)^{\frac{1}{1-p}} \tag{6}$$

Algorithm 1 The SAPCM algorithm

Initialize: $t = 0$, $\boldsymbol{\theta}_j \equiv \boldsymbol{\theta}_j(0)$ *%from **Max-Min Algorithm** [16]*
%Initialization of η_j's
for $j = 1$ **to** m
  Determine: $d_{\min}(\boldsymbol{\theta}_j) = \min_{\boldsymbol{\theta}_s \in X_{re} - \{\boldsymbol{\theta}_j\}} \|\boldsymbol{\theta}_j - \boldsymbol{\theta}_s\|^2$
  Set: $\eta_j = \frac{d_{\min}(\boldsymbol{\theta}_j)/2}{-\log\beta}$
end for
Repeat:
  %Update U
  Update: U (as described in Section **2.2**)
  $t = t + 1$
  %Update Θ
  for $j = 1$ **to** m
    $\boldsymbol{\theta}_j(t) = \sum_{i=1}^{N} u_{ij}(t-1)\mathbf{x}_i \Big/ \sum_{i=1}^{N} u_{ij}(t-1)$
  end for
  %Possible cluster elimination
  for $i = 1$ **to** N
    Determine: $u_{ir} = \max_{j=1,\ldots,m} u_{ij}$
    If $u_{ir} \neq 0$
      Set: $label(i) = r$
    else
      Set: $label(i) = 0$
    end if
  end for
  for $j = 1$ **to** m
    If $j \notin label$
      Remove C_j
      $m = m - 1$
    end if
  end for
  %Adaptation of η_j's
  for $j = 1$ **to** m
    $\eta_j(t) = \frac{1}{n_j(t)} \sum_{\mathbf{x}_i : u_{ij}(t) = \max_{r=1,\ldots,m} u_{ir}(t)} \|\mathbf{x}_i - \boldsymbol{\mu}_j(t)\|$
  end for
Until: a termination criterion is met

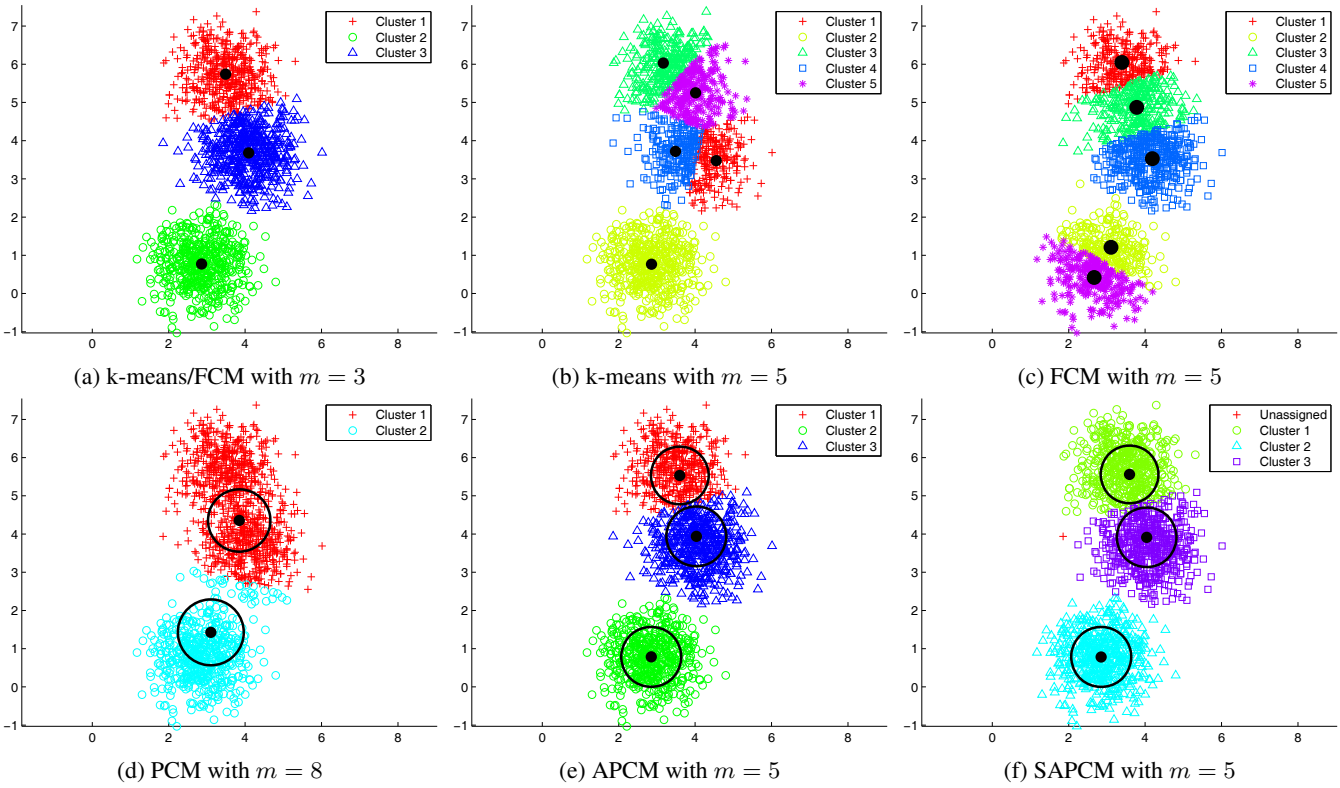
Then, we check whether $f(\hat{u}_{ij}) > 0$. If this is the case, then $f(u_{ij})$ has no roots in $[0, 1]$ and we set $u_{ij} = 0$, thus *imposing sparsity*. Otherwise, $f(u_{ij}) = 0$ has two solutions in $[0, 1]$[1]. Using a simple algebraic analysis, it turns out that the largest of these solutions is the one that minimizes $J(\Theta, U)$. To determine the largest of the solution(s) of $f(u_{ij}) = 0$, we apply the bisection method (e.g. [17]) in the range $[\hat{u}_{ij}, 1]$, which is known to converge very rapidly to the optimum u_{ij}. Other methods can also be used, e.g. [18].

Also, it turns out that a necessary condition for λ in order the equation $f(u_{ij}) = 0$ to have a solution (i.e., $f(\hat{u}_{ij}) < 0$) is $0 < \lambda < \min_{j=1,\ldots,m} \frac{\eta_j}{ep(1-p)}$, where e is the base of natural logarithms.

3. EXPERIMENTAL RESULTS

In this section, we test the proposed method in several experimental frameworks and illustrate the results (due to space limitations we

[1] Or one in the extreme case where $f(\hat{u}_{ij}) = 0$.



(a) k-means/FCM with $m = 3$ (b) k-means with $m = 5$ (c) FCM with $m = 5$

(d) PCM with $m = 8$ (e) APCM with $m = 5$ (f) SAPCM with $m = 5$

Fig. 1. Clustering results for **Experiment 1**. Note that in PCM the clustering result is extracted taking into account only the truly "different" clusters. Bolded dots represent the final clusters' representatives.

report here results from artificially generated experiments that highlight the special attributes of the proposed SAPCM). Moreover, we compare the results with those obtained from the k-means, the FCM, the PCM, the APCM and the algorithm proposed in [11]. The latter imposes sparsity on $\boldsymbol{u}_i$'s in a different way from that in SAPCM and uses the l_1 norm. In addition, in order to be fair, the representatives ($\boldsymbol{\theta}_j$) are initialized based on the Max-Min scheme and the parameters are fine tuned, in all algorithms. In all experiments, p is set equal to 0.5 for SAPCM algorithm.

Experiment 1: Let us consider a two-dimensional data set consisting of $N = 1500$ points, where three clusters C_1, C_2, C_3 are formed. Each cluster is modelled by a normal distribution. The means of the distributions are $\boldsymbol{c}_1 = [3.5,\ 5.7]^T$, $\boldsymbol{c}_2 = [2.8,\ 0.8]^T$ and $\boldsymbol{c}_3 = [4.1,\ 3.7]^T$, respectively, while their (common) covariance matrix is set to $0.4 \cdot I_2$, where I_2 is the 2×2 identity matrix. A number of 500 points are generated by each one of the distributions. Note that C_1 and C_3 clusters are close enough to each other, while they are far away from C_2. In all algorithms, after their convergence, each data point $\mathbf{x}_i$ is assigned to a cluster C_j, if $u_{ij} = \max_{r=1,\ldots,m} u_{ir}$. In particular in the SAPCM algorithm, for a data point $\mathbf{x}_i$ that has not been assigned to any cluster (i.e. $u_{ij} = 0, \forall j$), we estimate u_{ij}'s as $u_{ij} = \exp(-\frac{\|\mathbf{x}_i - \boldsymbol{\theta}_j(t)\|^2}{\eta_j})$ for each cluster and then we assign $\mathbf{x}_i$ to a cluster as before. In order to compare a clustering with the true data label information, we use the Rand Measure described e.g. in [1].

Table 1 shows the results of all the previously mentioned algorithms, where $m_{initial}$ and m_{final} denote the initial and the final number of clusters, respectively. Fig. 1 (a) shows the clustering result obtained using the k-means and the FCM algorithm with $m = 3$. Figs. 1 (b), (c) present the clustering results obtained using the k-means and the FCM algorithm with $m = 5$, respectively. Fig. 1 (d) depicts the performance of PCM for $m = 8$, while in addition, it shows the circled regions, centered at each representative $\boldsymbol{\theta}_j$ and having radius equal to η_j, in which cluster C_j has increased influence. Finally, Figs. 1 (e), (f) show the results of APCM (with $m = 5$ and $\beta = 0.1$) and SAPCM (with $m = 5$, $\beta = 0.1$ and $\lambda = 0.05$), respectively.

Table 1. The results of the data set: **Experiment 1**

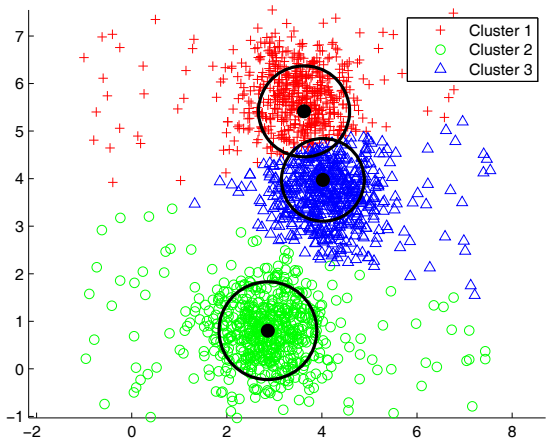
	$m_{initial}$	m_{final}	Rand Measure
k-means	3	3	**94.71%**
k-means	5	5	85.68%
FCM	3	3	**94.63%**
FCM	5	5	83.76%
PCM	3 to 5	1	33.29%
PCM	8	2	76.35%
Alg. of [11]	-	4	77.24%
APCM [14]	5 to 10	3	**94.55%**
SAPCM	5 to 10	3	**94.59%**

As it can be deduced from Fig. 1 and Table 1, when the k-means and the FCM are initialized by the (unknown in most cases) true number of clusters ($m = 3$), their performance is very satisfactory. However, any deviation from this value causes a significant degradation to the obtained clustering quality. On the other hand, the classi-

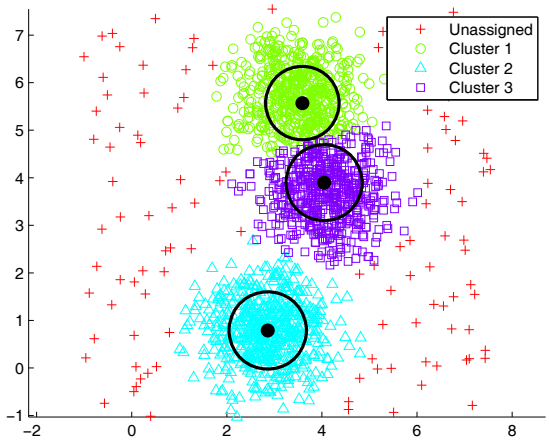
cal PCM fails to unravel the underying clustering structure, mainly due to the fact that two clusters are close enough to each other and the algorithm does not have the ability to adapt η_j's in order to distinguish them. Finally, both the APCM and the proposed SAPCM constantly produce very accurate results for various initial values of m, while the algorithm of [11] (with $\lambda = 10$ and $q = 2$) yields inferior performance compared to the previous ones.

We focus next exclusively on APCM and SAPCM.

Experiment 2: Let us consider the same two-dimensional data set as in **Experiment 1**, in which 200 data points are added randomly as noise in the region where data live (Fig. 2).



(a) APCM with $m = 5$ and $\beta = 0.1$



(b) SAPCM with $m = 5$, $\lambda = 0.1$ and $\beta = 0.1$

Fig. 2. Clustering results for **Experiment 2**.

Table 2. Mean Euclidean distance between $\boldsymbol{\theta}_j$'s and $\boldsymbol{c}_j$'s

	$m_{initial}$	m_{final}	Mean distance
APCM [14]	5 (10)	3	0.2262 (0.2263)
SAPCM	5 (10)	3	**0.1467 (0.1469)**

As it is shown in Fig. 2, when outliers or noisy data points are ignored by the clustering process (the SAPCM case), dense regions are detected more accurately and the representatives are placed very close to their exact centers. Table 2 shows the mean of the Euclidean distances between each representative ($\boldsymbol{\theta}_j$) and its closest mean ($\boldsymbol{c}_j$) for each one of the two algorithms. It can be seen that SAPCM estimates more accurately the centers of the clusters, as the outliers do not participate in the estimation of the representatives. In addition, in Fig. 2 it is shown how the outliers affect the estimation of η_j's. Obviously, in the APCM case, where each data point contributes to the estimation of one η_j parameter, η_j's increase, due to the long distance between the outliers and the means of the clusters. Thus the circled regions in which clusters C_j have increased influence, grow significantly and may affect neighboring clusters. This may lead APCM to fail in distinguishing between two closely located clusters (see next example). On the other hand, SAPCM imposes sparsity to a sufficient degree so that the remotely located from the mean of the cluster points are not taken into account, thus leading to smaller values for η_j's. This is an explanation of why the circled regions that correspond to clusters C_1 and C_3 do not overlap in Fig. 2(b).

Experiment 3: Let us consider again the set-up of **Experiment 1**, where now 300 points are generated by each one of the first two distributions and 500 points are generated by the third one. Note that clusters C_1 and C_3 have a great difference in their density (since both share the same covariance matrix but C_1 has significantly less points than C_3) and since they are closely located to each other, a clustering algorithm could consider them as a single cluster.

In this data set, the APCM algorithm fails to unravel the underlying clustering structure and unites clusters C_1 and C_3 thus leading to a two-cluster clustering result for several values of β. In order to get some further insight on how APCM attains such an outcome, let us consider the updating equation of $\boldsymbol{\theta}_j$'s. For the low-density cluster C_1, the data points of C_3 will give (probably) small but "more" contributions to the estimation of $\boldsymbol{\theta}_1$ than the rest most compatible with C_1 data points, due to the plurality of the former in the bordered area between the two clusters. Thus, the representative $\boldsymbol{\theta}_1$ will gradually be driven towards the denser region gaining more and more data points from the bordered area. This results in greater values for η_1 and finally leads to the unification of the clusters after some iterations. On the other hand, for a large enough value of λ ($\lambda = 0.3$), SAPCM heavily imposes sparsity so that the remotely located from the mean of the C_1 cluster points are not taken into account to the estimation of the parameters of cluster C_1, thus leading to smaller values for η_1. As a consequence, the unification of C_1 with its neighboring (denser) C_3 cluster is prevented (a relevant figure is omitted due to space limitation).

4. CONCLUSIONS

In this paper a novel sparse adaptive possibilistic clustering algorithm (SAPCM) has been proposed. The algorithm (a) encompasses a proper initialization and a new updating mechanism for the η parameters and (b) is immune to overestimates of the actual number of existing clusters. Moreover, SAPCM imposes a sparsity constraint on the degrees of compatibility of each data vector with the clusters. These modifications and additions to the original PCM make the new algorithm very flexible in unraveling the underying clustering structure via: (a) the improvement on the estimation of the cluster representatives, through the adaptation of η_j's and the exclusion of the outliers from contributing to this estimation and (b) its capability of detecting (in principle) the number of natural clusters. In extensive experiments, it is shown that SAPCM has a steadily superior performance, compared to k-means, FCM and the conventional PCM, irrespective of the initial estimate of the number of clusters. Also, in principle, it has the ability to recover closely located clusters of different densities.

5. REFERENCES

[1] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, fourth edition*, Academic Press, 2009.

[2] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum, 1981.

[3] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Transactions on Fuzzy Systems*, vol. I, pp. 98–110, 1993.

[4] 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.

[5] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 13, pp. 517–530, 2005.

[6] M. Barni, V. Cappellini, and A. Mecocci, "A possibilistic approach to clustering," *IEEE Transactions on Fuzzy Systems*, vol. IV, pp. 393–396, 1996.

[7] J.-S. Zhang and Y.-W. Leung, "Improved possibilistic c-means clustering algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 12, pp. 209–217, 2004.

[8] M.-S Yang and K.-L. Wu, "Unsupervised possibilistic clustering," *Pattern Recognition*, vol. 39, pp. 5–21, 2006.

[9] K. Treerattanapitak and C. Jaruskulchai, "Possibilistic exponential fuzzy clustering," *Journal of computer science*, vol. 28, pp. 311–321, 2013.

[10] R. Inokuchi and S. Miyamoto, "Sparse possibilistic clustering with l1 regularization," in *IEEE International Conference on Granular Computing*, 2007.

[11] Y. Hamasuna and Y. Endo, "On sparse possibilistic clustering with crispness classification function and sequential extraction," in *Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS)*, 2012.

[12] P. A. Forero, V. Kekatos, and G. B. Giannakis, "Robust clustering using outlier-sparsity regularization," *IEEE Transactions on Signal Processing*, vol. 60, pp. 4163–4177, 2012.

[13] M.-S. Yang, K.-L. Wu, J.-N. Hsieh, and J. Yu, "Alpha-cut implemented fuzzy clustering algorithms and switching regressions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, pp. 588–603, 2008.

[14] S. D. Xenaki, K. D. Koutroumbas, and A. A. Rontogiannis, "Adaptive possibilistic clustering," *IEEE International Symposium on Signal Processing and Information Technology*, 2013.

[15] R. Krishnapuram and J. M. Keller, "The possibilistic c-means algorithm: insights and recommendations," *IEEE Transactions on Fuzzy Systems*, vol. IV, pp. 385–393, 1996.

[16] B. Mirkin, *Clustering for Data Mining: A Data Recovery Approach*, Chapman Hall, 2005.

[17] G. Corliss, "Which root does the bisection algorithm find?," *Siam Review - SIAM REV*, vol. 19, pp. 325–327, 1977.

[18] A. S. Householder, *The Numerical Treatment of a Single Nonlinear Equation*, McGraw-Hill, 1970.