**Generalized entropy based possibilistic fuzzy C-means for clustering noisy data and its convergence proof**

**S. Askaria, N. Montazerina,[[1]](#footnote-2), M.H. Fazel Zarandib**

a Mechanical Engineering Department, Amirkabir University of Technology (Tehran Polytechnic), 424 Hafez Avenue, Tehran 1591634311, Iran.

b Industrial Engineering Department, Amirkabir University of Technology (Tehran Polytechnic), 424 Hafez Avenue, Tehran 1591634311, Iran.

**Abstract-**A Generalized Entropy based Possibilistic Fuzzy C-Means algorithm (GEPFCM) is proposed in this paper for clustering noisy data. The main objective of GEPFCM is to determine accurate cluster centers of noisy data by generalizing Entropy C-Means (ECM) combined with Possibilistic Fuzzy C-Means (PFCM). GEPFCM utilizes functions of distance instead of the distance itself in the fuzzy, possibilistic, and entropy terms of the clustering objective function to decrease noise contributions on the cluster centers. This study shows that GEPFCM is more accurate than PFCM algorithm. A measure based on the distance between the actual and computed cluster centers demonstrates that error of GEPFCM is about 80% less than that of PFCM.

***Keywords:*** Fuzzy C-Means, Possibilistic C-Means, Entropy C-Means, Noise, Noisy data, Fuzzy clustering, Convergence proof

**1. Introduction**

Fuzzy clustering groups datasets into fuzzy clusters. Each data point has a membership grade in each cluster depending on its similarity to that cluster. Cluster centers are pivotal in fuzzy clustering from which membership grades are calculated. Fuzzy clustering is used in different applications such as image segmentation [1-5], fuzzy time series [6,7], time series [8], color segmentation [9], fault diagnosis [10], gene selection [11], etc.

Fuzzy C-Means (FCM) as the basic algorithm of fuzzy clustering aims at maximizing compactness of the clusters [12]. Proof of convergence and optimality of this algorithm are presented in [13-17]. FCM is developed in various forms for different applications and purposes. Suppressed FCM is proposed to increase the difference between high and low membership grades [18]. Suppression is implemented by competition among the clusters such that lower membership grades are reduced and larger membership grades are increased for each data vector. The suppressed FCM gives more accurate partitions of the data with less iteration compared to the FCM itself [18]. Also a modified form of FCM is proposed that converges with less iteration [19]. Two online fuzzy clustering algorithms are presented for very large datasets based on medoids and data streams where data are produced continuously [20]. Data vectors with high membership grades in each cluster are considered as medoid candidates of the cluster. Using the medoids allows clustering non-numerical data. It is shown that medoid based algorithms outperform their conventional competitors. Relational version of medoid based FCM is presented in [21]. There is also a global form of FCM based on a simulated annealing method which gives more compact clusters compared to FCM [22]. FCM does not work well for high-dimensional data [23]. A sparse FCM is presented for theses data and it is shown that its performance is satisfactory. A fuzzy clustering algorithm is proposed by combining different algorithms such as FCM, ECM, etc. Performance of this hybrid algorithm is theoretically studied and a method is presented for choosing different parameters of the algorithm [24]. A kernel based FCM combined with genetic algorithm is proposed to improve FCM performance [25]. Genetic algorithm optimizes initial cluster centers and then kernel based FCM finalizes the clustering procedure. This algorithm outperforms the traditional FCM. A multivariate FCM assigns different weights to the variables which demonstrate importance of each variable for each cluster to improve clustering quality [26]. Suitable value of degree of fuzziness  is a controversial problem for which many researchers choose a value between 1.5 to 2.5 [27]. Large values of  improves FCM performance for clustering noisy data [27]. Euclidean norm is usually used in FCM. It is shown that FCM based on a metric other than the Euclidean norm is more robust for both hard and soft clustering [28]. A size insensitive algorithm is presented to decrease FCM sensitivity to the cluster size using a conditional value  associated with the  data vector [29]. Various forms of FCM are developed to cluster very large data (data that cannot be loaded into the working memory of the computer) [30] and an extension of fuzzy clustering algorithms to shell clustering is found in [31,32]. One of the main drawbacks with FCM and its developed versions is weak performance on noisy data. FPCM [33] is proposed as a combination of Possibilistic C-Means algorithm (PCM) [34-36] and FCM to handle noisy data. However, FPCM suffers from two major problems of PCM, namely coincident clusters and sensitivity to initialization. To avoid these drawbacks with PCM and FPCM, Possibilistic Fuzzy c-Means algorithm (PFCM) is presented as another combination of these algorithms which is supposed to be more suitable for clustering noisy data.

The following index is minimized in PFCM [37]:

|  |  |
| --- | --- |
| (1) |  |

Where  is number of data vectors,  is number of clusters,  is center of the  cluster,  is data dimension,  is  data vector ( column of data matrix ),  is membership grade of the  data vector in  cluster,  is distance,  is degree of fuzziness,  and  are relative importance of fuzzy membership grades  and typicalities  (elements of typicality matrix ),  is a positive nonzero real number as , and  is an appropriate norm matrix. Higher  and  increases fuzziness and overlap of the clusters. In fact,  is a degree of “belonging” assigned to data vectors which reduces with distance. Since noise points and outliers are far from the cluster centers, it is supposed that  decreases their effects on the cluster centers and because of this, PFCM is more appropriate than FCM for noisy data.

The following matrices including partition matrix , typicality matrix , and cluster centers matrix  minimize (1) [37]:

|  |  |
| --- | --- |
| (2) |  |

Where  is computed using FCM algorithm and  [37].  is in fact weighted distance of all data vectors from the center of  cluster. FCM and PCM are derived from PFCM by taking  and , respectively. There is also another clustering method based on entropy which is named Entropy C-Means (ECM). The following index is minimized for ECM which maximizes both compactness and entropy [38,39]:

|  |  |
| --- | --- |
| (3) |  |

Following values of  and  minimize (3) [38,39]:

|  |  |
| --- | --- |
| (4) |  |

Where  is the Lagrangian multiplier. This constrained form of ECM and its modified versions show weak performance on noisy data similar to FCM due to relatively high membership grades of noise points resulted from the constraint . Another form of ECM is presented in this paper by relaxing this constraint. It is shown that relaxed ECM combined with generalized forms of FCM and PCM is more accurate for noisy data.

Main objective of this paper is to generalize FCM, PCM, and PFCM algorithms using functions of distance  for fuzzy, possibilistic, and entropy terms in (1) and (3) to cluster noisy data. The term  is usually used in objective functions of fuzzy clustering algorithms as in (1) and (3) which results linear updating equation for prototypes as (2) and (4). Using a function of  instead of , leads to a nonlinear updating equation for the prototypes which is our approach in this work. Therefore, we use  instead of  where  is an appropriate function associated with  cluster to damp noise contributions and  and .

Rest of the paper is organized as follows. GEPFCM is formulated in Section 2 where its mathematical framework is discussed and its convergence proof is presented. Some noisy datasets are clustered by GEPFCM and PFCM in Section 3 to evaluate the algorithms performance. Clustering quality and runtime of algorithms are studied in Section 4. Application of GEPFCM algorithm to real datasets is studied in Section 5 and the paper is summarized in Section 6.

**2. Generalized Entropy based Possibilistic Fuzzy C-Means**

**2.1. Mathematical formulation**

Objective function of the GEPFCM algorithm is devised as:

|  |  |
| --- | --- |
| (5) |  |

Where , , and  are functions associated with fuzzy, possibilistic, and entropy terms, and  is a conditional value pertaining to the  data vector. Using , membership grades will never be undetermined which is a well-known problem with FCM and PFCM. We formulate the algorithm for both constrained and relaxed forms of the entropy terms, however, we just use the relaxed form of GEPFCM because it shows better performance on noisy data. Using Lagrange Multipliers, , , , and  matrices are calculated by minimizing the following function.



Cluster centers of the data are calculated as:



|  |  |
| --- | --- |
| (6) |  |

When convergence criterion  is met ( indicates iteration and  is a predefined number), the algorithm stops. For the membership grade :



Combining the constraint  and  results:

|  |  |
| --- | --- |
| (7) |  |

Typicality  is computed as:



|  |  |
| --- | --- |
| (8) |  |

Initial values of , , and  are calculated for GEPFCM by applying PFCM to the data. For relaxed form of GEPFCM  is calculated as:



|  |  |
| --- | --- |
| (9) |  |

If the constraint  is considered, then  is calculated as:





|  |  |
| --- | --- |
| (10) |  |

Constrained form of the algorithm has weak performance on noisy data because of uniform distribution of  of noise points in the clusters due to the constraint. So, we study relaxed form of GEPFCM which is more suitable for noisy data.

Various clustering algorithms result from GEPFCM after changing constants and functions of (5) as shown in Table 1. The letter G at the beginning of an algorithm name indicates “Generalized” and the letters R and C at the end of an algorithm name indicate “Relaxed” and “Constrained”, respectively. Note that if , then (10) results  and .

The letter R is omitted in the following discussions and GEPFCM is used instead of GEPFCMR.

Table 1. Various form of GEPFCM algorithm.

|  |  |  |  |
| --- | --- | --- | --- |
| Conditions on constants | Conditions on functions | Constraints | Algorithm |
| - | - |  | GEPFCMR |
| - | - |  | GEPFCMC |
|  | - |  | GPFCM |
|  |  |  | FCM [12] |
|  | - |  | GFCM |
|  |  | - | PCM [34] |
|  | - | - | GPCM |
|  |  |  | PFCM [37] |
|  |  | - | ECMR |
|  |  |  | ECMC [38] |
|  |  |  | EFCMR |
|  |  |  | EFCMC |
|  | - |  | GEFCMR |
|  | - |  | GEFCMC |
|  |  | - | EPCMR |
|  |  |  | EPCMC |
|  | - | - | GEPCMR |
|  | - |  | GEPCMC |
| - |  |  | EPFCMR |
| - |  |  | EPFCMC |

Noise contributions on the prototypes can be reduced using proper functions , , and . We enclose  cluster center with a hyper-sphere. Radius of this hyper-sphere is computed as:

|  |  |
| --- | --- |
| (11) |  |

Where  is the covariance norm matrix defined as:

|  |  |
| --- | --- |
| (12) |  |

The general form  is considered for the functions , , and  where  is chosen such that the following index is minimized to acquire maximum compactness.

|  |  |
| --- | --- |
| (13) |  |

 in (13) is calculated from (9) simply by assuming . One can compute  as a function of  and then choose optimal value of  that minimizes . For each , GEPFCM is applied to the data and then  is computed from (13). Cluster centers computed from the updating equations (6)-(10) are employed to calculate  in (13). Values of GEPFCM functions , , and  depend on . Therefore,  implicitly depends on .

It is worth noting that instead of  given in (13), other indices (such as cluster validity indices) expressing compactness, separation or both of them, may be suitable.

Flowchart of GEPFCM algorithm is shown in Fig. 1 where  indicates iteration. As shown in the flowchart, first FCM is applied to the data  and results of FCM algorithm initialize PFCM algorithm. PFCM algorithm is then applied to the data and its results initialize GEPFCM algorithm. Finally, GEPFCM algorithm is applied to the data to compute final cluster center matrix . FCM starts with a random partition matrix . Using  in (2) turns PFCM given in (1) and (2) into the FCM. It is well-known that FCM is insensitive to initialization. Since GEPFCM starts with FCM, it is insensitive to initialization as well.

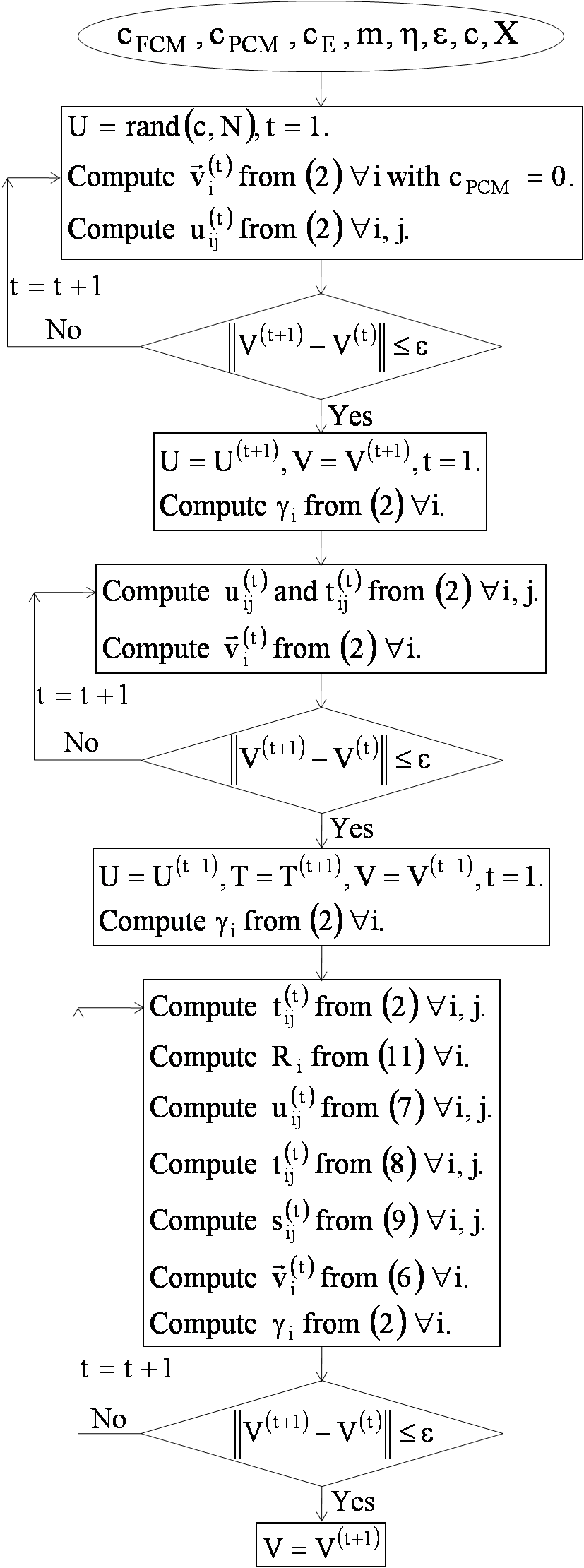


Fig. 1. Flowchart of GEPFCM algorithm.

The functions  and  are used in the present study. In all computations, it is assumed that: which are usually used in FCM and PFCM algorithms. Moreover,  (for  cluster) is used to make  dimensionless from which  is computed. These are general forms of the functions and constants, however, for simplified forms of GEPFCM they will be conformed to those given in Table 1.

Assume  where  is a constant then  is a hyper-ellipsoid which surrounds the  cluster. GEPFCM functions  and  damp noise contributions inside the hyper-ellipsoid which enables the algorithm to compute more accurate cluster centers compared to PFCM. The parameter  which depends on the data, controls size of the hyper-ellipsoid.

**2.2. Convergence proof of GEPFCM and parameters settings**

Convergence proof of fuzzy subspace clustering is given in [40] and that of FCM algorithm is given in [14]. Assuming , , and zeroing other constants of (5) results FCM objective function:



It is shown that  is equivalent to [14]:

|  |  |
| --- | --- |
| (14) |  |

 is a function of two variables  and  whereas  is only a function of . Therefore, studying convergence of  is easier.

We use a similar procedure to study convergence of GEPFCM algorithm. Objective function of GEPFCM is written as:



 is a function of variables , , , and which are present in its gradient as well. This makes it difficult to judge whether the objective function converges or not. Therefore, we need to find an equivalent objective function for GEPFCM as  which is only a function of cluster centers . So,  should satisfy the following equation:



|  |  |
| --- | --- |
| (15) |  |

The functions , , and  are calculated as follows to construct .



Consider 



|  |  |
| --- | --- |
| (16) |  |



Consider 



Since ,  is a real number.



|  |  |
| --- | --- |
| (17) |  |



|  |  |
| --- | --- |
| (18) |  |

Therefore, the following equivalent objective function results for GEPFCM algorithm:

|  |  |
| --- | --- |
| (19) |  |

Assuming ,  is written as:

|  |  |
| --- | --- |
| (20) |  |

Cluster centers of  are computed as:



|  |  |
| --- | --- |
| (21) |  |

Where  indicates iteration.  can be written as:



|  |  |
| --- | --- |
| (22) |  |

Ref. [14] shows that:



|  |  |
| --- | --- |
| (23) |  |

Substituting (22) in (23) yields:



|  |  |
| --- | --- |
| (24) |  |

Since  , if  then:

|  |  |
| --- | --- |
| (25) |  |

Therefore,  reduces with iteration  and GEPFCM converges.

We use the functions  and  and therefore .

It is worth noting that using , , and  with negative derivatives or choosing large , , and  may impair convergence of GEPFCM algorithm since they may result small or negative  in some iterations. Moreover, to reduce noise impacts on the cluster centers, according to (6), the derivatives , , and  should monotonically reduce with distance. Therefore, the functions , , and  should be chosen such that .

**3. GEPFCM application to sample noisy datasets**

Six two-dimensional noisy datasets are clustered in this section using GEPFCM and PFCM algorithms to evaluate performance of the proposed algorithm. Details of the datasets are given in Table 2. There are two clusters in DATA 1 and three clusters in DATA 2, DATA 3 and DATA 4. Actual data points in DATA 2 to DATA 4 are the same but number of noise points increases from DATA 2 to DATA 4 to study impact of number of noise points on the algorithms performance. DATA 5 and DATA 6 with six and nine clusters are used to examine performance of the algorithms at higher numbers of clusters.

Table 2. Details of sample datasets.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | DATA 1 | DATA 2 | DATA 3 | DATA 4 | DATA 5 | DATA 6 |
| No. of clusters | 2 | 3 | 3 | 3 | 6 | 9 |
| No. of data points | 204 | 306 | 306 | 306 | 612 | 918 |
| No. of noise points | 2000 | 1000 | 2000 | 3000 | 3060 | 5000 |

We compare performance of the relaxed form of GEPFCM with the original algorithm PFCM. Rows of a dataset may have different physical units which makes Euclidean distance meaningless. We use covariance norm matrix which gives dimensionless distance. Covariance norm matrix is defined as follows which creates ellipsoidal clusters with Mahalanobis distance:

|  |  |
| --- | --- |
| (26) |  |

A measure is needed to compare performance of the algorithms. We define  as a vector that its  element is the distance between  actual and computed cluster centers.  is calculated as:

|  |  |
| --- | --- |
| (27) |  |

Where  and  are computed and actual cluster centers, respectively. Length of  divided by the number of clusters is defined as the measure of performance:

|  |  |
| --- | --- |
| (28) |  |

DATA 1 is clustered and shown in Fig. 2 (a). Fig. 2 (b) shows variation of  with . In this figure and all other figures of this paper, + and × indicate cluster centers computed by GEPFCM and PFCM algorithms, respectively. Results show that GEPFCM is more accurate compared to PFCM. Actual cluster centers and those calculated by PFCM and GEPFCM are:



Performance of the algorithms is evaluated as follows using (27) and (28).



Error of GEPFCM is 97.8% () lower than that of PFCM.



Fig. 2. (a) Results of clustering DATA 1 by PFCM and GEPFCM and (b) variation of  with .

Results of clustering DATA 2, DATA 3 and, DATA 4 are shown in Figs. 3, 4 and 5, respectively. It is perceived that GEPFCM outperforms PFCM. The actual data points in Fig. 3 to Fig. 5 are the same but numbers of random noise points are different. For a larger number of noise points, cluster centers computed by PFCM are more and more pulled towards the centroid of the entire dataset which is expected from (2) and accuracy of this algorithm decreases. These figures show that prototypes computed by PFCM are entirely inaccurate and any further calculation or inference based on these prototypes is irrelevant to the data and cannot reflect nature, behavior, and structure of the data. On the other hand, GEPFCM determines accurate prototypes by damping noise contributions using the functions . Actual prototypes of DATA 2 and those computed by PFCM and GEPFCM are:



Performance of the algorithms are:



Error of GEPFCM is 98.9% smaller than that of PFCM.



Fig. 3. (a) Results of clustering DATA 2 by PFCM and GEPFCM and (b) variation of  with .

Actual and computed prototypes of DATA 3 are:



Performance of the algorithms are:



Error of GEPFCM is 97.8% less than that of PFCM.



Fig. 4. (a) Results of clustering DATA 3 by PFCM and GEPFCM and (b) variation of  with .

Actual and calculated cluster centers of DATA 4 are:



Performance of the algorithms are:



Error of GEPFCM is 96.3% smaller than that of PFCM.

Clustering errors of DATA 2 to DATA 4 show that accuracy of both algorithms decreases by increasing number of noise points.



Fig. 5. (a) Results of clustering DATA 4 by PFCM and GEPFCM and (b) variation of  with .

DATA 5 and DATA 6 are clustered and shown in Figs. 6 and 7, respectively. It is observed that GEPFCM is superior to PFCM again. Prototypes of DATA 5 and performance of PFCM and GEPFCM are:









Error of GEPFCM is 92.8% smaller than that of PFCM.



Fig. 6. (a) Results of clustering DATA 5 by PFCM and GEPFCM and (b) variation of  with .

Prototypes of DATA 6 are:







Performance of PFCM and GEPFCM are:



Error of GEPFCM is 86.5% lower than that of PFCM.

Figures 6 and 7 show that when the clusters are closer to the center of the entire data, performance of PFCM improves which is expected from (2). For the cluster close to the center of the dataset, noise points of the opposite sides of the cluster center cancel their contributions which reduces noise impacts on the PFCM performance. It does not happen when the cluster is far apart from the dataset center and close to the boundaries of the data domain which decreases PFCM accuracy. GEPFCM initializes by the prototypes computed using PFCM which are relatively good. GEPFCM employs a hyper-sphere surrounding each prototype as a shield to decrease noise impacts on the cluster centers. For the data points inside the hyper-sphere of the  cluster,  and consequently final location of the prototype is highly influenced by these data points. The data points with  are outside the hyper-sphere and their impacts on the cluster center considerably reduces by GEPFCM functions , , and .



Fig. 7. (a) Results of clustering DATA 6 by PFCM and GEPFCM and (b) variation of  with .

The three-dimensional noisy data shown in Fig. 8 consist four clusters. Number of actual data points and noise points are 4400 and 8000, respectively and optimal value of  is 25 for these data.



Fig. 8. Three-dimensional noisy data.

Actual cluster centers of these data and those calculated by PFCM and GEPFCM algorithms are as follows:







GEPFCM is considerably more accurate than PFCM in clustering noisy three-dimensional data.

**4. Clustering quality and runtime of algorithms**

Clustering quality and runtime of the algorithms are compared in this section.

**4.1. Clustering quality**

Clustering quality measures compactness and separation of the clusters. Two well-known indices including Xie-Beni Index (XBI) and Davies-Bouldin Index (DBI) [41] are used in this work to compare quality of clustering performed by PFCM and GEPFCM algorithms. Both membership grades and typicalities are considered in these indices [42]. Lower value of these indices indicates higher compactness and separation and better clustering quality. XBI is defined as:

|  |  |
| --- | --- |
| (29) |  |

DBI is defined as:

|  |  |
| --- | --- |
| (30) |  |

XBI and DBI are computed for the datasets of Table 2 and the three-dimensional data shown in Fig. 8 using PFCM and GEPFCM algorithms and are given in Table 3. GEPFCM outperforms PFCM in terms of both indices.

Table 3. XBI and DBI for different datasets calculated by PFCM and GEPFCM algorithms.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | DATA1 | DATA2 | DATA3 | DATA4 | DATA5 | DATA6 | 3D DATA |
| PFCM | XBI | 0.9912 | 0.3615 | 0.4671 | 0.4903 | 0.4254 | 0.3870 | 0.3517 |
| DBI | 2.5624 | 0.7093 | 0.5827 | 0.6935 | 2.4788 | 1.0028 | 0.6018 |
| GEPFCM | XBI | 0.5629 | 0.3572 | 0.4413 | 0.4692 | 0.3733 | 0.3742 | 0.2951 |
| DBI | 0.8201 | 0.5685 | 0.4189 | 0.5823 | 1.9794 | 0.8787 | 0.5152 |

**4.2. Runtime of algorithms**

Runtime of PFCM and GEPFCM algorithms for clustering the datasets shown in Table 2 and the three-dimensional data shown in Fig. 8 are given in Table 4. Runtime of GEPFCM is higher than that of PFCM because of the nonlinear updating equation of cluster centers. However, GEPFCM yields more accurate cluster centers whereas those computed by PFCM cannot be used because they are very different from the actual cluster centers.

Table 4. Runtime (seconds) of PFCM and GEPFCM algorithms.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | DATA1 | DATA2 | DATA3 | DATA4 | DATA5 | DATA6 | 3D DATA |
| PFCM | 8.4485 | 3.0906 | 8.4682 | 13.0284 | 83.7185 | 773.4225 | 57.0032 |
| GEPFCM | 31.6760 | 14.7551 | 29.6097 | 63.0511 | 165.2847 | 1094.4832 | 141.7745 |

**5. GEPFCM application to real datasets**

GEPFCM algorithm is applied to real datasets in this section for noise rejection and handling datasets with lots of outliers.

**5.1. Noise rejection**

Gas companies need to fit a line to Gas Consumption (GC) data as a function of Heating Degree Days (HDD) for consumption management and savings. Standard does not allow a curve fitting with . Consumption data analysis shows that noise points or outliers cause poor curve fitting and low . Since there are a large number of consumers, it is not possible to detect noise points visually. It is shown that GEPFCM is capable of detecting and rejecting noise points. Consumption data of each consumer can be grouped into two fuzzy clusters one for cold seasons (high consumption) and the other for hot seasons (low consumption). Data of each consumer is denoised individually. The following maximum typicality is computed for the  data point after applying GEPFCM on the data:

|  |  |
| --- | --- |
| (31) |  |

Cluster centers s in (31) are computed using GEPFCM which reduces noise impacts on the cluster centers. A threshold  is then defined and data points with  are detected as noise points and are removed from the data.  is used in this work. Consumption data of each consumer have two variables, gas consumption  and HDD which is defined as  where  is temperature. Physical units of these variables are different. Therefore, Mahalanobis distance should be used instead of Euclidean distance.

Three datasets with two, four and one noise points are shown in Fig. 9 (a). These are gas consumption data of three consumers collected during three years. The noise points are detected by GEPFCM in Fig. 9 (b) where  and  indicate noise points and cluster centers computed by GEPFCM, respectively. Variation of  with  is shown in Fig. 9 (c).

|  |
| --- |
|  |
|  |
|  |

Fig. 9. (a) Dataset, (b) noise points detected by GEPFCM and (c) variation of  with .

As shown in Fig. 10, rejection of noise points considerably improves of curve fitting. It is well-known that gas consumption GC increases with HDD whereas the noise point in the third dataset distorts this fact and causes the fitted line to indicate that gas consumption decreases with HDD.

|  |  |
| --- | --- |
| (a) | (b) |
|  |  |
|  |  |
|  |  |

Fig. 10. Fitting a straight line to (a) noisy data and (b) denoised data.

**5.2. More experiments on real datasets**

We cluster datasets with higher numbers of outliers to study PFCM and GEPFCM performance. Six real noisy datasets of gas consumption with cluster centers computed by PFCM (indicated by ×) and GEPFCM (indicated by +) are shown in Fig. 11. Cluster centers calculated by PFCM algorithm are highly influenced by noise points and are displaced towards the noises whereas those computed by GEPFCM algorithm are insensitive to noise points because the exponential functions damp noise impacts on the cluster centers.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Fig. 11. Clustering six noisy real datasets using PFCM and GEPFCM algorithms.

XBI and DBI are computed for these datasets using both PFCM and GEPFCM algorithms and are given in Table 5. Quality of clustering by GEPFCM is better than that of PFCM in terms of both indices.

Table 5. XBI and DBI for six real noisy datasets calculated by PFCM and GEPFCM algorithms.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | DATA1 | DATA2 | DATA3 | DATA4 | DATA5 | DATA6 |
| PFCM | XBI | 0.6942 | 0.8012 | 0.6538 | 0.6625 | 0.9336 | 0.7699 |
| DBI | 1.2813 | 1.7816 | 0.8764 | 1.2219 | 0.5945 | 1.1373 |
| GEPFCM | XBI | 0.6600 | 0.6359 | 0.4837 | 0.5183 | 0.8059 | 0.7139 |
| DBI | 1.1657 | 1.4247 | 0.6974 | 0.7139 | 0.7764 | 1.1138 |

Runtime of PFCM and GEPFCM algorithms for clustering these datasets are given in Table 6. Runtime of GEPFCM algorithm is higher than that of PFCM algorithm because of nonlinear updating equation for cluster centers.

Table 6. Runtime (seconds) of PFCM and GEPFCM algorithms for clustering six real noisy datasets.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | DATA1 | DATA2 | DATA3 | DATA4 | DATA5 | DATA6 |
| PFCM | 0.0601 | 0.0615 | 0.0702 | 0.0710 | 0.0887 | 0.0455 |
| GEPFCM | 0.5502 | 0.1941 | 0.1699 | 0.1719 | 0.1600 | 0.1732 |

**6. Conclusions**

A Generalized Entropy based Possibilistic Fuzzy C-Means algorithm (GEPFCM) is proposed in this work for clustering noisy data. GEPFCM algorithm outperforms the traditional PFCM algorithm. Experiments on different datasets show that error of GEPFCM algorithm is more than 80% smaller than that of PFCM algorithm. GEPFCM is applied to real gas consumption data to remove noise points and improve quality of curve fitting. It is shown that noise points are correctly rejected by this algorithm.

**References**

* 1. F. Zhao, Fuzzy clustering algorithms with self-tuning non-local spatial information for image segmentation, Neurocomputing, 106 (2013) 115-125.
  2. Y. Han, P. Shi, An improved ant colony algorithm for fuzzy clustering in image segmentation, Neurocomputing, 70 (2007) 665-671.
  3. F. Zhao, H. Liu, J. Fan, A multiobjective spatial fuzzy clustering algorithm for image segmentation, Applied Soft Computing, 30 (2015) 48-57.
  4. S. Agrawal, R. Panda, L. Dora, A study on fuzzy clustering for magnetic resonance brain image segmentation using soft computing approaches, Applied Soft Computing 24 (2014) 522-533.
  5. H. Zhu, X. Pan, Robust fuzzy clustering using nonsymmetric student's t finite mixture model for MR image segmentation, Neurocomputing, 175 (2016) 500-514.
  6. S. Askari, N. Montazerin, A high-order multi-variable Fuzzy Time Series forecasting algorithm based on fuzzy clustering, Expert Systems with Applications, 42 (2015) 2121-2135.
  7. S. Askari, N. Montazerin, M.H. Fazel Zarandi, A clustering based forecasting algorithm for multivariable fuzzy time series using linear combinations of independent variables, Applied Soft Computing, 35 (2015) 151-160.
  8. F.M. Pouzols, A.B. Barros, Automatic clustering-based identification of autoregressive fuzzy inference models for time series, Neurocomputing, 73 (2010) 1937-1949.
  9. M. Hanmandlu, O.P. Verma, S. Susan, V.K. Madasu, Color segmentation by fuzzy co-clustering of chrominance color features, Neurocomputing, 120 (2013) 235-249.
  10. R. Eslamloueyan, Designing a hierarchical neural network based on fuzzy clustering for fault diagnosis of the Tennessee-Eastman process, Applied Soft Computing, 11 (2011) 1407-1415.
  11. S. Ghosh, S. Mitra, R. Dattagupta, Fuzzy clustering with biological knowledge for gene selection, Applied Soft Computing, 16 (2014) 102-111.
  12. J.C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm, Computers & Geosciences, 10 (2-3) (1984) 191-203.
  13. J.C. Bezdek, A convergence theorem for the fuzzy ISODATA clustering algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2 (1) (1980) 1-8.
  14. L. Groll, J. Jakel, A new convergence proof of fuzzy c-means, IEEE Transactions on Fussy Systems, 13 (2005) 717-720.
  15. R.J. Hathaway, J.C. Bezdek, Local convergence of the fuzzy c-means algorithms, Pattern Recognition, 19 (6) (1986) 477-480.
  16. W. Wei, J.M. Mendel, Optimality tests for the fuzzy c-means algorithm, Pattern Recognition, 27 (11) (1994) 1567-1573.
  17. T. Kim, J.C. Bezdek, R.J. Hathaway, Optimality tests for fixed points of the fuzzy c-means algorithm, Pattern Recognition, 21 (6) (1988) 651-663.
  18. L. Szilagyi, S.M. Szilagyi, Generalization rules for the suppressed fuzzy c-means clustering algorithm, Neurocomputing, 139 (2014) 298-309.
  19. M.S. Kamel, S.Z. Selim, New algorithms for solving the fuzzy clustering problem, Pattern Recognition, 27 (3) (1994) 421-428.
  20. N. Labroche, Online fuzzy medoid based clustering algorithms, Neurocomputing, 126 (2014) 141–150.
  21. F. de A.T. de Carvalho, F.M. de Melo, Y. Lechevallier, A multi-view relational fuzzy c-medoid vectors clustering algorithm, Neurocomputing, 163 (2015) 115–123.
  22. K.S. Al-Sultan, S.Z. Selim, A global algorithm for the fuzzy clustering problem, Pattern Recognition, 26 (9) (1993) 1357-1361.
  23. X. Qiu, Y. Qiu, G. Feng, P. Li, A sparse fuzzy c-means algorithm based on sparse clustering framework, Neurocomputing, 157 (2015) 290-295.
  24. J. Yu, M-S. Yang, A Generalized Fuzzy Clustering Regularization Model With Optimality Tests and Model Complexity Analysis, IEEE Transactions on Fuzzy Systems, 15 (5) (2007) 904-915.
  25. Y. Ding, X. Fu, Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm, Neurocomputing, (2016), In Press.
  26. B.A. Pimentel, R.M.C.R. de Souza, Multivariate Fuzzy C-Means algorithms with weighting, Neurocomputing, 174 (2016) 946–965.
  27. K-L. Wu, Analysis of parameters elections for fuzzy c-means, Pattern Recognition, 45 (2012) 407-415.
  28. K-L. Wu, M-S. Yang, Alternative c-means clustering algorithms, Pattern Recognition, 35 (2002) 2267-2278.
  29. P-L. Lin, P-W. Huang, C.H. Kuo, Y.H. Lai, A size-insensitive integrity-based fuzzy c-means method for data clustering, Pattern Recognition, 47 (2014) 2042-2056.
  30. T. Havens, J.C. Bezdek, C. Leckie, L. Hall, M. Palaniswami, Fuzzy c-means algorithms for very large data, IEEE Transactions on Fuzzy Systems, 20 (6) (2012) 1130-1146.
  31. R. Krisnapuram, H. Frigui, O. Nasroui, Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation-Part I, IEEE Transactions on Fuzzy Systems, 3 (1) (1995) 29-43.
  32. R. Krisnapuram, H. Frigui, O. Nasroui, Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation-Part II, IEEE Transactions on Fuzzy Systems, 3 (1) (1995) 44-60.
  33. N.R. Pal, K. Pal, J.C. Bezdek, A mixed c-means clustering model, IEEE International Conference on Fuzzy Systems, (1997) 11-21.
  34. R. Krishnapuram, J. Keller, A possibilistic approach to clustering, IEEE Transactions on Fuzzy Systems, 1 (2) (1993) 98-110.
  35. R. Krishnapuram, J. Keller, The possibilistic c-Means algorithm: Insights and recommendations, IEEE Transactions on Fuzzy Systems, 4 (3) (1996) 385-393.
  36. D. Anderson, J.C. Bezdek, M. Popescu, J. Keller, Comparing fuzzy, probabilistic, and possibilistic partitions, IEEE Transactions on Fuzzy Systems, 18 (5) (2010) 906-917.
  37. N.R. Pal, K. Pal, J.M. Keller, J.C. Bezdek, A Possibilistic Fuzzy c-Means Clustering Algorithm, IEEE Transactions on Fuzzy Systems, 13 (4) (2005) 517-530.
  38. R.P. Li, M. Mukaidono, A maximum-entropy approach to fuzzy clustering, IEEE International Conference on Fuzzy Systems, 4 (1995) 2227-2232.
  39. R.P. Li, M. Mukaidono, Gaussian clustering method based on maximum-fuzzy-entropy interpretation, Fuzzy Sets and Systems, 102 (1999) 253-258.
  40. G.J. Gan, J.H. Wu, A convergence theorem for the fuzzy subspace clustering (FSC) algorithm, Pattern Recognition, 41 (2008) 1939-1947.
  41. J.Z.C. Lai, E.Y.T. Juan, F.J.C. Lai, Rough clustering using generalized fuzzy clustering algorithm, Pattern Recognition, 46 (2013) 2538-2547.
  42. C. Zhang, Y. Zhou, T. Martin, A validity index for fuzzy and possibilistic c-means algorithm, Proceedings of IPMU'08, (2008) 877-882.

1. Corresponding Author: [mntzrn@aut.ac.ir](mailto:mntzrn@aut.ac.ir), [mntzrn@hotmail.com](mailto:mntzrn@hotmail.com) [↑](#footnote-ref-2)