Neurocomputing 127 (2014) 266–280

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Agreement-based fuzzy C-means for clustering data with blocks of features

Hesam Izakian [a,*], Witold Pedrycz [a,b,c]

[a] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada
[b] Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[c] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

ARTICLE INFO

Article history:
Received 22 March 2013
Received in revised form
18 June 2013
Accepted 21 August 2013
Communicated by: M. Sato-Ilic
Available online 7 October 2013

Keywords:
Fuzzy C-means clustering
Euclidean distance
Collaborative clustering
Consensus-based clustering
Particle swarm optimization

ABSTRACT

In real-world problems we encounter situations where patterns are described by blocks (families) of features where each of these groups comes with a well-expressed semantics. For instance, in spatiotemporal data we are dealing with spatial coordinates of the objects (say, x–y coordinates) while the temporal part of the objects forms another collection of features. It is apparent that when clustering objects being described by families of features, it becomes intuitively justifiable to anticipate their different role and contribution to the clustering process of the data whereas the clustering is sought to be reflective of an overall structure in the data set. To address this issue, we introduce an agreement based fuzzy clustering—a fuzzy clustering with blocks of features. The detailed investigations are carried out for the well-known algorithm of fuzzy clustering that is fuzzy C-means (FCM). We propose an extended version of the FCM where a composite distance function is endowed with adjustable weights (parameters) quantifying an impact coming from the blocks of features. A global evaluation criterion is used to assess the quality of the obtained results. It is treated as a fitness function in the optimization of the weights through the use of particle swarm optimization (PSO). The behavior of the proposed method is investigated in application to synthetic and real-world data as well as a certain case study.

1. Introduction

In real-world applications there are a number of situations where objects are composed (described) by families of semantically different blocks of features exhibiting significantly different characteristics. For example, in image processing, each pixel comes with its spatial position, the level of brightness, color, etc [9], or in multivariate time series, each variable captures a certain aspect (temporal nature) of the overall phenomenon under discussion. Spatiotemporal data form another example of data of interest—here we encounter spatial and temporal variables as the two separate and semantically distinct entities. Finding available structure in this type of data is an active area of research with numerous possible applications. For example, in evaluating economic growth of countries there are various parameters including purchasing power, unemployment rate, gross domestic product (GDP) etc. To group and analyze countries with similar economic growth, one alternative is to cluster these data. As the second example, we can refer to event detection systems. Here a number of data streams of different nature come into the system. One may employ clustering techniques to reveal the available structure within these data to detect and characterize incident anomalies. Revealing the existing climate patterns in a geographical region is another example that can be fulfilled by employing clustering techniques over a set of spatio-temporal climate data.

Clustering is a useful tool for understanding and visualizing available structures in data. Fuzzy C-means (FCM) proposed by Dunn [1] and Bezdek [11] is one of the commonly used and efficient objective function-based clustering techniques. In this method, instead of assigning each object to a single cluster, the Boolean class membership is relaxed by admitting membership grades assuming values in the unit interval.

In this study, we introduce a generalized version of fuzzy C-means clustering to cluster data with blocks of features coming from distinct sources. Clustering of this type of data poses some significant challenges. First, the diverse dimensionality and the range of the features originating from the corresponding data sources may easily lead to bias towards some data sources when carrying out clustering. Moreover, each data source comes with its own structure and a notion of distance could have a different meaning. It becomes apparent that in comparison with the generic FCM, we seek for clustering capable of dealing with the diversity of the blocks of features. One of the alternatives sought here comes

* Corresponding author. Tel.: +1 780 7169026.
E-mail addresses: izakian@ualberta.ca (H. Izakian), wpedrycz@ualberta.ca (W. Pedrycz).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.08.006

in the form of agreement-based clustering where clustering is intended to form a structure while achieving a significant level of structural "agreement" among all blocks (and the corresponding structures). For this purpose, we investigate the use of an augmented distance function in which distances computed for the individual blocks of features are aggregated (concatenated) by means of some weights. These weights are used to control the impact coming from each block of features to the clustering process. Their optimization is realized by minimizing a certain performance index.

Clustering objects with blocks of features originating from distinct sources or different data sites has been considered in number of studies coming usually under the name of collaborative clustering [7,22,23,25,29] and consensus-based clustering [5,6,8,12–21]. Fig. 1 shows an essence of collaborative clustering.

As shown in this figure, in collaborative clustering there is some communication between different data sources, and the algorithm looks for structure in each source by considering some hints coming from some other sources. These hints take on a format of partition matrices [23], prototypes [29] or proximity matrices [7,22]. Fig. 2 shows the overall scheme of consensus-based clustering techniques.

In this category, usually the available information about the existing structure in data sources is collected in the form of cluster labels or partition matrices, and a new feature space (or similarity measure) is constructed using these guidance mechanisms. Subsequently the algorithm re-clusters the data using the new feature space.

Strehl and Ghosh [5] proposed normalized mutual information to evaluate the shared information among initial clusters. Three heuristics namely cluster-based similarity partitioning algorithm (CSPA), HyperGraph partitioning algorithm (HGPA), and Meta-CLustering Algorithm (MCLA) have been used to form consensus with a high level of shared information. As the initial clusters in these methods were hard clusters, authors in [8] extended the above heuristics to deal with fuzzy clusters as initial clusters for building consensus. In [13] authors modeled the initial clusters coming from different data sources using a bipartite graph. A graph partitioning method has been used to form a final consensus. In [14], initial clusters of different data sources are viewed as independent sources of evidence of structures in data, and a voting mechanism has been used to generate a similarity matrix among objects. Finally the objects are clustered using a hierarchical agglomerative clustering algorithm by considering the new similarity measure.

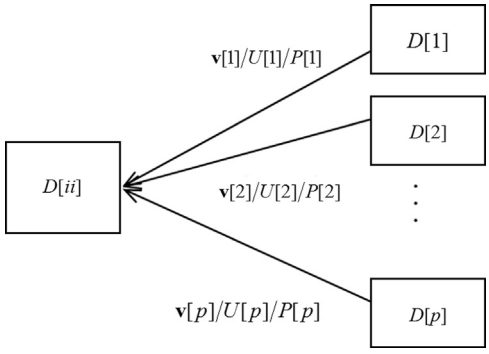


Fig. 1. Overall scheme of collaborative clustering. **v**, *U*, and *P* stand for cluster centers (prototypes), partition matrix and proximity matrix, respectively.

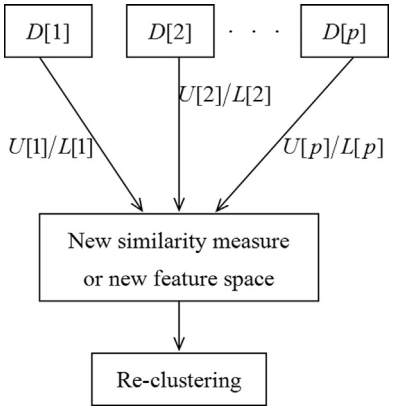


Fig. 2. Overall scheme of consensus-based clustering. *U* and *L* stand for partition matrix and cluster labels, respectively.

Ayad and Kamel [15] proposed a cumulative voting algorithm for different number of clusters to build computationally efficient consensus. In this method, a probabilistic mapping is introduced for cluster label alignment. In [18] a voting mechanism has been formulated as a multi-response regression problem to form consensus from an aggregated ensemble representation. In [17] authors proposed two fast and efficient centroid based ensemble merging algorithms that combine partitions of data comparable to the best existing approaches and are scalable to extremely large data sets. In [19] a partition relevance analysis is considered to estimate the significance of partition matrices before combining them and a new similarity measure between partition matrices has been proposed.

In [12] a consensus-driven fuzzy clustering is proposed. In this method some proximity matrices are constructed using partition matrices from different data sites. The objective of the algorithm was to from a consensus over a data site to preserve its original structure, while minimize the distance of its corresponding proximity matrix from the other proximity matrices available in other data sites. A gradient-based method has been used to realize optimization and form the final consensus results. Pedrycz [9] proposed a method to cluster semantically distinct families of variables. In this method, a prediction criterion has been used to optimize the effect of variables in the clustering process.

Most of the collaborative and consensus-based clustering methods proposed in literature, work in a passive form. In these methods, instead of using the feature space of data sources for clustering, they use the results of clusterings (in from of cluster labels or partition matrix) that has been performed over each data source separately. The proposed method in this paper supports an active mode while the feature space of the individual data sources is exploited to form an overall feature space in which clustering takes place. Fig. 3 visualizes the essence of the problem in which we aim at clustering objects with features coming from distinct data sources.

The objective of the proposed approach is to control the effect of each source of data (blocks of features) in the clustering process

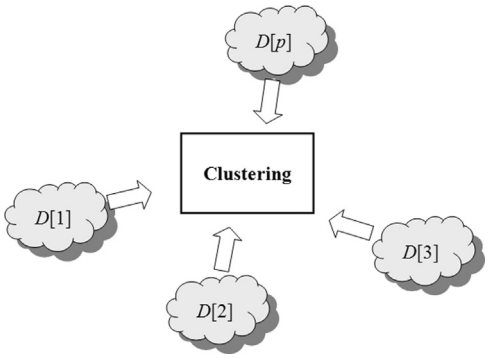


Fig. 3. The essence of the agreement-based clustering.

to achieve some generally agreed upon clusters among all data sources that preserve their original structures. A composite distance function along with an evaluation criterion that quantifies the level of agreement among all data sources has been introduced to optimize the effect of each source of data (block) in clustering.

The proposal advocated here exhibits an evident facet of originality. With this regard, it is worth contrast the proposed approach with the algorithms encountered in the literature. A visible approach or a collection of similar approaches concerns a weighting scheme where the features are weighted. There could be weighting schemes realized for individual clusters. For instance, authors in [27] proposed a feature weighting assignment to improve some clustering validity indices being regarded as sound evaluation criteria. A gradient descent technique has been employed to learn the feature weights. In [28] a feature weighting approach using a weighted Euclidean distance function was proposed to specify an influence of different features for each cluster. To do so, the FCM objective function has been revisited and optimized for the new distance function.

Let us stress that the method proposed here exhibits some fundamental differences in comparison with the mechanism of feature weighting. The essential differences are as follows:

- Feature weighting methods assign a weight to each single feature, while our proposed method assign a weight to each block of features. To clarify the difference between these two methods, let us assume clustering a multivariate time series data having five variables each comprising a time series with length 30. The proposed method assigns a single weight to each variable and as the result there are five weights in total. On the other hand, feature weighting approaches have to assign a weight to each feature and as the result there are 150 weights. It is clear that optimizing 150 weights is a challenging problem. Moreover, weighting each feature in a time series for clustering could be meaningless.
- Feature weighting methods assume that all the features in data are of the same nature. As an example, let us consider a spatiotemporal data with two features associated with the spatial part (e.g. x–y coordinates) and 100 features for temporal part (e.g. a time series). Clustering this type of data using feature weighting methods leads to a bias towards the temporal part of data. However, finding the optimal weights for 102 features still is a challenging problem.
- In the feature weighting approaches the influence of each feature is increased or decreased based on some performance indices, and as the result, the final revealed structure may be less or more suitable for each feature. In the proposed method, all blocks of features exhibit the same importance in the clustering process and the algorithm tends to generate a final structure that is suitable for all data sources.

This paper is organized into five sections. In Section 2, the problem is formulated and an augmented fuzzy C-means clustering is proposed. Section 3 introduces an evaluation criterion and elaborates on its optimization procedure. Section 4 reports the experimental studies dealing with synthetic and real-world data as well as a real–world application. Conclusions are covered in Section 5.

2. Fuzzy clustering with blocks of features

Let us consider n objects $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ whose features are coming from p data sources (blocks) $D[1]$, $D[2]$,…,$D[p]$. Each data source describes the objects from a different point of view. By concatenating these features, each object is described with the use of $\mathbf{x}_k = [\mathbf{x}_k(1) \mid \mathbf{x}_k(2) \mid \ldots \mid \mathbf{x}_k(p)]^T$, $k=1, 2, \ldots, n$, where $\mathbf{x}_k(j)$ is the feature vector corresponding to jth data source, $D[j]$, for kth object. Since in each data source like $D[j]$ there are r_j features, altogether we have the following representation:

$$\mathbf{x}_k = [x_{k1}(1), x_{k2}(1), \cdots, x_{kr_1}(1) \mid \ldots \mid x_{k1}(p), x_{k2}(p), \cdots, x_{kr_p}(p)]^T \tag{1}$$

Note that number of features in different data sources can be different. In this paper we propose a fuzzy C-means clustering to deal with this type of data. The FCM method partitions n objects into c fuzzy clusters. The result is a collection of c prototypes $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c$ and a partition matrix U describing the membership degrees of objects to clusters, where $U=[u_{ik}]$, $i=1,2,\ldots,c$, $k=1,2,\ldots,n$, $u_{ik} \in [0, 1]$, $\sum_{i=1}^{c} u_{ik} = 1 \ \forall k$, and $0 < \sum_{k=1}^{n} u_{ik} < n \ \forall i$. This result arises through the minimization of the following objective function

$$J = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^m d^2(\mathbf{v}_i, \mathbf{x}_k) \tag{2}$$

where $(m > 1)$ is fuzzification coefficient and $d(.)$ stands for a distance function. Usually the Euclidean distance has been considered in literature.

To deal with the data structure represented in (1), in this paper we define the following distance function between object $\mathbf{x}_k$ and prototype $\mathbf{v}_i$

$$d^2_{\lambda_1,\ldots,p}(\mathbf{v}_i, \mathbf{x}_k) = \lambda_1 \ \|\mathbf{v}_i(1) - \mathbf{x}_k(1)\|^2 + \lambda_2 \ \|\mathbf{v}_i(2) - \mathbf{x}_k(2)\|^2 + \cdots + \lambda_p \ \|\mathbf{v}_i(p) - \mathbf{x}_k(p)\|^2 \tag{3}$$

where $\sum_{j=1}^{p} \lambda_j = 1$, $0 \le \lambda_j \le 1$ for all j and $\|.\|$ denotes the Euclidean distance.

Using the distance function specified above, the impact of each data source in the clustering process can be easily controlled. Assigning $\lambda_j = 0$, removes the contribution of data source $D[j]$ to the overall clustering process, while $\lambda_j = 1$, removes the contribution of other data sources and considers only $D[j]$ in the clustering process. Higher values of λ_j increase the impact of $D[j]$ and decrease the impact of the other data sources in the clustering process. Considering (3) as the distance function, the FCM objective function is expressed as

$$J = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^m d^2_{\lambda_1,\ldots,p}(\mathbf{v}_i, \mathbf{x}_k). \tag{4}$$

The minimization of J is realized through an iterative process in which we successively compute the prototypes and the partition matrix in the form:

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} u_{ik}^m \mathbf{x}_k}{\sum_{k=1}^{n} u_{ik}^m} \tag{5}$$

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(d_{\lambda_1,\ldots,p}(\mathbf{v}_i, \mathbf{x}_k) / d_{\lambda_1,\ldots,p}(\mathbf{v}_j, \mathbf{x}_k)\right)^{2/(m-1)}} \tag{6}$$

The detailed derivations are provided in Appendix A.

3. Evaluation criterion

In previous section, we described a fuzzy clustering approach to deal with data with blocks of features coming from distinct sources. As the weights $\lambda_1, \lambda_2, \ldots, \lambda_p$ in the introduced distance function control the effect (impact) of each data source in the clustering process, the quality of clusters directly depends on them. In this section, we propose an evaluation criterion to optimize these embedded weights.

Since our objective is to reveal a general structure over all data sources, this structure should have a high level of "agreement" among the available structures in separate data sources. To measure the level of agreement, the FCM objective function has been

considered. Assuming that U is the partition matrix resulting from clustering objects (with blocks of features) using the proposed distance function in (3), the quality of the clusters can be quantified using the following evaluation criterion

$$Q = \frac{J(D[1]|U)}{J(D[1])} + \frac{J(D[2]|U)}{J(D[2])} + \cdots + \frac{J(D[p]|U)}{J(D[p])} \tag{7}$$

where $J(D[j]|U)$ is the value of the FCM objective function for data source $D[j]$ by considering U as its partition matrix and calculating its prototypes. $J(D[j])$ stands for the FCM objective function obtained when clustering data source $D[j]$ separately. In fact, (7) expresses how much the revealed general structure, U, is suitable (acceptable) for each separate data source in terms of the corresponding FCM objective function. Because the feature spaces for distinct data sources exhibit various magnitude and dimensionality, $J(D[j])$, $j=1,2,\ldots, p$ used as denominator in (7) serves as a normalization term. Moreover, since in clustering each data source separately, the other sources are not taken into account, the resulting partition matrix is the optimal one for this particular data source and obviously $J(D[j]|U) \geq J(D[j])$, and as the result the inequality $Q \geq p$ always holds. In the case $Q = p$, the available structures determined for distinct data sources are in a perfect agreement and the resulting structure by the proposed method is exactly the same as the available structures in various data sources. Lower value of Q indicates that the formed general structure is at a higher level of agreement with distinct data sources, while higher value of Q is indicative of a lower level of agreement. Therefore, the problem of finding optimal weights $\lambda_1, \lambda_2, \ldots, \lambda_p$ can be considered as an optimization problem: determine the values of $\lambda_1, \lambda_2, \ldots, \lambda_p$ in order to minimize Q. Since checking all the possible combinations of values of the weights is time consuming (especially for higher number of data sources), using a meta-heuristic algorithm to find near-optimal weights could be a viable alternative.

There are numerous works reported in the literature (e.g. [24,30]) exploiting the merits of evolutionary algorithms in clustering. In this study, a particle swarm optimization (PSO) [10] is used as an efficient population based searching algorithm to find (near) optimal weights. PSO starts with a number of potential solutions (called particles) and in some iteration tries to improve the quality of particles using some searching strategies. Since the problem search space here is p-dimensional, each particle is encoded as a vector with p elements $\lambda_1, \lambda_2, \ldots, \lambda_p$ following the constraints imposed in (3). In the first step of the algorithm, number of particles and their corresponding velocity vectors (with p elements in a pre-specified range) are generated randomly. For each particle, the encoded weights are used to produce a general structure over all data sources and the proposed evaluation criterion in (7) is considered as the quality (fitness) of that particle. In each iteration of the algorithm, the velocity vectors and the particles are updated using (8) and (9), respectively.

$$y_{ki}^{t+1} = w \times y_{ki}^{t} + c_1 r_{1i}(pbest_{ki}^{t} - z_{ki}^{t}) + c_2 r_{2i}(gbest_{i}^{t} - z_{ki}^{t}), \tag{8}$$

$$k=1, 2,\ldots, N,\ i=1, 2, \ldots, p,\ y_{ki}^{t} \in [y_{\min}, y_{\max}]$$

$$z_{ki}^{t+1} = z_{ki}^{t} + y_{ki}^{t+1},\ z_{ki}^{t} \in [0,\ 1] \tag{9}$$

where y_{ki}^{t} is the ith element of the velocity of the kth particle in tth step, z_{ki}^{t} is the ith element of the kth particle in tth step of the algorithm, N is the number of particles in the swarm and p is the dimensionality of the search space (number of data sources here). Also *pbest* (personal best) is the best solution the particle has revealed and *gbest* (global best) is the best solution the whole swarm has obtained during the search process, w is inertia weight, r_{1i} and r_{2i} are random values in range [0, 1] sampled from a uniform distribution and c_1 and c_2 are acceleration coefficients, controlling the impact of *pbest* and *gbest* in the search process. The algorithm improves the quality of solutions in number of iterations and finally, the best particle (with the best fitness value) is considered as the (near) optimal weights.

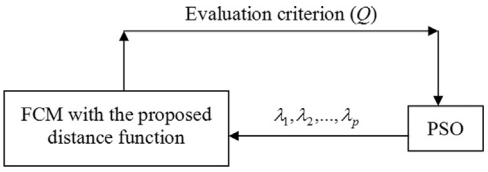


Fig. 4. The overall scheme of the proposed method.

Fig. 4 shows the overall computing scheme of the proposed method. At the first step of the algorithm, PSO generates a set of particles each comprising p weights, $\lambda_1, \lambda_2, \ldots, \lambda_p$. For each particle, these weights are exploited to cluster the objects with distinct data sources (using the composite distance function in (3)) and the quality of clusters is evaluated using the proposed criterion in (7) which serves as the fitness function of the PSO. In the next step, PSO manipulates the generated particles using calculated fitness values to improve their quality.

4. Experimental studies

In this section, we illustrate the proposed method by using a synthetic data set, three data sets coming from the UCI machine learning repository, and the Alberta climate data.

4.1. Synthetic data

For illustrative purposes and in order to clarify the performance of the proposed evaluation criterion (7), we generated five data sources and investigated the behavior of the proposed method. Fig. 5(a)–(e) show the data sources $D[1]$ to $D[5]$.

As shown in these figures, each object is composed of 11 features that are associated with five data sources with the following geometries:

- $D[1]$ is a two-dimensional data with features in range [0, 1] and has a visible structure for number three clusters.
- $D[2]$ is a two-dimensional data with features in range [0, 1], but there is no a visible structure in this data source.
- $D[3]$ is a two-dimensional data with features in range [0, 1] and has a visible structure for four clusters.
- $D[4]$ is a two-dimensional data with features in range [0, 2] and has a visible structure for three clusters.
- $D[5]$ is a three-dimensional data with features in range [0, 1] and has a visible structure for three clusters.

Strong (more distinguishable) structure versus *weak (less distinguishable) structure*. In this experiment, we consider two data sources $D[1]$ and $D[2]$ and investigate the effect of the values of λ_1 and λ_2 on the evaluation criterion (Q) to form a general structure for number of clusters $c=3$. The fuzzification coefficient was set to $m=2$ in all experiments. Fig. 6 shows the values of Q versus different values of λ_1. Notice that because of the two data sources in this experiment, we have $\lambda_2 = 1 - \lambda_1$.

As can be seen from this figure, the optimal weights are $\lambda_1 = 0.6$ and $\lambda_2 = 0.4$, that means $D[1]$ has higher impact on forming globally acceptable clusters. The prototypes for these data sources before forming the general structure are as follows:

$$\begin{matrix} \mathbf{v}_1[1] = [0.717 & 0.243] \\ \mathbf{v}_2[1] = [0.734 & 0.798] \\ \mathbf{v}_3[1] = [0.236 & 0.219] \end{matrix} \quad \text{and} \quad \begin{matrix} \mathbf{v}_1[2] = [0.699 & 0.277] \\ \mathbf{v}_2[2] = [0.563 & 0.768] \\ \mathbf{v}_3[2] = [0.201 & 0.376] \end{matrix}$$

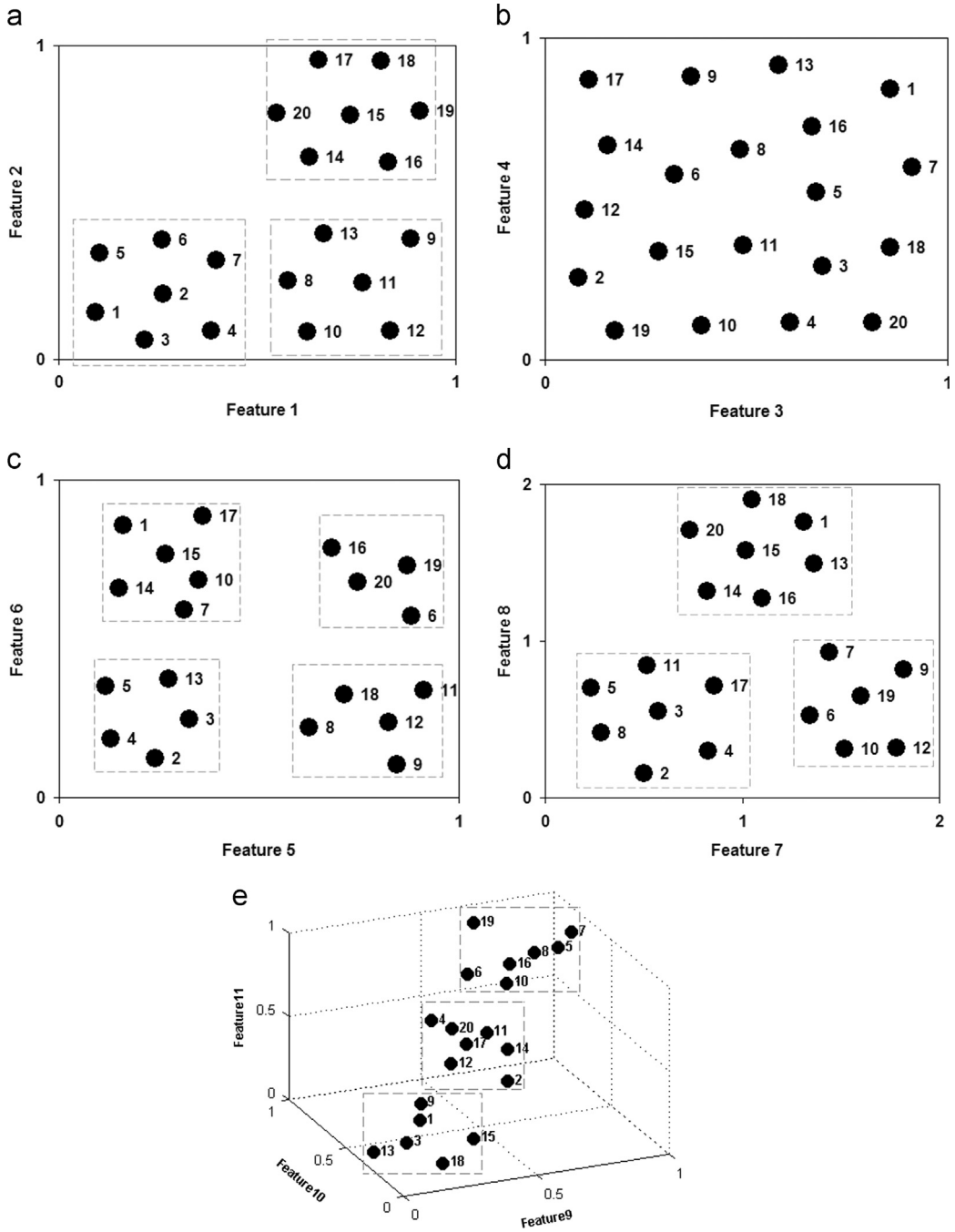


Fig. 5. Five synthetic data sources. (a) $D[1]$, (b) $D[2]$, (c) $D[3]$, (d) $D[4]$, and (e) $D[5]$.

Once the overall general structure has been formed, the updated prototypes are changed to:

$$\mathbf{v}_1[1]=[0.668 \quad 0.323] \qquad \mathbf{v}_1[2]=[0.621 \quad 0.444]$$
$$\mathbf{v}_2[1]=[0.732 \quad 0.752] \quad \text{and} \quad \mathbf{v}_2[2]=[0.432 \quad 0.578]$$
$$\mathbf{v}_3[1]=[0.279 \quad 0.231] \qquad \mathbf{v}_3[2]=[0.392 \quad 0.419]$$

As it can be seen, the prototypes corresponding to data source $D[2]$ exhibit more changes in comparison with the prototypes describing $D[1]$. Since $D[1]$ has a more visible structure, its FCM objective function $J(D[1])$, has lower value in comparison with $D[2]$ and as a result the algorithm pays more attention to $D[1]$ to achieve lower values for $J(D[1]|U)/J(D[1])$. Also one can note that the situation where $\lambda_1=0$ is the worst case in this experiment because of the existence of a stronger structure in $D[1]$.

Let us consider $D[1]$ and $D[3]$ and form the general structure for the number of clusters set to $c=3$ and $c=4$. Fig. 7(a) and (b) show the effect of λ_1 and $\lambda_3=1-\lambda_1$ on the evaluation criterion. The optimal value of λ_1 is 0.55 for $c=3$ (Fig. 7(a)) and 0.45 for $c=4$ (Fig. 7(b)).

For $c=3$, $D[1]$ has a stronger (more visible) structure and we have $\lambda_1>\lambda_3$, while for $c=4$, as $D[3]$ has more visible structure, we have $\lambda_1<\lambda_3$.

Data sources with different magnitudes of features. Let us consider $D[1]$ and $D[4]$. Both of data sources have a visible structure for $c=3$. The magnitude of features in $D[1]$ is in range $[0, 1]$, while for $D[4]$ it is in range $[0, 2]$. Fig. 8 shows Q for different values of λ_1. Similar to the previous experiments, we have $\lambda_4=1-\lambda_1$.

The optimal value of Q (see Fig. 8) occurred around $\lambda_1=0.75$ and $\lambda_4=0.25$. The reason is that the magnitude of features in $D[1]$

is lower than the magnitude of features in $D[4]$ and the algorithm assigns a higher value to λ_1 to prevent bias towards $D[4]$ in the formation of the general structure.

Data sources with different number of features. In this experiment we consider $D[1]$ and $D[5]$. $D[5]$ has three features and a visible structure for $c=3$. Fig. 9 shows the values of Q for different values of λ_1. The optimal Q occurs for higher value of λ_1 ($\lambda_1 > \lambda_5$). The reason is the same as in the previous experiment: considering higher value for λ_1 in order to prevent bias towards $D[5]$ in the clustering process.

Forming general structure for $D[1]$ to $D[5]$. In this experiment we consider all data sources to form a general structure for number of clusters $c=3$ and $c=4$. For the PSO algorithm the following parameters after a fine-tuning has been chosen: number of particles $N=5$ (equal to number of data sources), $c_1=c_2=2$, number of iterations$=10N=50$, range of velocity elements$=[-0.3, +0.3]$. For the number of clusters set to $c=3$, the optimal weights are as follows $[\lambda_1=0.288, \lambda_2=0.184, \lambda_3=0.235, \lambda_4=0.087, \lambda_5=0.209]$,and for $c=4$, the optimal weights are

$[\lambda_1=0.252, \lambda_2=0.185, \lambda_3=0.307, \lambda_4=0.072, \lambda_5=0.184]$.

Overall, $D[1]$ and $D[3]$ have higher weights in comparison with the weights associated with other data sources. $D[2]$, $D[4]$ and $D[5]$ have lower weights because of their weak structure, higher range of features and higher dimensionality, respectively. Also for $c=3$, $\lambda_1 > \lambda_3$, while for $c=4$ $\lambda_1 < \lambda_3$ because of the existing structures in these data sources. Fig. 10(a)–(e) shows the clusters in each data source (visualized in the form of the contour plot of membership degrees) for $c=3$ and Fig. 11(a)–(e) shows the clusters after forming general structure. For $D[5]$ the clusters are plotted over the first two features.

Obviously, there is a significant change in the initial clusters after forming the general structure. Also Fig. 12 shows the PSO convergence process for $c=3$ and $c=4$. The most significant improvements have been observed in the first few generations. Moreover, for $c=4$, Q has a higher value than $c=3$. The reason is that having more clusters means that more details about the available structures in the separate data sources are considered. As the result, the level of agreement between data sources has decreased.

4.2. Real-world data

In this sub section, we consider three real-world data sets from UCI machine learning repository (http://archive.ics.uci.edu/ml/datasets.html) to investigate the performance and functioning of the proposed method.

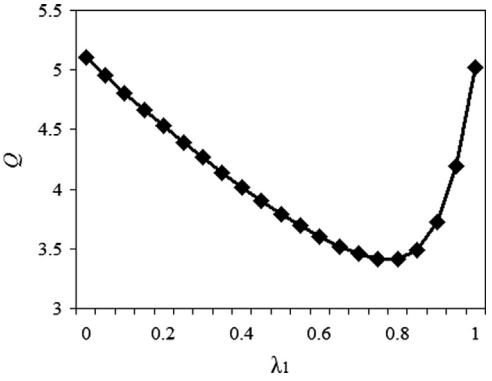


Fig. 8. Q versus different values of λ_1 in forming general structure over $D[1]$ and $D[4]$ for $c=3$.

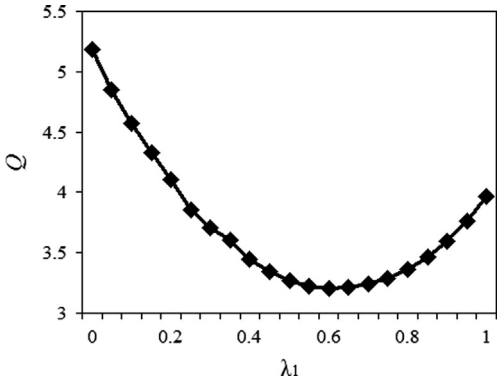


Fig. 6. Evaluation criterion (Q) versus different values of λ_1 in the formation of the general structure over $D[1]$ and $D[2]$.

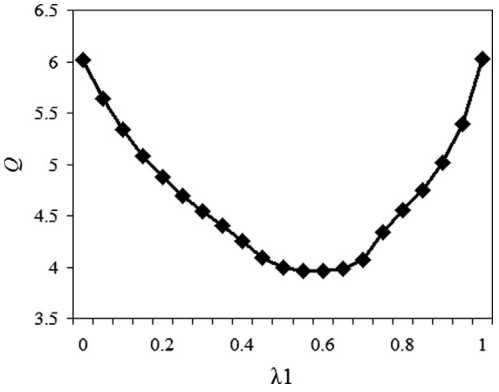


Fig. 9. Q versus different values of λ_1 in forming general structure over $D[1]$ and $D[5]$ for $c=3$.

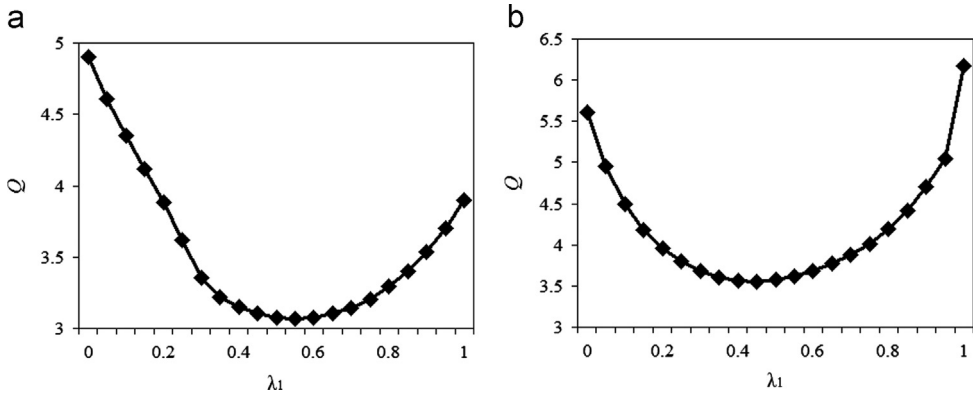


Fig. 7. Evaluation criterion (Q) versus different values of λ_1 in forming general structure over $D[1]$ and $D[3]$. (a) $c=3$, and (b) $c=4$.

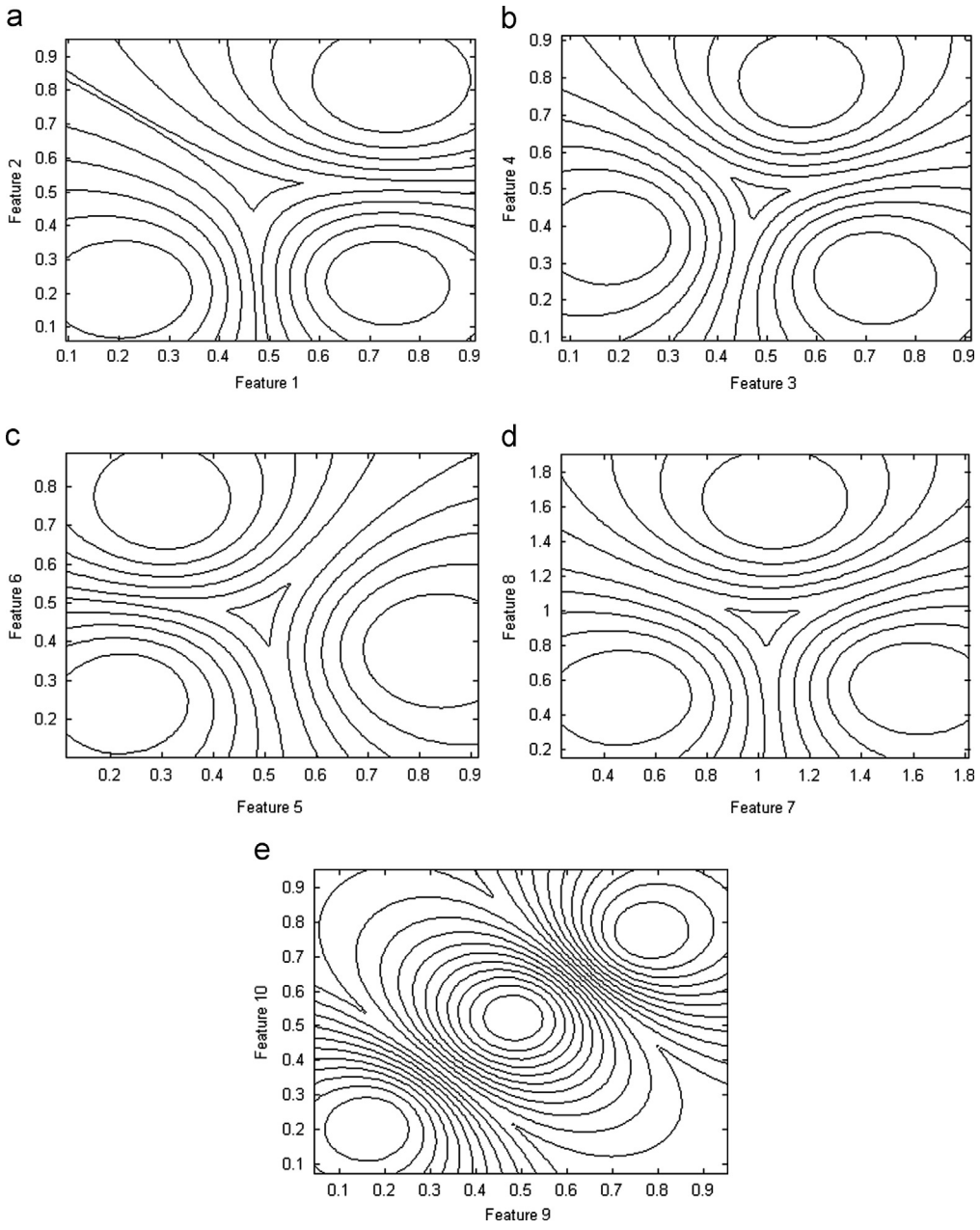


Fig. 10. Contour plot of membership degrees before forming the general structure. (a) $D[1]$, (b) $D[2]$, (c) $D[3]$, (d) $D[4]$ and (e) $D[5]$.

Iris data set: this data set comprises four features describing 150 iris plants. We form two data sources over this data set as follows:

$D[1]$: petal length and width.
$D[2]$: sepal length and width.

Boston Housing data set: this data set concerns housing values in suburbs of Boston based on 13 features. We form three data sources by considering three subsets of features as follows [6]:

$D[1]$: features 1, 5 and 11 (crime rate, nitric oxides concentration, and pupil-teacher ratio) that help characterize the quality of the environment.
$D[2]$: features 6, 7 and 8 (room, age, distances employment centers) to assess the properties of real estate.
$D[3]$: features 12 and 13 (B, lower status) for evaluation based on the characterization of the population.

Fire data set: this data set is composed of 517 forest fires records in the northeast region of Portugal. The aim of this data set was to predict the burned area using 13 different features. In this data set we form three data sources by considering three subsets of semantically different features as follows:

$D[1]$: features 1, 2 (x–y coordinates) that represent the location information of the fires occurrences.
$D[2]$: features 5, 6, 7 and 8 (FFMC, DMC, DC, ISI) that are some indices coming from FWI system.
$D[3]$: features 9, 10, and 11 (temperature, humidity, wind speed) capturing the weather-related information.

Since in each data source there are features in different ranges, the z-score standardization has been applied. For the PSO algorithm, similar to the synthetic data set, the following values of the parameters has been chosen: N, number of particles, is set

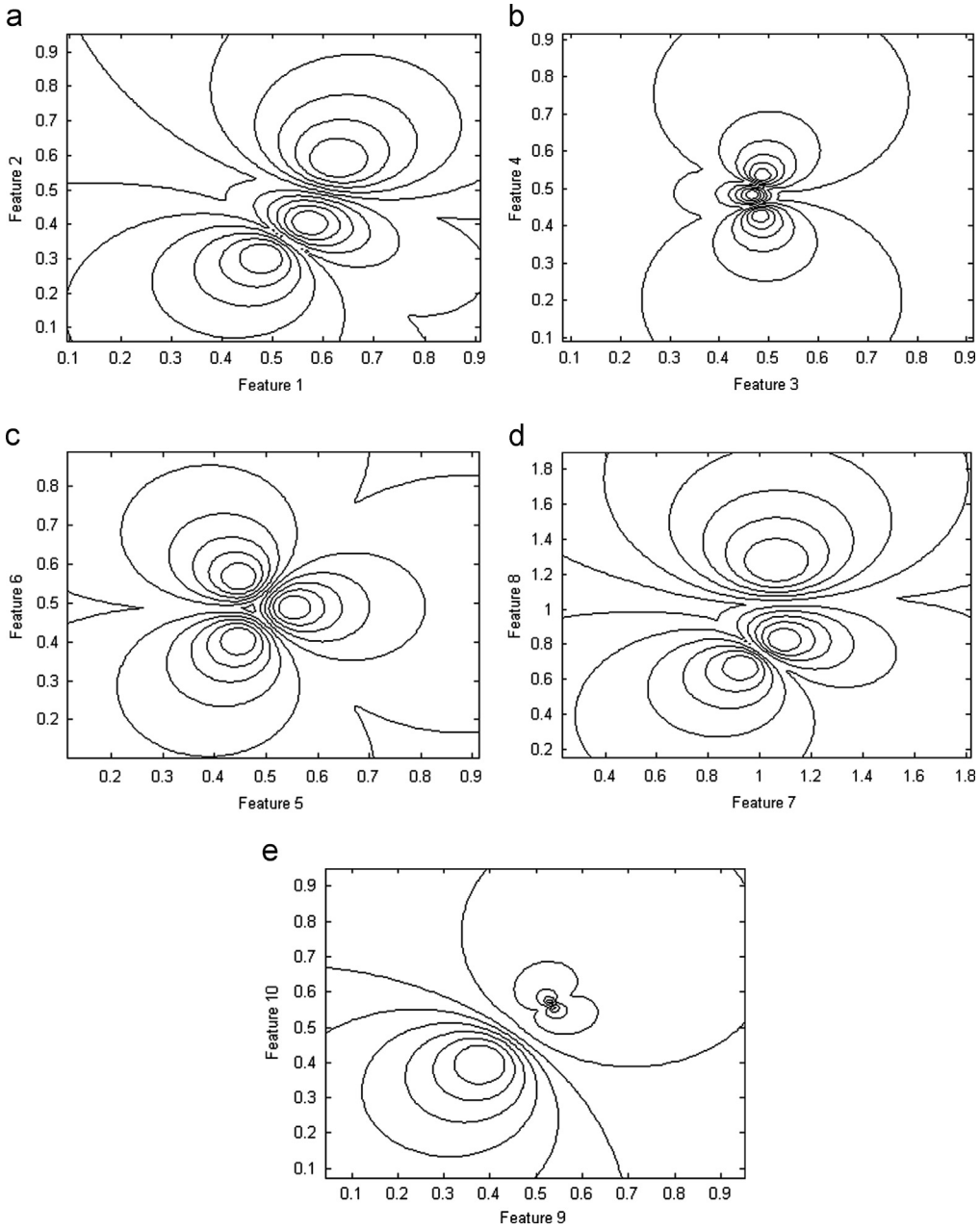


Fig. 11. Contour plot of membership degrees after forming general structure. (a) $D[1]$, (b) $D[2]$, (c) $D[3]$, (d) $D[4]$ and (e) $D[5]$.

equal to the number of data sources, $c_1 = c_2 = 2$, number of iterations $= 10\,N$, range of velocity elements $= [-0.3, +0.3]$. Also to assess the effectiveness of the proposed method we compared it with three following scenarios:

1) In the first scenario, for each separate data source we calculate its partition matrix obtained using FCM and then consider the following criterion to evaluate the average level of agreement among the available structures revealed in separate data sources:

$$Q_{avg} = \frac{1}{p} \sum_{i=1}^{p} \sum_{j=1}^{p} \frac{J(D[j]|U[i])}{J(D[j])} \tag{10}$$

where $U[i]$ is the partition matrix calculated using FCM for data source $D[i]$, $J(D[j]|U[i])$ is the FCM objective function for $D[j]$ by considering $U[i]$ as its partition matrix, and $J(D[j])$ is FCM objective function for the separate data source $D[j]$.

2) In the second scenario, we consider the data source with the highest level of agreement with other data sources and evaluate the level of its agreement using the following criterion:

$$Q_{\min} = \min \sum_{j=1}^{p} \frac{J(D[j]|U[i])}{J(D[j])} \text{ for } i = 1, 2, \ldots, p \tag{11}$$

3) Finally, in the third scenario, we consider "standard" FCM (all the weights in (3) are set to be equal to 1) to cluster all data sources and use (7) to evaluate the level of agreement among data sources (Q_{FCM}).

Table 1 shows the values of the evaluation criteria, Q_{avg}, $Q_{\min}$, Q_{FCM}, and Q for the optimal weights achieved by PSO. For the proposed method, the results are reported as the average and standard deviation of achieved Q in 40 independent runs. Also because of existing two data sources in Iris data set, PSO is not used and instead, a simple enumeration has been exercised.

As can be seen from this table, the proposed method generates clusters of higher quality in terms of the proposed evaluation

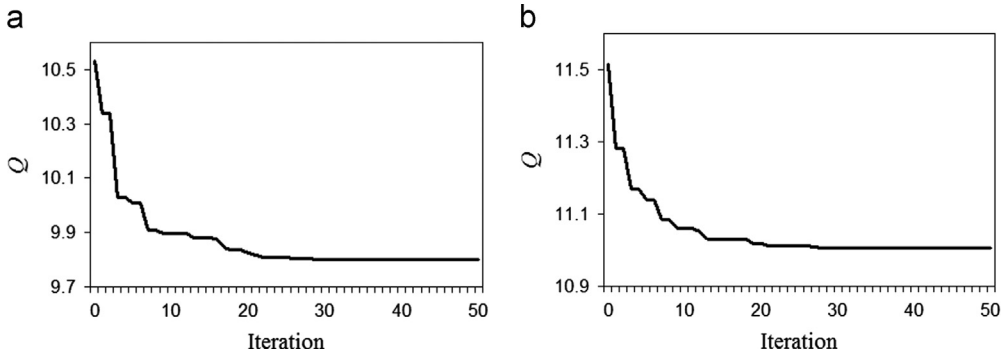


Fig. 12. Convergence of PSO optimization process for (a) $c=3$ and (b) $c=4$.

Table 1
The values of the evaluation criteria for number of clusters $c=2$ to 6. For the proposed method, the results are reported as the average and standard deviation obtained over 40 independent runs.

Data set	Iris				Housing				Fire			
	Q_{avg}	$Q_{\min}$	Q_{FCM}	Q	Q_{avg}	$Q_{\min}$	Q_{FCM}	Q	Q_{avg}	$Q_{\min}$	Q_{FCM}	Q
$c=2$	2.484	2.230	2.202	2.163	4.260	3.975	3.472	3.456 ± 0	4.359	4.197	3.582	3.568 ± 0
$c=3$	3.595	2.758	2.892	2.543	5.716	5.645	4.287	4.192 ± 0	5.216	4.965	3.864	3.812 ± 0
$c=4$	4.294	3.062	3.210	2.668	7.460	6.887	4.866	4.459 ± 0	6.928	6.488	4.316	4.225 ± 0
$c=5$	4.609	3.588	3.379	2.917	8.618	7.891	4.828	4.638 ± 0.012	8.628	7.742	4.654	4.538 ± 0
$c=6$	4.919	4.010	3.398	2.786	10.053	9.489	5.013	4.814 ± 0.023	9.965	8.773	4.991	4.844 ± 0.001

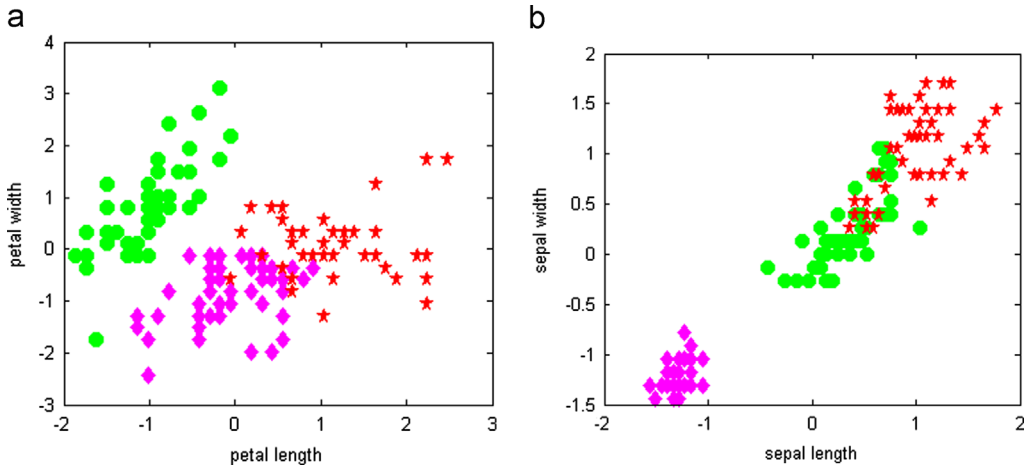


Fig. 13. Revealed clusters using FCM for each data source in Iris data for $c=3$. (a) $D[1]$, and (b) $D[2]$.

criterion. Also in most cases the standard deviation is almost equal to zero (to the third decimal digit) that shows the stability of the proposed method. Moreover, increasing the number of clusters reduces the level of agreement among different data sources. Fig. 13 shows the revealed clusters using FCM in the two defined data sources in Iris data set for $c=3$ and Fig. 14 shows the reveled clusters using the proposed method (optimal weights). There are some slight changes in the revealed clusters. In the proposed method, different data sources collaborate to form the clusters and the density of the collaboration is controlled using the weights (λ) and the evaluation criterion (Q), while in FCM the density of collaboration is the same for all data sources.

4.3. A case study: Alberta climate data

This data set is composed of 173 stations located in Alberta province, Canada. For each station, its spatial coordinates, and the recorded daily average temperature, daily precipitation, and daily average humidity in the form of time series have been provided. These data are available online at www.agric.gov.ab.ca. Fig. 15 shows a snapshot of the system with one highlighted station along with its temperature, precipitation, and humidity time series in 2010.

Since in this data set, for each station there are four sources of data (spatial coordinates, temperature, precipitation, and humidity) we use the proposed method to form some general structures over all data sources.

In this study we use a representation of time series to have shorter and more efficient features for clustering purpose. There are different methods to represent time series data [2]. Two efficient and popular representation methods, namely discrete Fourier Transform (DFT) and Piecewise Aggregate Approximation (PAA) have been used in this paper. Note that selecting the representation method is application dependent and each method captures some special features of time series [26].

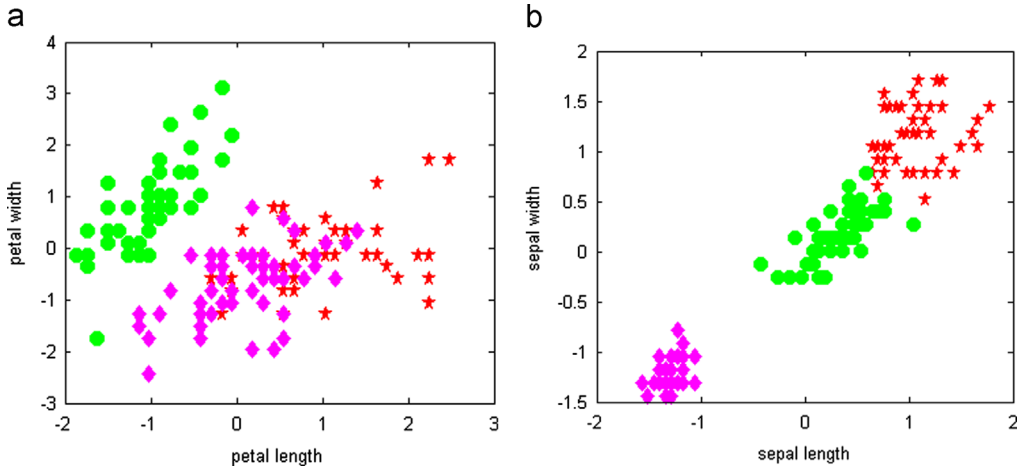


Fig. 14. Revealed clusters using the proposed method for each data source in Iris data for $c=3$. (a) $D[1]$, and (b) $D[2]$.

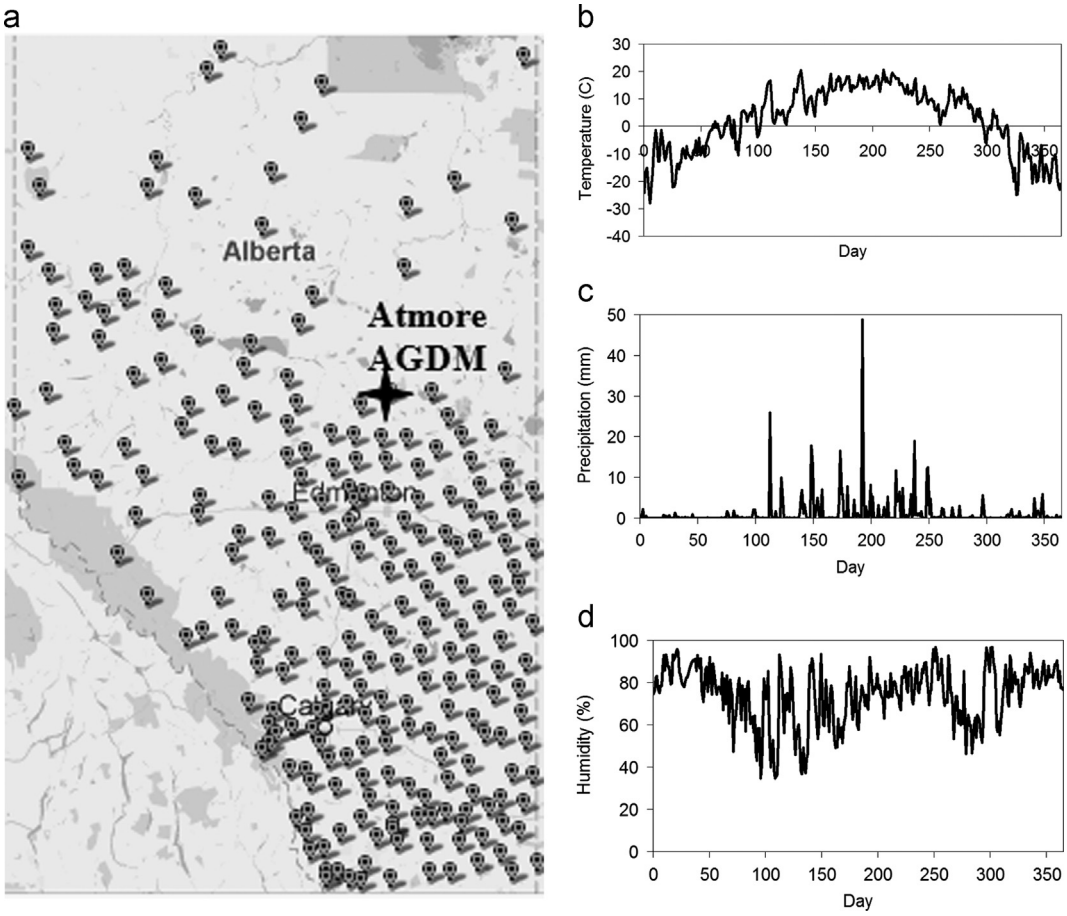


Fig. 15. A snapshot of the Alberta agriculture system. (a) A set of stations in Alberta along with one highlighted station, (b) temperature time series corresponding to the highlighted station in 2010, (c) precipitation time series, and (d) humidity time series.

DFT represents time series in frequency domain. It transforms a sequence of length n, into n complex numbers each describing a sine/cosine wave. Faloutsos et al. [3] noted that the most important features of each sequence are the first k (real and imaginary) coefficients ($k<<n$) of the DFT transform, while the other coefficients are approximately equal to zero.

PAA represents time series in time domain. It divides the time series with length n into k ($k<<n$) equal length segments and determines the mean value of data points within each segment as the representatives [4].

In this experiment DFT and PAA representations with length 8, 16 and 24, and number of clusters 2–5 are considered. Notice that the length of time series representation is application-dependent and higher length of representation includes more details about time series, while lower length of representation hides details.

For the PSO algorithm the number of particles is set equal to the number of data sources ($N=4$) and the number of iterations is $10N$. Table 2 compares the results in terms of the average level of agreements among separate data sources (Q_{avg}), highest available level of agreement among separate data sources ($Q_{\min}$), level of

Table 2
Experimental results for Alberta climate data in 2010. DFT and PAA representations with length 8, 16 and 24, and number of clusters $c=2$, 3, 4, and 5 have been considered. For the optimal weights, the results are reported in the form of average and standard deviation of Q in 40 independent runs.

Representation	$c=2$				$c=3$				$c=4$				$c=5$			
	Q_{avg}	$Q_{\min}$	Q_{FCM}	Q	Q_{avg}	$Q_{\min}$	Q_{FCM}	Q	Q_{avg}	$Q_{\min}$	Q_{FCM}	Q	Q_{avg}	$Q_{\min}$	Q_{FCM}	Q
DFT(8)	5.485	5.184	5.23	4.716 ± 0.034	6.7	6.363	6.208	5.213 ± 0.124	7.689	6.839	6.929	5.575 ± 0.202	7.992	7.682	6.79	5.684 ± 0.168
DFT(16)	4.952	4.712	4.945	4.504 ± 0.017	5.656	5.324	5.619	4.837 ± 0.074	6.091	5.562	5.591	4.946 ± 0.074	6.353	5.683	5.774	5.071 ± 0.064
DFT(24)	4.909	4.789	4.834	4.479 ± 0.047	5.496	5.211	5.121	4.793 ± 0.071	5.828	5.382	5.424	4.842 ± 0.074	6.054	5.505	5.549	4.983 ± 0.056
PAA(8)	5.231	5.079	4.849	4.562 ± 0.013	6.299	6.01	5.662	4.925 ± 0.042	6.892	6.332	5.67	5.08 ± 0.033	7.329	6.733	5.853	5.184 ± 0.047
PAA(16)	4.835	4.768	4.634	4.41 ± 0.012	5.481	5.41	4.995	4.672 ± 0.027	5.792	5.521	5.258	4.765 ± 0.029	5.98	5.663	5.344	4.857 ± 0.033
PAA(24)	4.825	4.718	4.626	4.438 ± 0.045	5.488	5.113	5.007	4.691 ± 0.018	5.73	5.398	5.221	4.773 ± 0.02	5.955	5.5	5.282	4.871 ± 0.075

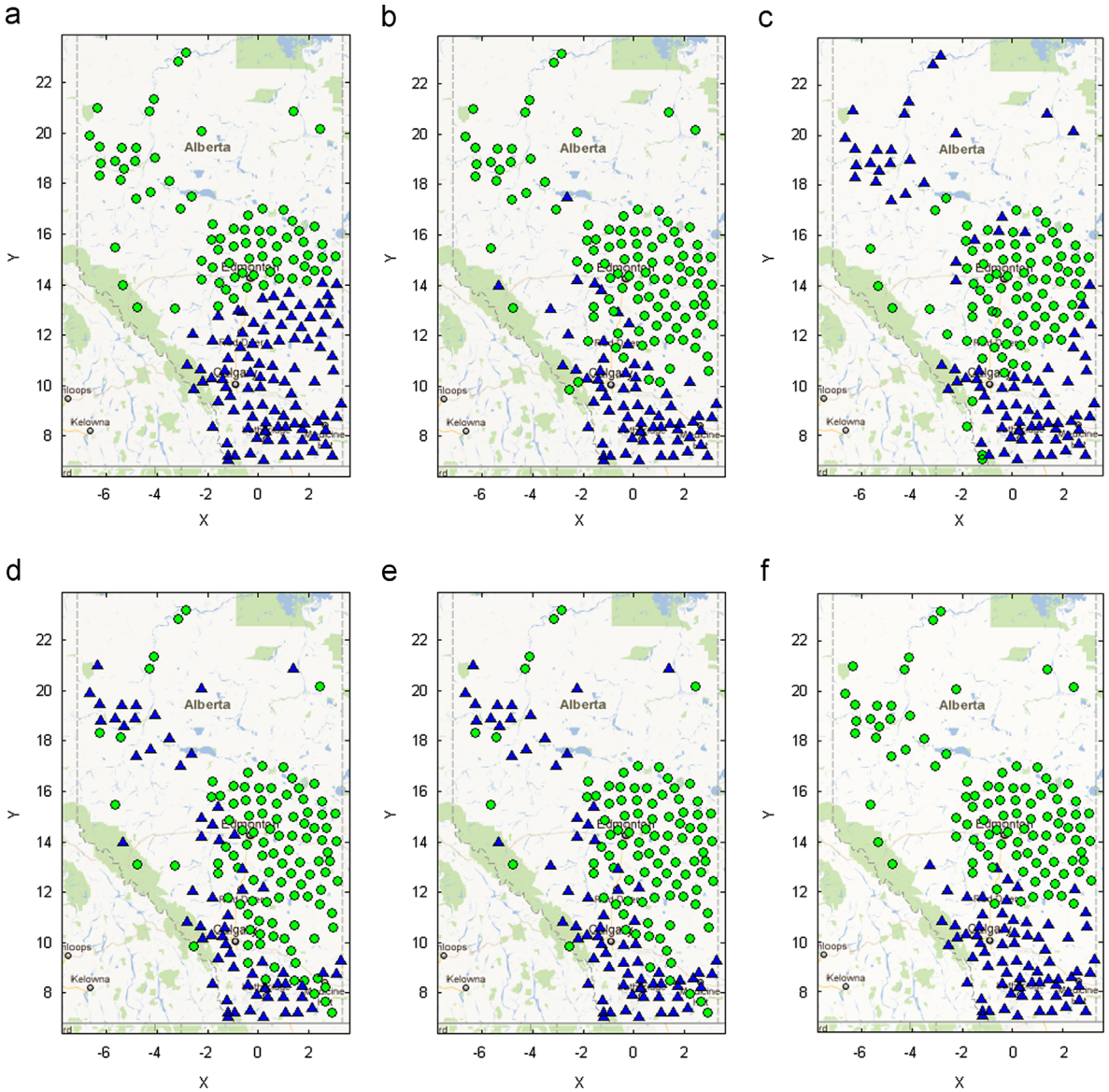


Fig. 16. Revealed clusters for Alberta climate data for (a) spatial part of data, (b) temperature part, (c) precipitation part, (d) humidity part, (e) all parts and using FCM method and (f) all parts using the optimal weights. Number of clusters $c=2$ and DFT(24) representation has been used.

agreement achieved by FCM (Q_{FCM}), and the level of agreement achieved by optimal weights (Q). For the last one, the results are reported as the average and standard deviation in 40 independent runs.

As shown in this table, the proposed method can produce the structures with a higher level of agreement among all data sources. In most cases, increasing the number of clusters (granularity) increases the value of Q (which means reduces the level of agreement). In fact, by increasing the number of clusters, more details about the available structures in each separate data source is considered and as the result the level of agreement between structures in different data sources is decreased. On the other hand,

by increasing the length of time series representation, the clusters are built with higher level of agreement because by increasing the length of representation of time series, the degree of overlap between clusters and as a result the FCM objective function (that has been used in (7) as denominator) is increased and the value of Q is decreased. Furthermore, different parameters (e.g. number of clusters, type and length of representation, etc) have various impacts on the available structures in each data source and affect the level of agreements achieved by the proposed method over data sources.

Fig. 16(a)–(d) show the clusters for different data sources separately, Fig. 16(e) shows the clusters reveled by FCM over all data sources, and Fig. 16(f) shows the clusters produced by the proposed method (optimal weights) over all data sources. We used two clusters $c=2$ and the DFT(24) representation of the time series.

Moreover Fig. 17(a)–(d) show the clusters for different data sources using PAA(24) representation and number of clusters $c=3$, Fig. 17(e) shows the clusters reveled by FCM over all data sources, and Fig. 17(f) shows the clusters produced by the proposed method over all data sources.

In fact, in both Figs. 16(f) and 17(f) the revealed clusters are the ones that have the highest agreement with the available structures in distinct data sources.

4.4. Comparative studies

In this sub-section, we compare the agreement-based clustering method with a feature weighting approach proposed in [28], called Fuzzy Clustering with Weighting of Data Variables (FCWDV). In this technique, a distance function between data point $\mathbf{x}_k$ and cluster center $\mathbf{v}_i$ is defined as

$$d^2(\mathbf{x}_k, \mathbf{v}_i) = \sum_{s=1}^{r} \alpha_{is}^t (x_{ks} - v_{is})^2 \tag{12}$$

where, x_{ks} indicates the sth variable (feature) of $\mathbf{x}_k$, r is the dimensionality of data and α_{is} is a weight indicating the influence of feature s on ith cluster with the following constraint:

$$\sum_{s=1}^{r} \alpha_{is} = 1, \text{ for } i = 1, 2, \ldots, c. \tag{13}$$

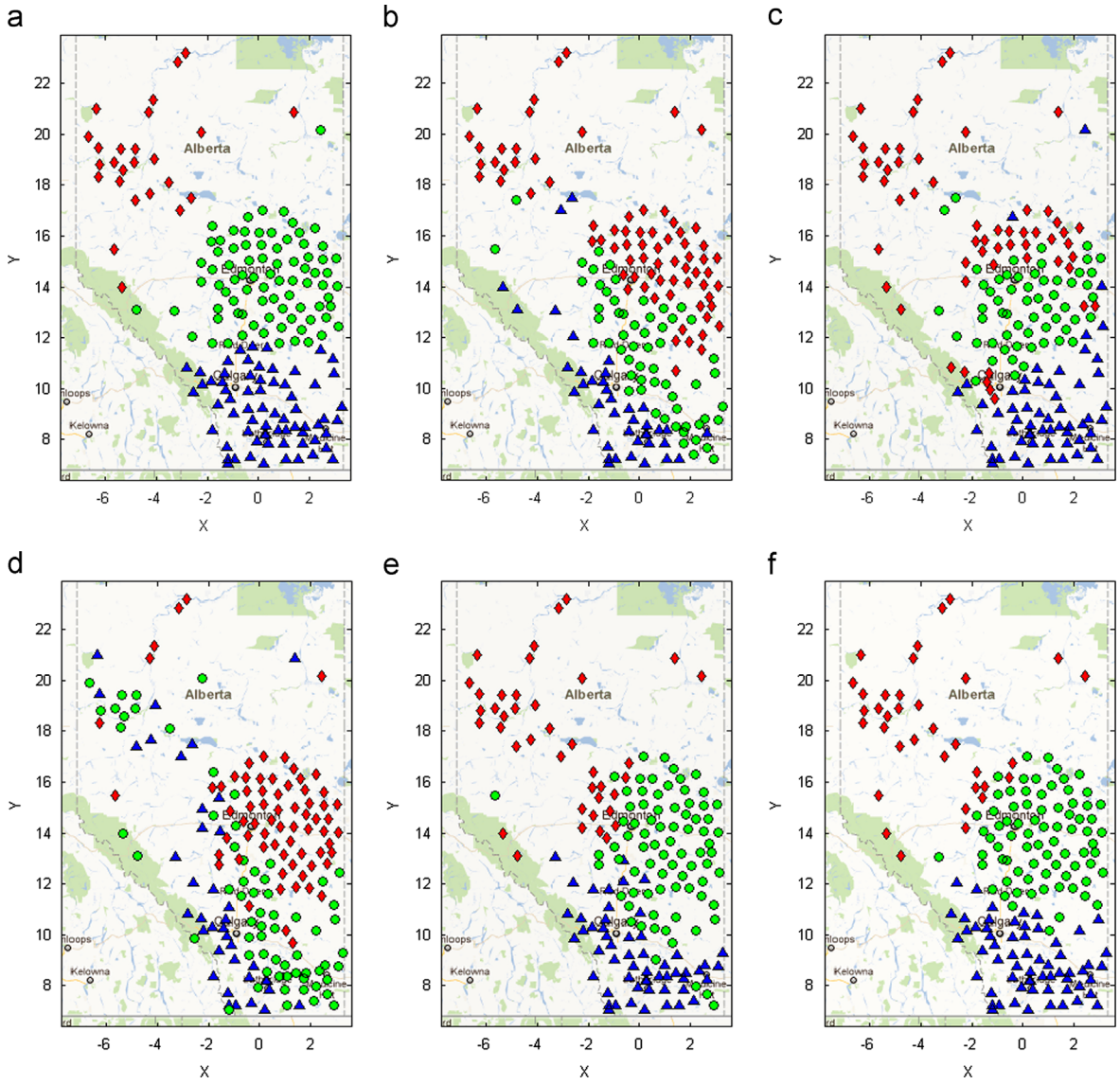


Fig. 17. Revealed clusters for Alberta climate data (a) spatial part of data, (b) temperature part, (c) precipitation part, (d) humidity part, (e) all parts and using FCM method and (f) all parts using the optimal weights. Number of clusters $c=3$ and PAA(24) representation has been used.

Table 3
Comparison between the proposed approach and FCWDV over different data sets and number of clusters $c=2$, 3, 4, and 5. Results are reported in form of average and standard deviation of ANMI for 10 independent runs.

Data set	Method	$c=2$	$c=3$	$c=4$	$c=5$
Synthetic	FCWDV	0.175 ± 0.048	0.241 ± 0.026	0.323 ± 0.024	0.396 ± 0.032
	Proposed	0.249 ± 0.006*	0.238 ± 0.012	0.357 ± 0.033*	0.393 ± 0.04
Iris	FCWDV	0.925 ± 0	0.785 ± 0*	0.724 ± 0.037*	0.655 ± 0.053
	Proposed	0.925 ± 0	0.768 ± 0.002	0.694 ± 0.001	0.689 ± 0.002*
Housing	FCWDV	0.321 ± 0	0.343 ± 0	0.352 ± 0	0.399 ± 0.001
	Proposed	0.375 ± 0*	0.444 ± 0.002*	0.452 ± 0.005*	0.401 ± 0.003
Fire	FCWDV	0.183 ± 0	0.147 ± 0.001	0.174 ± 0.009	0.179 ± 0.005
	Proposed	0.291 ± 0.001*	0.253 ± 0.013*	0.256 ± 0.004*	0.280 ± 0.003*
Climate DFT(24)	FCWDV	0.284 ± 0.002	0.460 ± 0	0.536 ± 0.002	0.525 ± 0.005*
	Proposed	0.307 ± 0.002*	0.467 ± 0.001*	0.545 ± 0.008*	0.518 ± 0.008
Climate PAA(24)	FCWDV	0.230 ± 0.001	0.413 ± 0.031	0.456 ± 0.011	0.448 ± 0.013
	Proposed	0.311 ± 0.002*	0.488 ± 0.005*	0.502 ± 0.008*	0.498 ± 0.005*

Entries marked with asterisk indicate that the difference between the achieved result for that method is statistically significant (with 95% confidence) in comparison with the result produced by another method.

The objective function in this technique is the FCM objective function and can be minimized by calculating cluster centers, weight matrix, and partition matrix in an iterative fashion.

For a consistent comparison with the proposed agreement-based clustering in this paper, we adopted FCWDV to assign a weight to each block of features, instead of assigning a weight to each single feature. An Average Normalized Mutual Information (ANMI) [5] is considered as the evaluation criterion. Let us consider C_1 and C_2 as two separate results of clustering with c clusters over data sources $D[1]$ and $D[2]$. The normalized mutual information (NMI) between C_1 and C_2 is defined as

$$\phi(C_1,\ C_2)=\frac{\sum_{i=1}^{c}\sum_{j=1}^{c} n_{ij}(\log)\ n\times n_{ij}/n_i\times n_j}{\sqrt{(\sum_{i=1}^{c} n_i\ \log\ n_i/n)\ (\sum_{j=1}^{c} n_j\ \log\ n_j/n)}} \tag{14}$$

where, n is number of data objects, n_i is number of objects in ith cluster of C_1, n_j is number of objects in jth cluster of C_2, and n_{ij} is number of objects in ith cluster of C_1 and jth cluster of C_2. Assuming $\hat{C}$ as the final clustering result, the ANMI criterion is defined as follows

$$\overline{\phi}(\hat{C}, C_{1,2,\ldots,p})=\frac{1}{p}\sum_{i=1}^{p} NMI(\hat{C},\ C_i). \tag{15}$$

$\overline{\phi}$ in (13) is always in range [0, 1], and a higher value of this criteria is desired. In this comparison, $\hat{C}$ stands for the resulting clusters obtained when using the agreement-base method or FCWDV and C_1, C_2, … C_p are the clusters obtained when completing fuzzy clustering of each separate data source $D[1]$, $D[2]$,…, $D[p]$, respectively.

Table 3 shows the results obtained over different data sets studied earlier in this paper for the number of clusters running from 2 to 5; the results are reported in terms of average and standard deviation of ANMI produced for 10 independent runs.

The t-test with $\alpha=0.05$ (95% confidence) has been applied and entries marked with asterisk (*) positioned in each cell indicate that the difference between the achieved result for that method (with the specified number of clusters) is statistically significant in comparison with the result produced by another method. As shown in this table, in most entries the proposed approach achieves results that are significantly better than those produced by the FCWDV method.

5. Conclusions

In this paper, we have proposed a fuzzy clustering approach to deal with data with blocks of features coming from different sources. A distance function has been proposed to control the effect of each source in the clustering process and the FCM objective function has been adopted to cope with the new distance function. An evaluation criterion is introduced and a particle swarm optimization is employed to find the optimal weights embedded in the new distance function. The proposed method has been studied over synthetic and real data sets with different characteristics. Experimental results show that the introduced method reveals interesting structures from data with blocks of features coming from distinct sources.

Acknowledgments

Support from Alberta Innovates—Technology Futures and Alberta Advanced Education & Technology, Natural Sciences and Engineering Research Council of Canada, and the Canada Research Chair Program is gratefully acknowledged.

Appendix A

By inserting the proposed distance function (3) into the original FCM objective function we have:

$$J=\sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{m}(\lambda_1\ \|\mathbf{v}_i(1)-\mathbf{x}_k(1)\|^2+\cdots+\lambda_p\ \|\mathbf{v}_i(p)-\mathbf{x}_k(p)\|^2) \tag{A1}$$

To calculate the membership degrees we define the augmented objective function where the constraints are handled by Lagrange multiplier, γ, for patterns $q=1,2,\ldots,n$:

$$L=\sum_{i=1}^{c} u_{iq}^{m}(\lambda_1\ \|\mathbf{v}_i(1)-\mathbf{x}_q(1)\|^2+\cdots+\lambda_p\ \|\mathbf{v}_i(p)-\mathbf{x}_q(p)\|^2)-\gamma(\sum_{i=1}^{c} u_{iq}-1) \tag{A2}$$

We have

$$\frac{\partial L}{\partial u_{rq}}=m\ u_{rq}^{m-1}(\lambda_1\ \|\mathbf{v}_r(1)-\mathbf{x}_q(1)\|^2+\cdots+\lambda_p\ \|\mathbf{v}_r(p)-\mathbf{x}_q(p)\|^2)-\gamma=0 \tag{A3}$$

From (A3) we have:

$$\gamma^{1/(m-1)}=u_{rq}(m\ (\lambda_1\ \|\mathbf{v}_r(1)-\mathbf{x}_q(1)\|^2+\cdots+\lambda_p\ \|\mathbf{v}_r(p)-\mathbf{x}_q(p)\|^2))^{1/(m-1)} \tag{A4}$$

Since in FCM we have $\sum_{j=1}^{c} u_{jq}=1$, we get

$$\sum_{j=1}^{c}\frac{\gamma^{1/(m-1)}}{(m\ (\lambda_1\ \|\mathbf{v}_j(1)-\mathbf{x}_q(1)\|^2+\cdots+\lambda_p\ \|\mathbf{v}_j(p)-\mathbf{x}_q(p)\|^2))^{1/(m-1)}}=1 \tag{A5}$$

and

$$\sum_{j=1}^{c}\frac{u_{rq}(m\,(\lambda_1\,\|\mathbf{v}_r(1)-\mathbf{x}_q(1)\|^2+\cdots+\lambda_p\,\|\mathbf{v}_r(p)-\mathbf{x}_q(p)\|^2))^{1/(m-1)}}{(m\,(\lambda_1\,\|\mathbf{v}_j(1)-\mathbf{x}_q(1)\|^2+\cdots+\lambda_p\,\|\mathbf{v}_j(p)-\mathbf{x}_q(p)\|^2))^{1/(m-1)}}=1 \tag{A6}$$

From (A6) we have

$$u_{rq}=\frac{1}{\sum_{j=1}^{c}((\lambda_1\,\|\mathbf{v}_r(1)-\mathbf{x}_q(1)\|^2+\cdots+\lambda_p\,\|\mathbf{v}_r(p)-\mathbf{x}_q(p)\|^2)/(\lambda_1\,\|\mathbf{v}_j(1)-\mathbf{x}_q(1)\|^2+\cdots+\lambda_p\,\|\mathbf{v}_j(p)-\mathbf{x}_q(p)\|^2))^{1/(m-1)}} \tag{A7}$$

And finally

$$u_{rq}=\frac{1}{\sum_{j=1}^{c}(d^2_{\lambda_1,\ldots,p}(\mathbf{v}_r,\mathbf{x}_q)/d^2_{\lambda_1,\ldots,p}(\mathbf{v}_j,\mathbf{x}_q))^{1/(m-1)}}. \tag{A8}$$

To calculate the prototypes we split (A1) into p objective functions $J_1, J_2, \ldots, J_p$ as follows:

$$\begin{aligned}
J_1&=\sum_{i=1}^{c}\sum_{k=1}^{n}u_{ik}^m\lambda_1\,\|\mathbf{v}_i(1)-\mathbf{x}_k(1)\|^2,\\
J_2&=\sum_{i=1}^{c}\sum_{k=1}^{n}u_{ik}^m\lambda_2\,\|\mathbf{v}_i(2)-\mathbf{x}_k(2)\|^2\\
&\vdots\\
J_p&=\sum_{i=1}^{c}\sum_{k=1}^{n}u_{ik}^m\lambda_p\,\|\mathbf{v}_i(p)-\mathbf{x}_k(p)\|^2
\end{aligned} \tag{A9}$$

The minimization of $J_1, J_2, \ldots, J_p$ leads to the minimization of (A1). To determine $\mathbf{v}_r(1)$ coming from J_1 we have:

$$\frac{\partial J_1}{\partial \mathbf{v}_r(1)}=2\lambda_1\sum_{k=1}^{n}u_{rk}^m(\mathbf{v}_r(1)-\mathbf{x}_k(1))=0 \tag{A10}$$

Finally we obtain

$$\mathbf{v}_r(1)=\frac{\sum_{k=1}^{n}u_{rk}^m\mathbf{x}_k(1)}{\sum_{k=1}^{n}u_{rk}^m} \tag{A11}$$

In the same manner, the prototypes corresponding to $J_2, \ldots, J_p$ can be computed.

As $\mathbf{v}_r=[\mathbf{v}_r(1)|\mathbf{v}_r(2)|\cdots|\mathbf{v}_r(p)]$, we have:

$$\mathbf{v}_r=\left(\frac{\sum_{k=1}^{n}u_{rk}^m\mathbf{x}_k(1)}{\sum_{k=1}^{n}u_{rk}^m}\middle|\frac{\sum_{k=1}^{n}u_{rk}^m\mathbf{x}_k(2)}{\sum_{k=1}^{n}u_{rk}^m}\middle|\ldots\middle|\frac{\sum_{k=1}^{n}u_{rk}^m\mathbf{x}_k(p)}{\sum_{k=1}^{n}u_{rk}^m}\right)=\frac{\sum_{k=1}^{n}u_{rk}^m\mathbf{x}_k}{\sum_{k=1}^{n}u_{rk}^m}. \tag{A12}$$

References

[1] J.C. Dunn, A fuzzy relative of the ISODATA process, and its use in detecting compact well-separated clusters, Journal of Cybernetics 3 (3) (1973) 32–57.

[2] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, E. Keogh, Querying and mining of time series data: experimental comparison of representations and distance measures, in: Proceedings of VLDB Endowment, Auckland, New Zealand, 2008, pp. 1542–1552.

[3] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, Fast subsequence matching in time-series databases, in: Proceedings of the ACM SIGMOD International Conference On Management of Data, 1994, pp. 419–429.

[4] E. Keogh, K. Chakrabarti, M. Pazzani, S. Mehrotra, Dimensionality reduction for fast similarity search in large time series databases, Knowledge and Information Systems 3 (3) (2001) 263–286.

[5] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, Journal of Machine Learning Research 3 (2002) 583–617.

[6] W. Pedrycz, K. Hirota, Forming consensus in the networks of knowledge, Engineering Applications of Artificial Intelligence 20 (2007) 657–666.

[7] W. Pedrycz, V. Loia, S. Senatore, P-FCM: a proximity-based fuzzy clustering, Fuzzy Sets and Systems 148 (2004) 21–41.

[8] K. Punera, J. Ghosh, Consensus based ensembles of soft clusterings, Applied Artificial Intelligence 22 (7) (2008) 780–810.

[9] W. Pedrycz, A. Bargiela, Fuzzy clustering with semantically distinct families of variables: descriptive and predictive aspects, Pattern Recognition Letters 31 (2010) 1952–1958.

[10] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948, 1995.

[11] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Functions, Plenum, New York, 1981.

[12] W. Pedrycz, K. Hirota, A consensus-driven fuzzy clustering, Pattern Recognition Letters 29 (2008) 1333–1343.

[13] X.Z. Fern, C.E. Brodley, Solving cluster ensemble problems by bipartite graph partitioning, in: 21st International Conference on Machine Learning, Banff, Canada, 2004.

[14] A.L.N. Fred, A.K. Jain, Combining multiple clusterings using evidence accumulation, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (6) (2005) 835–850.

[15] H. Ayad, M.S. Kamel, Cumulative voting consensus method for partitions with variable number of clusters, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (1) (2008) 160–173.

[16] N. Ilc, A. Dobnikar, Generation of a clustering ensemble based on a gravitational self-organising map, Neurocomputing 96 (2012) 47–56.

[17] P. Hore, L.O. Hall, D.B. Goldgof, A scalable framework for cluster ensembles, Pattern Recognition 42 (2009) 676–688.

[18] H.G. Ayad, M.S. Kamel, On voting-based consensus of cluster ensembles, Pattern Recognition 43 (2010) 1943–1953.

[19] S. Vega-Pons, J. Correa-Morris, J. Ruiz-Shulcloper, Weighted partition consensus via kernels, Pattern Recognition 43 (2010) 2712–2724.

[20] F. Wang, C. Yang, Z. Lin, Y. Li, Y. Yuan, Hybrid sampling on mutual information entropy-based clustering ensembles for optimizations, Neurocomputing 73 (2010) 1457–1464.

[21] A.L.V. Coelho, E. Fernandes, K. Faceli, Inducing multi-objective clustering ensembles with genetic programming, Neurocomputing 74 (2010) 494–498.

[22] V. Loia, W. Pedrycz, S. Senatore, Semantic Web Content Analysis: a study in proximity-based collaborative clustering, IEEE Transactions on Fuzzy Systems 15 (6) (2007) 1294–1312.

[23] W. Pedrycz, Collaborative fuzzy clustering, Pattern Recognition Letters 23 (2002) 1675–1686.

[24] S. Das, A. Abraham, A. Konar, Automatic kernel clustering with multi-elitist particle swarm optimization algorithm, Pattern Recognition Letters 29 (2008) 688–699.

[25] L.F.S. Coletta, L. Vendramin, E.R. Hruschka, R.J.G.B. Campello, W. Pedrycz, Collaborative fuzzy clustering algorithms: some refinements and design guidelines, IEEE Transactions on Fuzzy Systems 20 (3) (2012) 444–462.

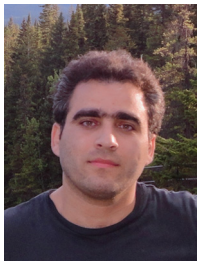
[26] H. Izakian, W. Pedrycz, I. Jamal, Clustering spatio-temporal data: an augmented fuzzy C-means, IEEE Transactions on Fuzzy Systems 25 (5) (2013) 855–868.

[27] X. Wang, Y. Wang, L. Wang, Improving fuzzy c-means clustering based on feature-weight learning, Pattern Recognition Letters 25 (2004) 1123–1132.

[28] A. Keller, F. Klawonn, Fuzzy clustering with weighting of data variables, International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems 8 (2000) 735–746.

[29] W. Pedrycz, P. Raia, Collaborative clustering with the use of fuzzy C-means and its quantification, Fuzzy Sets and Systems 159 (2008) 2399–2427.

[30] S. Das, A. Abraham, A. Konar, Automatic clustering using an improved differential evolution algorithm, IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans 38 (1) (2008) 218–237.



Hesam Izakian received the M.S. degree in Computer Engineering from the University of Isfahan, Iran. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is working under the supervision of Professor Witold Pedrycz and his research interests include Computational Intelligence, knowledge discovery and data mining, pattern recognition, and Software Engineering.

Witold Pedrycz is Professor and Canada Research Chair (CRC—Computational Intelligence) in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. He is also with the Systems Research Institute of the Polish Academy of Sciences, Warsaw, Poland. He also holds an appointment of special professorship in the School of Computer Science, University of Nottingham, UK. In 2009 Dr. Pedrycz was elected a foreign member of the Polish Academy of Sciences. In 2012 he was elected a Fellow of the Royal Society of Canada. Witold Pedrycz has been a member of numerous program committees of IEEE conferences in the area of fuzzy sets and neurocomputing. In 2007 he received a prestigious Norbert Wiener award from the IEEE Systems, Man, and Cybernetics Council and in 2013 a Killam Prize. He is a recipient of the IEEE Canada Computer Engineering Medal 2008. In 2009 he has received a Cajastur Prize for Soft Computing from the European Centre for Soft Computing for "*pioneering and multifaceted contributions to Granular Computing*".

His main research directions involve Computational Intelligence, fuzzy modeling and Granular Computing, knowledge discovery and data mining, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and Software Engineering. He has published numerous papers in this area. He is also an author of 15 research monographs covering various aspects of Computational Intelligence and Software Engineering.

Dr. Pedrycz is intensively involved in editorial activities. He is an Editor-in-Chief of ***Information Sciences*** and Editor-in-Chief of ***IEEE Transactions on Systems, Man, and Cybernetics—Systems***. He currently serves as an Associate Editor of ***IEEE Transactions on Fuzzy Systems*** and is a member of a number of editorial boards of other international journals.