Feature-Weighted Possibilistic C-Means Clustering with a Feature-Reduction Framework

Miin-Shen Yang and Josephine B.M. Benjamin

***Abstract*—In 1993, Krishnapuram and Keller proposed possibilistic c-means (PCM) clustering where the PCM had various extensions in the literature. However, the PCM algorithm with its extensions treats data points under equal importance for features. In real applications, different features in a data set should take different importance with different weights. In this paper, we first propose a feature-weighted PCM (FW-PCM). We then construct a feature-reduction framework. Therefore, we give a feature-weighted possibilistic c-means clustering with a feature-reduction framework, termed as a feature-weighted reduction PCM (FW-R-PCM) algorithm. The proposed FW-R-PCM can improve clustering performance of PCM by calculating feature weights to identify important features, and so it can consequently eliminate these irrelevant features to reduce feature dimension. Its theoretical behavior and computational complexity are also analyzed. The effectiveness and usefulness of FW-R-PCM are demonstrated through experimental results using synthetic and real data sets, where comparisons of FW-R-PCM with PCM, FW-PCM and some existing feature-weighted clustering algorithms are also made.**

***Index Terms*—Clustering; Possibilistic c-means (PCM), Feature weights; Feature-reduction schema; Feature-weighted PCM (FW-PCM); Feature-weighted reduction PCM (FW-R-PCM).**

I. INTRODUCTION

CLUSTERING is an unsupervised learning in pattern recognition and machine learning. It is a method for finding clusters in a data set characterized by the greatest similarity within the same cluster and the greatest dissimilarity between different clusters to explore hidden patterns/structure within the data set. In clustering, partitional methods are the most popular in which the k-means is a traditional clustering algorithm with applications in various areas [1-5]. Fuzzy sets by Zadeh [6] introduced the idea of partial membership described by membership functions. Fuzzy clustering with fuzzy c-partitions using fuzzy set concept was first proposed by Ruspini [7]. The most used fuzzy clustering method is the fuzzy c-means (FCM) algorithm that was proposed by Dunn [8] and Bezdek [9]. Although the FCM algorithm has been successfully applied in various applications [10-13], there exist some drawbacks. For example, these FCM memberships do not always correspond to the degrees of belonging of data points to clusters, and it may have trouble in a noisy environment. These drawbacks of FCM were improved by Krishnapuram and Keller [14] by creating a possibilistic approach to clustering, called a possibilistic c-means (PCM) algorithm. PCM uses a possibilistic type of membership functions to describe the degree of belongingness of each point to each cluster.

In general, the PCM method gives results that are more robust to noise and outliers than FCM. However, in real applications, there may exist irrelevant features in data sets, and they might degrade the performance of clustering algorithms. By identifying and removing these irrelevant features using clustering algorithms, it can improve accuracy and computational time and also reduce feature dimensions. Recently, Ruspini et al. [13] gave a historical perspective for fuzzy clustering in which they had also given some highlights of PCM and properties. In Ruspini et al. [13], they did not mention any feature-weighted PCM. However, considering feature weights in PCM should be important and useful. In fact, feature-weighted techniques were widely used in clustering [15], such as the k-means and fuzzy c-means (FCM) clustering algorithms. Various feature-weighted k-means and FCM had been proposed in the literature, for example, feature-weighted k-means [16-18], and feature-weighted FCM [19-23]. As we know, there are less researches that consider feature-weighted PCM in the literatures. In this paper, we focus on feature weighting behaviors about PCM.

The PCM algorithm and its extensions always consider all features of data to be of equal relevance. Assigning equal weights will give the same influence to all features in clustering algorithms. In some instances, unequal feature weights are given to refine the relative influence if it is suspected that certain features are more relevant than others in clustering. Weight values are usually given within the interval [0,1] which sum to 1. Features with higher weights will have greater relevance as compared to those with smaller weights. By assigning different feature weights, it is able to identify the relative importance of each feature. This may be used to reduce these redundant features and will improve the comprehensibility of clustering algorithms. In this paper, we first propose a PCM algorithm with feature weights, called a feature-weighted PCM (FW-PCM), which can automatically compute feature weights for different features. We then construct a feature-reduction framework for FW-PCM and propose a feature-weighted reduction PCM (FW-R-PCM) that can identify irrelevant features with lower feature weights and discarding these irrelevant features. The proposed FW-R-PCM algorithm can reduce feature dimension and also decrease computational time with better clustering performance. The theoretical behavior and computational complexity are also analyzed. The effectiveness and usefulness of the proposed FW-R-PCM algorithm are

This work was supported in part by the Ministry of Science and technology (MOST) of Taiwan under Grant MOST-107-2118-M-033-002-MY2. Miin-Shen Yang is with Department of Applied Mathematics, Chung Yuan Christian University, Chung-Li 32023, Taiwan, where Miin-Shen Yang is the Corresponding Author, e-mail: msyang@math.cycu.edu.tw. Josephine B.M. Benjamin is with Department of Applied Mathematics, Chung Yuan Christian University, Chung-Li 32023, Taiwan, and Department of Mathematics, University of Santo Tomas, Manila, Phillipines.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.2968879, IEEE Transactions on Fuzzy Systems

demonstrated through experimental results using synthetic and real data sets, where comparisons of the proposed FW-R-PCM algorithm with PCM, FW-PCM and several existing feature-weighted clustering algorithms are made. The good aspects of the proposed FW-R-PCM will be demonstrated using experimental results and comparisons.

The remainder of this paper is organized as follows. In Section II, we first review some related works with the PCM, and weighted PCM algorithms. In Sections III, we first propose a feature-weighted PCM (FW-PCM) algorithm. We also demonstrate these good behaviors of the proposed FW-PCM. We then construct a feature-reduction framework for the FW-PCM. We propose a novel PCM algorithm with a feature-reduction framework, called feature-weighted reduction PCM (FW-R-PCM). To evaluate the performance of FW-R-PCM, we use numerical and real data sets to compare FW-R-PCM with PCM, FW-PCM and several feature-weighted k-means and feature-weighted FCM in Section IV. Experimental results using synthetic and real data sets actually demonstrate the effectiveness and usefulness of the proposed FW-R-PCM algorithm. Finally, conclusions are stated in Section V.

II. RELATED WORKS

In this section, some related works proposed in the literature are reviewed. Let $X=\{x_1,x_2,\cdots,x_n\}$ be a data set of *n* data points in a *d*-dimensional Euclidean space $\mathbb{R}^d$ with the i^{th} data point x_i that is a column vector. Let $U=[\mu_{ki}]_{c\times n}\in M_{hcn}$ be a $c\times n$ partition matrix with $M_{hcn}=\left\{U=[\mu_{ki}]_{cn}\middle|\forall k,i,\mu_{ki}\in\{0,1\},\sum_{k=1}^{c}\mu_{ki}=1\ \forall i\right\}$ where μ_{ki} is a hard membership of the i^{th} point in the k^{th} cluster. Based on fuzzy concept, $\mu_{ki}\in\{0,1\}$ can be extended to $\mu_{ki}\in[0,1]$. Let $U=[\mu_{ki}]_{c\times n}\in M_{fcn}$ be a $c\times n$ partition matrix with $M_{fcn}=\left\{U=[\mu_{ki}]_{cn}\middle|\forall k,i,\mu_{ki}\in[0,1],\sum_{k=1}^{c}\mu_{ki}=1\ \forall i\right\}$ where μ_{ki} is a fuzzy membership of the i^{th} point in the k^{th} cluster. Let $V=\{v_1,\cdots,v_c\}$ be the set of *c* cluster centers with $v_k^T=[v_{kj}]_{1\times d}$ where v_{kj} is the j^{th} feature of the k^{th} cluster center. Let $\mathrm{w}^T=[w_j]_{1\times d}$ be the weight vector with w_j as a feature weight of the j^{th} feature. The fuzzy *c*-means (FCM) algorithm with a partition matrix $U=[\mu_{ki}]_{c\times n}\in M_{fcn}$ is an iterative algorithm using the necessary conditions for minimizing the FCM objective function J_{FCM} with $J_{FCM}(U,V)=\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ki}^m\|x_i-v_k\|^2$ where $m>1$ is the degree of fuzziness [9-10]. Although FCM and its extensions are useful in clustering, their memberships do not well present the membership degrees of data points to classes in a noisy environment.

To improve this weakness of FCM, Krishnapuram and Keller [14] relaxed the constraint $\sum_{k=1}^{c}\mu_{ki}=1$ in FCM with a possibilistic membership matrix $U=[\mu_{ki}]_{c\times n}\in M_{pcn}$ with $M_{pcn}=\left\{U=[\mu_{ki}]_{cn}\middle|\forall k,i,\mu_{ki}\in[0,1],0<\sum_{i=1}^{n}\mu_{ki}<n\right\}$ where μ_{ki} is a possibilistic membership of the i^{th} point in the k^{th} cluster. Krishnapuram and Keller [14] proposed the possibilistic *c*-means (PCM) objective function as follows:

$$J_{PCM}(U,V)=\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ki}^m\|x_i-v_k\|^2+\sum_{k=1}^{c}\eta_k\sum_{i=1}^{n}(1-\mu_{ki})^m$$

Thus, the PCM clustering with a possibilistic partition matrix $U=[\mu_{ki}]_{c\times n}\in M_{pcn}$ is an iterative algorithm using the necessary conditions for minimizing the PCM objective function $J_{PCM}(U,V)$ with $\mu_{ki}=1\Big/\left(1+\left(\|x_i-v_k\|^2/\eta_i\right)^{\frac{1}{m-1}}\right)$ and $v_k=\sum_{i=1}^{n}\mu_{ki}^m x_i\Big/\sum_{i=1}^{n}\mu_{ki}^m$, $1\le i\le n$, $1\le k\le c$, where the parameter η_k may be a user-defined constant, or it can be chosen as the average fuzzy intracluster distance of cluster *i* or the average α-cut intracluster distance of cluster *i* with $\eta_k=K\left(\sum_{i=1}^{n}\mu_{ki}^m\|x_i-v_k\|^2\Big/\sum_{i=1}^{n}\mu_{ki}^m\right)$ or $\eta_k=\dfrac{\sum_{x_i\in(\pi_k)_\alpha}\|x_i-v_k\|^2}{|x_i\in(\pi_k)_\alpha|}$, where the constant $K>0$. The results of the PCM algorithm are sensitive to initializations and parameter selection. On the other hand, PCM clustering results may have a tendency to produce coincident clusters if a larger initial number c of clusters is given [24-26].

In the literature, various feature-weighted k-means and feature-weighted FCM had been proposed [16-23]. In this section, we review some feature-weighted clustering algorithms for comparisons with our proposed FW-R-PCM algorithm. In Huang et al. [16], they proposed the (feature) weighted k-means (W-KM). To extend the W-KM to subspace clustering using an entropy approach, Jing et al. [17] proposed entropy weighted k-means (EW-KM) by adding a weight entropy so that it can simultaneously minimize the within cluster dispersion and maximize the weight entropy. The EW-KM determines subsets of important dimensions in each cluster. The EW-KM objective function is

$$J_{EWKM}(U,V)=\sum_{k=1}^{c}\sum_{i=1}^{n}\sum_{j=1}^{d}\mu_{ki}w_{kj}\left(x_{ij}-v_{kj}\right)^2+\gamma\sum_{k=1}^{c}\sum_{i=1}^{n}w_{kj}\log w_{kj}$$

where $U=[\mu_{ki}]_{c\times n}\in M_{hcn}$, and $\gamma\ge 0$ is a parameter to control the size of feature weights.

Hung et al. [27] proposed a new weighted FCM (NW-FCM) algorithm to improve the performance of FCM and feature-weighted c-means models. NW-FCM is proposed to solve high-dimensional multi-class pattern recognition problems. Its objective function is given as $J_{NW-FCM}(U,V,W)=\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ki}^m\|x_j-M_{ij}\|^2$ where $U=[\mu_{ki}]_{c\times n}\in M_{fcn}$

and $M_{ij}=\sum_{\substack{i=1\\ i\neq j}}^{n}(\|v_k-x_i\|^{-1}\mu_{ki}x_i \Big/ \sum_{\substack{t=1\\ t\neq i}}^{n}\|v_k-x_t\|^{-1}\mu_{kt})$. Furthermore, Pimentel and de Souza [28] proposed a weighted multivariate fuzzy c-means membership (WM-FCM-M) method in which each object, cluster and feature has a suitable weight. It aims to identify the relevance of a feature when computing the membership degree for an object regarding a cluster. Feature weights in WM-FCM-M use the membership degrees and distances between a given object and each prototype in order to measure the relevance of the feature to final membership degree regarding a given cluster. We mention that weights in WM-FCM-M [27] are assigned on the objects, clusters and features while NW-FCM [28] includes the weighted mean in the objective function of FCM. Recently, Yang & Nataliani [23] proposed the feature-reduction fuzzy c-means (FRFCM) clustering algorithm by extending the EWKM objective function from $U=[\mu_{ki}]_{c\times n}\in M_{hcn}$ to $U=[\mu_{ki}]_{c\times n}\in M_{fcn}$ and also adding a parameter to control the weights of features. Yang & Nataliani [23] used FRFCM to eliminate irrelevant features with small weights.

Although there are various feature-weighted k-means and feature-weighted FCM in the literature [16-23, 27-28], there are only a few feature-weighted PCM. We next review these related works. Schneider [29] first proposed an algorithm called sample-weighted possibilistic c-means (SW-PCM), in which a set of membership weights are substituted into the PCM objective function. These weights relaxes a uniform bias found in PCM as a means of minimizing the effects of noisy data. Schneider [29] gave the following objective function

$$J_{SWPCM}(U,V)=\sum_{k=1}^{c}\sum_{i=1}^{n}(\mu_{ki})^m\|x_i-v_k\|^2+\sum_{k=1}^{c}\eta_k\sum_{i=1}^{n}(\varphi_i-\mu_{ki})^m$$

where $U=[\mu_{ki}]_{c\times n}\in M_{pcn}$ and $\varphi_i>0$ are weights. According to $J_{SWPCM}(U,V)$, Schneider [29] considered weights as sample weights, not feature weights. Schneider's replacement of the unity in $(1-\mu_{ki})^m$ constraint with sample weights will reduce the maximum membership values for outliers. Furthermore, Zhang and Chen [30] proposed a weighted possibilistic c-means (W-PCM) algorithm based on the SW-PCM objective function $J_{SWPCM}(U,V)$ to accelerate clustering speed of SW-PCM for clustering big data. More recently, Zhang et al. [31] proposed a secure SW-PCM algorithm based on the SW-PCM objective function $J_{SWPCM}(U,V)$ according to the BGV encryption scheme in which BGV is used to encrypt the raw data for the privacy preservation on cloud. For both Zhang and Chen [30] and Zhang et al. [31], they consider weights as sample weights, not feature weights, based on the Schneider's objective function $J_{SWPCM}(U,V)$.

Another weighted and constrained PCM clustering algorithm (WC-PCM) was proposed by Bahrampour et al. [32] for process fault detection and diagnosis (FDI) in offline and online modes for both known and novel faults. This algorithm incorporates simultaneously possibilistic clustering method and local attribute weighting for time series segmentation which allows different weights to be allocated to different features responsible for distinguishing process faults. Bahrampour et al. [32] proposed the objective function as applied to time series clustering as follows:

$$J_{WCPCM}(U,V,W,\psi,\Gamma)=\sum_{k=1}^{c}\sum_{i=1}^{n}(\mu_{ki})^m\frac{1}{A_k(t_i)}\sum_{j=1}^{d}(w_{kj})^q(x_{ij}-v_{kj})^2$$
$$+\sum_{k=1}^{c}\eta_k\sum_{i=1}^{n}(1-\mu_{ki})^m \text{ where } U=[\mu_{ki}]_{c\times n}\in M_{pcn},\quad w_{kj}\in[0,1]$$

and $\sum_{j=1}^{d}w_{kj}=1\quad\forall k,j$. Also, the parameter $A_k(t_i)=\exp(-1(t_i-\alpha_k)^2/\sigma_k^2)$ is the Gaussian membership function for time constraint, $\psi=[\alpha_1,\alpha_2,\cdots,\alpha_c]^T$ and $\Gamma=[\sigma_1,\sigma_2,\cdots,\sigma_c]^T$ are centers and variances of membership function, respectively, q is the discriminant exponent, and w_{kj} represents the relevance weight of the feature j in the k^{th} cluster. We mention that, according to our knowledge, the WC-PCM algorithm proposed by Bahrampour et al. [32] is the first work to consider feature weights for PCM that are $w_{kj}\in[0,1],\forall k,j,$ and $\sum_{j=1}^{d}w_{kj}=1,\forall k$. Recently, Hameed & Mohammed [33] proposed a spam filtering approach based on weighted version of possibilistic c-means, called WPCM-SF, which can efficiently distinguish between spam and legitimate email messages. The WPCM-SF objective function proposed by Hameed & Mohammed [33] as a spam filtering model is

$$J_{WPCM-SF}(U,V,W)=\sum_{k=1}^{c}\sum_{i=1}^{n}(\mu_{ki})^m d_w^2(x_i-v_k)+\sum_{k=1}^{c}\eta_k\sum_{i=1}^{n}(1-\mu_{ki})^m$$

where $U=[\mu_{ki}]_{c\times n}\in M_{pcn}$ and $d_w(x_i,v_k)=\sqrt{\sum_{j=1}^{d}(w_j)^2(x_{ij}-v_{kj})^2}$ is the weighted Euclidean distance and $W=[w_1,w_2,\cdots,w_d]$ is the feature weight vector. In this algorithm, weight assignments using information gain algorithm is used as initializations while initial cluster centers and typicality values are obtained using the PCM algorithm. These initializations were then utilized for spam filtering.

The SW-PCM and its extensions proposed by Schneider [29], Zhang and Chen [30] and Zhang et al. [31] consider assigning sample weights to data points. In the case of Schneider [29], SW-PCM tends to reduce the effects of outliers and noisy points in the clustering process while the W-PCM of Zhang and Chen [30] assigns low weight values to missing data points so that it reduces the destructive effect of these missing data points in the clustering process. On the other hand, WC-PCM by Bahrampouur et al. [32] and WPCM-SF by Hameed & Mohammed [33] are more specific to a particular problem application instead of a general case. Also, with WPCM-SF, weights are not updated during the clustering process but were given at the start generated by information gain algorithm. Although both WC-PCM and WPCM-SF have considered feature weights in PCM, they use these feature weights to help minimizing membership function

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.2968879, IEEE Transactions on Fuzzy Systems

values of data points, not completely focusing on feature-weighted PCM procedures. We next propose a novel feature-weighted PCM algorithm.

III. FEATURE-WEIGHTED POSSIBILISTIC C-MEANS WITH A FEATURE-REDUCTION FRAMEWORK

In the last section, we mention that there are less related works with feature weights for the PCM algorithm. Thus, we first propose a novel feature-weighted PCM algorithm and then construct a feature-reduction framework for the proposed feature-weighted PCM algorithm.

III-1 Feature-Weighted Possibilistic C-Means Clustering Algorithm

To extend the PCM algorithm with a feature-weighted schema, we consider feature weights $w_j, j=1,\cdots,d,$ and propose the following objective function:

$$J_{FW-PCM}(U,W,V)=\sum_{k=1}^{c}\sum_{i=1}^{n}\sum_{j=1}^{d}w_j\mu_{ki}^{m}\left(x_{ij}-v_{kj}\right)^2 +\lambda\sum_{k=1}^{c}\sum_{i=1}^{n}\sum_{j=1}^{d}w_j\left(1-\mu_{ki}\right)^{m}+\frac{n}{c}\sum_{j=1}^{d}w_j\log w_j \quad (1)$$

subject to $U=[\mu_{ki}]_{c\times n}\in M_{pcn}$ with $w_{kj}\in[0,1],\forall k, j,$ and $\sum_{j=1}^{d}w_{kj}=1,\forall k$. The clustering algorithm based on $J_{FW-PCM}(U,W,V)$, called the feature-weighted PCM (FW-PCM), is through the updating equations for minimizers of the objective function $J_{FW-PCM}(U,W,V)$. Thus, the FW-PCM algorithm allows different feature weights to be allocated to features of data in which each feature can define the amount of contribution of that feature in forming the cluster. In the FW-PCM objective function $J_{FW-PCM}(U,W,V)$, we define λ with $\lambda=\beta/(m^2c)$ where $\beta=\sum_{i=1}^{n}\|x_i-\bar{x}\|^2/n$ and $\bar{x}=\sum_{i=1}^{n}x_i/n$. The factor (n/c) in the entropy term of feature weights w_j is used to control the strength of clustering on features. The role of λ and (n/c) will be discussed later.

Theorem 1 The necessary conditions for minimizing the FW-PCM objective function $J_{FW-PCM}(U,W,V)$ are

$$\mu_{ki}=\left(1+\left(\lambda^{-1}\sum_{j=1}^{d}w_j\left(x_{ij}-v_{kj}\right)^2\right)^{\frac{1}{m-1}}\right)^{-1} \quad (2)$$

$$v_{kj}=\sum_{i=1}^{n}(\mu_{ki})^m x_{ij}\Big/\sum_{i=1}^{n}(\mu_{ki})^m \quad (3)$$

$$w_j=\frac{\exp\left(\frac{-c\left(\sum_{k=1}^{c}\sum_{i=1}^{n}(\mu_{ki})^m\left(x_{ij}-v_{kj}\right)^2+\lambda\sum_{k=1}^{c}\sum_{i=1}^{n}(1-\mu_{ki})^m\right)}{n}\right)}{\sum_{s=1}^{d}\exp\left(\frac{-c\left(\sum_{k=1}^{c}\sum_{i=1}^{n}(\mu_{ki})^m\left(x_{ij}-v_{kj}\right)^2+\lambda\sum_{k=1}^{c}\sum_{i=1}^{n}(1-\mu_{ki})^m\right)}{n}\right)} \quad (4)$$

Proof: We prove it using Langrange multipliers. The Lagrangian of $J_{FW-PCM}(U,W,V)$ is

$$J^{\mathrm{L}}_{FW-PCM}(U,W,V)=\sum_{k=1}^{c}\sum_{i=1}^{n}\sum_{j=1}^{d}w_j\mu_{ki}^{m}\left(x_{ij}-v_{kj}\right)^2 +\lambda\sum_{k=1}^{c}\sum_{i=1}^{n}(1-\mu_{ki})^m+\frac{n}{c}\sum_{j=1}^{d}w_j\log(w_j)-\gamma_1\left(\sum_{i=1}^{n}\mu_{ki}-1\right)-\gamma_2\left(\sum_{i=1}^{n}w_j-1\right),$$

where γ_1 and γ_2 are the Lagrange multipliers. First, we fix $V=V^*$ and $W=W^*$ in $J^{\mathrm{L}}_{FW-PCM}(U,W,V)$ and take the partial derivative of $J^{\mathrm{L}}_{FW-PCM}(U,W^*,V^*)$ with respect to μ_{ki} . We obtain

$$\frac{\partial J^{\mathrm{L}}_{FW-PCM}}{\partial u_{ki}}=m\sum_{j=1}^{d}w_j(\mu_{ki})^{m-1}\left(x_{ij}-v_{kj}\right)^2+m\lambda\sum_{j=1}^{d}w_j(1-\mu_{ki})^{m-1}+\gamma_1$$

Set it to be zero with more manipulation, we get equation (2). Next, we fix $U=U^*$ and $W=W^*$ in $J^{\mathrm{L}}_{FW-PCM}(U,W,V)$ and take the partial derivative of $J^{\mathrm{L}}_{FW-PCM}(U^*,W^*,V)$ with respect to v_{kj}, we get $\frac{\partial J^{\mathrm{L}}_{FW-PCM}}{\partial v_{kj}}=-2\sum_{i=1}^{n}w_j(\mu_{ki})^m\left(x_{ij}-v_{kj}\right)$. Set it as zero, we obtain equation (3), that is $v_{kj}=\sum_{i=1}^{n}(\mu_{ki})^m x_{ij}\Big/\sum_{i=1}^{n}(\mu_{ki})^m$. Similarly, we fix $U=U^*$ and $V=V^*$ in $J^{\mathrm{L}}_{FW-PCM}(U,W,V)$ and take the partial derivative of $J^{\mathrm{L}}_{FW-PCM}(U^*,W,V^*)$ with respect to w_j, we get $\frac{\partial J^{\mathrm{L}}_{FW-PCM}}{\partial w_j}=\sum_{k=1}^{c}\sum_{i=1}^{n}(\mu_{ki})^m\left(x_{ij}-v_{kj}\right)^2+\lambda\sum_{k=1}^{c}\sum_{i=1}^{n}(1-\mu_{ki})^m+\frac{n}{c}\left(\log(w_1)+1\right)+\gamma_2$ and also $\frac{\partial J^{\mathrm{L}}_{FW-PCM}}{\partial \gamma_2}=\sum_{i=1}^{n}w_j-1$. Set them to be zero with more manipulation, we get equation (4). Thus, the theorem is proven. ■

Thus, the proposed FW-PCM clustering algorithm can be summarized as follows:

FW-PCM Algorithm

Fix $\epsilon>0,\ 2\le c\le n$ where c is the number of clusters. Give initial cluster centers $V^{(0)}$ and feature weight $W^{(0)}=\left[w_j\right]_{1xd},\ 1\le j\le d$ with $\sum_{j=1}^{d}w_j=1$. Set $q=1$.

Step1: Calculate possibilistic membership $U^{(q)}$ by equation (2) using $V^{(q-1)}$ and $W^{(q-1)}$.

Step 2: Update cluster center $V^{(q)}$ by equation (3) using $U^{(q)}$.

Step 3: Update feature weight $W^{(q)}$ by equation (4) using $V^{(q)}$ and $U^{(q)}$

Step 4: IF $\left\|\|W^{(q)}\|-\|W^{(q-1)}\|\right\|<\epsilon$, STOP

ELSE, set $q=q+1$ and go back to Step 1.

We discuss the role of the constant parameter λ in the FW-PCM objective function. The parameter λ defined by $\lambda=\beta/(m^2c)$ is a scale parameter for the constraint term of possibilistic memberships $[\mu_{ki}]_{c\times n}\in M_{pcn}$. The defined λ is

similar to that of the second term in the PCA objective function proposed by Yang and Wu [34]. We can observe from Eq. (2) that the membership function value is dependent on the value of λ. If λ is low, μ_{ki} is small, and if λ is high, μ_{ki} is large. Thus, $\lim_{\lambda\to\infty}\mu_{ki}=1$. In the equation $\lambda=\beta/(m^2c)$, β is the sample variance to measure the degree of separation in the data set which acts as a normalization term, m is the fuzzifier in PCM, and c is the number of clusters. The fuzzifier m determines the rate of decrease of the membership value. In Eq. (2), when $m\to 1^+$, $\mu_{ki}=1$ if $\sum_{j=1}^{d} w_j\left(x_{ij}-v_{kj}\right)^2<\lambda$; $\mu_{ki}=1/2$ if $\sum_{j=1}^{d} w_j\left(x_{ij}-v_{kj}\right)^2=\lambda$; $\mu_{ki}=0$ if $\sum_{j=1}^{d} w_j\left(x_{ij}-v_{kj}\right)^2>\lambda$. The cluster number, c, also has influence on the membership values. Similar to that in PCA [34], c is used to control the steep degree of the membership functions. The above discussion justifies why we use $\lambda=\beta/(m^2c)$ as a multiplier in the second term of the FW-PCM objective function J_{FW-PCM}.

Although the feature weights produced by FW-PCM can measure the importance of features where smaller feature weights may indicate more irrelevant features, FW-PCM does not have feature-reduction behaviors to eliminate these exactly irrelevant features. Based on the FW-PCM objective function, we next construct a feature-reduction framework so that it can automatically eliminate these irrelevant features.

III-2 A Feature Reduction Framework

A problem known as the "curse of dimensionality" arises when one has to cluster a data set originating from a high dimensional space. This problem can be resolved through dimension reduction process that reduces the number of features which are considered to be irrelevant. Dimension reduction procedure can be a solution that supports scalable object retrieval and satisfies precision of query results (Xu & Wunsch [35]). Recently, Yang & Nataliani [23] proposed an algorithm called feature-reduction FCM (FRFCM) that can eliminate irrelevant features identified with small weights during the FCM clustering processes. Here, we borrow the idea of Yang & Nataliani [23] to construct a feature-reduction framework for the proposed FW-PCM algorithm.

Recall that the FW-PCM objective function is

$$J_{FW-PCM}(U,W,V)=\sum_{k=1}^{c}\sum_{i=1}^{n}\sum_{j=1}^{d} w_j\mu_{ki}^{m}\left(x_{ij}-v_{kj}\right)^2+\lambda\sum_{k=1}^{c}\sum_{i=1}^{n}(1-\mu_{ki})^m+\frac{n}{c}\sum_{j=1}^{d} w_j\log w_j$$

To construct a feature-reduction framework for FW-PCM, we consider extra parameters δ_j, $j=1,\cdots,d$ for controlling feature weights. We propose a novel objective function as follows:

$$J_{\text{FW-R-PCM}}(U,W,V)=\sum_{k=1}^{c}\sum_{i=1}^{n}\sum_{j=1}^{d}\delta_j w_j\mu_{ki}^{m}\left(x_{ij}-v_{kj}\right)^2+\lambda\sum_{k=1}^{c}\sum_{i=1}^{n}\sum_{j=1}^{d} w_j(1-\mu_{ki})^m+\frac{n}{c}\sum_{j=1}^{d} w_j\log\left(\delta_j w_j\right) \quad (5)$$

subject to $U=[\mu_{ki}]_{c\times n}\in M_{pcn}$ with $w_{kj}\in[0,1],\forall k,j$, and $\sum_{j=1}^{d} w_{kj}=1,\forall k$. The clustering algorithm based on $J_{\text{FW-R-PCM}}(U,W,V)$, called the feature-weighted reduction PCM (FW-R-PCM), is through the updating equations for minimizers of the FW-R-PCM objective function $J_{\text{FW-R-PCM}}(U,W,V)$. The first two terms of FW-R-PCM is similar with the FW-PCM objective function, and the third term is a feature-weighted entropy. We form the FW-R-PCM objective function, by multiplying the parameter δ_j in the first and third term of the FW-R-PCM objective function.

Our question is how to estimate parameters δ_j, $j=1,\cdots,d$. In statistics, the coefficient of variance (CV), defined as $CV=\sigma/\mu$, is often used as the index of dispersion. The reciprocal of CV is also known as signal-to-noise ratio (SNR), i.e. $SNR=\mu/\sigma$, that is widely used in quality engineering to evaluate the performance of a system. Further, in physics, Fano factor (FF), which can be seen as a similar CV, had been proposed and defined as $FF=\sigma^2/\mu$ (see [36]). The $FF=\sigma^2/\mu$ can measure data points that tend to be closer to the cluster center when FF is small, otherwise data points will be far from the cluster center. Since our goal is to be able to discard irrelevant features, these features must exhibit large values of dispersion and only those features with smaller dispersions will be retained. If we consider the reciprocal of Fano factor, that is similar as SNR being the reciprocal of CV, then we have μ/σ^2, i.e., mean-to-variance ratio. That is, we define $\delta_j=\mu/\sigma^2$ that will relatively influence each feature regulated by its corresponding weight w_j so that δ_j can be used to control the strength of feature weights.

Based on the Lagrangian of the FW-R-PCM objective function $J_{\text{FW-R-PCM}}(U,W,V)$, the necessary conditions for minimizing $J_{\text{FW-R-PCM}}(U,W,V)$ can be derived as follows:

$$\mu_{ki}=\left(1+\left(\lambda^{-1}\sum_{j=1}^{d}\delta_j w_j\left(x_{ij}-v_{kj}\right)^2\right)^{\frac{1}{m-1}}\right)^{-1} \quad (6)$$

$$v_{kj}=\sum_{i=1}^{n}\left(\mu_{ki}\right)^m x_{ij}\Big/\sum_{i=1}^{n}\left(\mu_{ki}\right)^m \quad (7)$$

$$w_j=\frac{\frac{1}{\delta_j}\exp\left(\frac{-c\left(\sum_{k=1}^{c}\sum_{i=1}^{n}\delta_j\left(u_{ki}\right)^m\left(x_{ij}-v_{kj}\right)^2+\lambda\sum_{k=1}^{c}\sum_{i=1}^{n}\left(1-u_{ki}\right)^m\right)}{n}\right)}{\sum_{s=1}^{d}\frac{1}{\delta_j}\exp\left(\frac{-c\left(\sum_{k=1}^{c}\sum_{i=1}^{n}\delta_j\left(u_{ki}\right)^m\left(x_{ij}-v_{kj}\right)^2+\lambda\sum_{k=1}^{c}\sum_{i=1}^{n}\left(1-u_{ki}\right)^m\right)}{n}\right)} \quad (8)$$

The proof of Eqs. (6), (7) and (8) is similar to Theorem 1. Based on updating equations (6)-(8), the proposed FW-R-PCM clustering algorithm can be summarized as follows:

FW-R-PCM Algorithm

Fix $\in>0$, $2\le c\le n$ where c is the number of clusters.

Randomly choose initial cluster center, $V^{(0)}$ and feature

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.2968879, IEEE Transactions on Fuzzy Systems

weight $W^{(0)} = \left[w_j \right]_{1xd}, 1 \le j \le d$ with $\sum_{j=1}^{d} w_j = 1$. Set $q = 1$.

Step1: Calculate $\delta_j = \overline{X} / Var(X)$.

Step 2: Calculate the possibilistic values, $U^{(q)}$, by equation (6) using $V^{(q-1)}$ and $W^{(q-1)}$.

Step 3: Update cluster centers $V^{(q)}$ by equation (7) using $U^{(q)}$.

Step 4: Update feature weights, $W^{(q)}$, by equation (8) using $V^{(q)}$ and $U^{(q)}$.

Step 5: Discard the total d_s number of these j features having $\le 1/\sqrt{ncd}$ for $W^{(q)}$, and set $d_{new} = d - d_s$.

Step 6: Adjust $W^{(q)}$ to satisfy the condition $\sum_{j=1}^{d} w_{kj}^{(q)} = 1, \forall k$.

Step 7: IF $\left\| W^{(q)} \right\| - \left\| W^{(q-1)} \right\| < \in$ STOP

ELSE, set $q = q + 1$, $d = d_{new}$ and go back to Step 1.

The FW-R-PCM algorithm starts with initializing the feature weights of each cluster and the cluster centers. After which, the possibilistic membership values are calculated. The result is then used to update the cluster centers and the feature weights. Once new feature weights are computed, those with small weights are discarded based on the threshold $1/\sqrt{ncd}$. Adjustment of points and cluster centers using the new features are made. The process is then repeated using these new sets of points and cluster centers until final feature weights are obtained. During the iterative process, features with smaller weights are discarded and the remaining features are updated with new weights. This will result to a decrease in the number of features until it reaches convergence.

In most data sets of high dimensionality, several features may seem irrelevant. Nevertheless, they were gathered, probably, for the sake of including additional information for the entire data. Sometimes these features may not actually contribute to the goal of the researcher and needed to be discarded. The question is, if we have a data set of high dimensionality, which of these features will be irrelevant and how they can be discarded. In the proposed algorithm, equal weights were initially assigned to each of these features and in the process, those features with very small weights are discarded by using a suitable threshold. The given data set has n data points and d dimensions and the restriction that $\sum_{i=1}^{n} w_j = 1$. Intuitively, if the dimension is quite large, the threshold may be chosen as $1/d$. This threshold to reduce the features may not be fitted for most data sets especially on those cases where d is small. In this algorithm, we considered additional two factors, n, as the total number of data points and c, the number of clusters and use $1/\sqrt{ncd}$ as our basis for discarding small weights. If we have a large n and d, $1/\sqrt{ncd}$ will be very small and may not detect smaller weights. By taking the square root of n, c and d, the denominator becomes smaller which may deem suitable as threshold of the algorithm to discard the irrelevant features in the data set.

We finally analyze the complexity of the FW-R-PCM algorithm. FW-R-PCM is scalable to the number d of features, the number n of data points, and the number c of clusters. A feature-weight entropy is added to the FW-R-PCM objective function that calculates the new weights of each feature in each cluster. To analyze the complexity, we consider three major computational steps in the FW-R-PCM algorithm: (1) Compute the possibilisitic partition memberships, u_{ik}; (2) Update the cluster centers, v_k; and (3) compute the feature weights, w_j. In computing possibilisitic partition memberships, once the feature weights of each cluster and the cluster centers were initialized, a possibilistic membership value is assigned to each data point. This step has a complexity of $O(ncd)$. Using the possiblitic partition matrix, U, cluster centers are updated by finding the means of objects in the same cluster. Thus, for c clusters, the computational complexity is $O(nc)$. The last step is to update feature weights for all clusters based on the possibilistic partition matrix, U, and the cluster centers, V. The computational complexity is $O(ncd^2)$. Thus, the complexity of the FW-R-PCM algorithm should be $O(ncd^2)$.

IV. EXPERIMENTAL COMPARISONS AND RESULTS

In this section, we investigate the performance of the proposed FW-PCM and FW-R-PCM algorithms. Several synthetic data sets are constructed to get more insights to behaviors of FW-R-PCM in identifying relevant features with reducing irrelevant ones. We also make comparisons of the proposed algorithms with PCM [14], FRFCM [23], EW-KM [17], NW-FCM [28], SW-PCM [29] and W-PCM [30]. The performance of a clustering algorithm can be evaluated by calculating its accuracy rate (*AR*). The *AR* is defined as $AR = \sum_{i=1}^{k} r_i / n$, where r_i is the number of points in $C_i^{'}$ that are also in C_i and n is the number of data points in which $C = \{C_1, C_2, \cdots, C_c\}$ is the set of c clusters for the given data set and $C^{'} = \{C_1^{'}, C_2^{'}, \cdots, C_c^{'}\}$ is the set of c clusters generated by the algorithm. AR is the percentage of points that are correctly identified in the clustering results. Another evaluation of clustering performance is the Rand Index (*RI*) [37] where it measures the similarity between two clustering algorithms. Let (X_i, X_j) be a given pair of points in the data set. Let a be the number of pairs of points if both points belong to the same cluster in C and the same cluster in $C^{'}$, b be the number of pairs of points if both points belong to two different clusters in C and two different clusters in $C^{'}$, c be the number of pairs of points if the two points belong to the same cluster in C and different clusters in $C^{'}$, and d be the number of pairs of points if the two points belong to two

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.2968879, IEEE Transactions on Fuzzy Systems

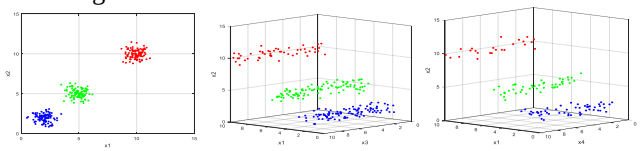
different clusters in C and to the same cluster in C'. The *RI* is calculated as $R=(a+d)/M$ where $M=n(n-1)/2$ is the total number of possible pairs of points in the data set and *n* is the number of data points.

Except RI, more external validation indexes are also used to evaluate the performance of clustering algorithms. The Jaccard similarity index (*JI*) [38] compares members for two sets X and Y to see which members are shared and which are distinct. To calculate *JI*, it uses $J(X,Y)=|X\cap Y|/|X\cup Y|$. On the other hand, the adjusted Rand Index (*ARI*) [39-40] is a form of the Rand Index (*RI*) that is adjusted for the chance grouping of elements. It may yield negative values if the index is less than the expected index. The normalized mutual information (*NMI*) [41] measures the amount of information on the presence/absence of a term that contributes to making the correct classification decision [41, 23]. NMI can be calculated using $NMI=I(X:Y)/\left([H(X)+H(Y)]/2\right)$, where $H(X)$ and $H(Y)$ are the marginal entropies of X and Y, respectively, and $I(X:Y)$ is a mutual information between $H(X)$ and $H(Y)$. The normalized Hubert Statistic Γ [42] defined by the equation $\Gamma=(1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}P(i,j)\bullet Q(i,j)$ where $M=N(N-1)/2$, N is the number of objects in a dataset, P is the proximity matrix of the data set and Q is an $N\times N$ matrix whose (i,j) element is equal to the distance between the representative points (v_{ci},v_{cj}) of the clusters where the objects x_i and x_j belong. For simplicity of notation for Hubert statistic, we shall use *HI* instead of Γ. The Fowkles-Mallows index (*FMI*) [43] can be defined based on the number of points common or uncommon in the two clusterings. Let *TP* as the number of pairs of points that are present in the same cluster in both C and C', *FP* as the number of pairs of points that are present in the same cluster in C and but not in C', *FN* as the number of pairs of points that are present in the same cluster in C' and but not in C, *FP* as the number of pairs of points that are in different clusters in both C and C'. To compute FMI, we use the equation $FMI=\sqrt{\frac{TP}{TP+FP}\bullet\frac{TP}{TP+FN}}$. Except for *A*RI and *HI*, the values obtained ranges from 0 to 1. Higher values of these indexes imply that they are more similar. It indicates the existence of compact clusters. For each clustering algorithm, we make 100 simulations in all examples. This will check the consistency of results for each clustering algorithm.

Example 1. In this example, a 3-cluster data set with 300 data points is generated from a Gaussian mixture model with the parameters $(u_1,\Sigma_1)=([2,2],0.3I_2)$, $(u_2,\Sigma_2)=([5,5],0.3I_2)$ and $(u_3,\Sigma_3)=([10,10],0.3I_2)$. These are illustrated in Fig. 1(a). Since our objective is to identify relevant features, we extend the data set from the 2 features $\{x_1,x_2\}$ to 4 features $\{x_1,x_2,x_3,x_4\}$ by adding two features, x_3 and x_4, that are uniformly distributed over intervals [0,10] and [0, 20], as shown in Figs. 1(b) and (c), respectively. Since the 4-D data set projected on the 2-D plane of $\{x_1,x_2\}$ obtains the same as the original 2-D data set, it implies that the two added features, x_3 and x_4, are irrelevant features. We also show different structures by constructing a scatter plot for different combinations of any two features in the data set, as shown in Table I. If we look at the scatter plot of x_3 and x_4, their combination only shows that no vivid clustering structure is evident. We first apply PCM on the original features x_1 and x_2 with m=1.5 (as suggested by Krishnapuram & Keller [25]) and the initial cluster centers from the FCM algorithm. The obtained clustering results from PCM are shown in Fig. 2(a) that give good clustering with almost the same structure as the original data of Fig. 1(a). However, if we apply PCM with the same initials on the 4-D data set of four features $\{x_1,x_2,x_3,x_4\}$, then the obtained clustering results, as shown in Fig. 2(b) with the 3-D of $\{x_1,x_2,x_3\}$ and Fig. 2(c) with the 2-D of $\{x_1,x_2\}$, present very poor clustering, where all data are almost in one cluster, except the other three data points are in the other two clusters. This tells us that the added features, x_3 and x_4, actually disrupt clustering results of the PCM algorithm and it is heavily affected by these irrelevant features with poor clustering performance. This is an obvious drawback of PCM in which we would like to improve it. We apply the proposed FW-PCM algorithm on the 4-D data set of four features $\{x_1,x_2,x_3,x_4\}$ with the same initials as PCM and equal initial feature weights. Next, we randomly choose one of the 100 simulations and observe the behavior of the weights of the features in each iteration from FW-PCM. The results are shown in Table II. It is seen that FW-PCM produces very small weights to these irrelevant features x_3 and x_4 after the first iteration, and continues until it stops at the 5th iteration. To analyze the behavior of FW-PCM, the measures of *AR* and these external validation indexes *RI*, *ARI*, *NMI*, *HI*, *JI* and *FMI* are used to evaluate the performance. The results for each iteration are also summarized in Table II. From Table II, it is seen that FW-PCM obtains *AR*=0.97, *RI*=0.66, *ARI*=0.25, *NMI*=0.37, *HI*=0.33, *JI*=0.34 and *FMI*=0.51 during the first iteration. The values of *RI*, *ARI*, *NMI*, *HI*, *JI* and *FMI* will increase in the succeeding iterations until it stops after the 5th iteration. To further observe the behavior of FW-PCM for the 4-D data set with four features, we use initial equal weights, $m=1.5$ and randomly choose 4 out of 100 simulations with different randomly generated initial cluster centers for the FW-PCM algorithm. These final feature weights and the values of *RI*, *ARI*, *NMI*, *HI*, *JI* and *FMI* are shown in Table III. From Table III, FW-PCM actually obtains stable results with high accuracy rate and good validation index values. Its 3-D projection and 2-D projection are also shown in Figs. 2(d) and

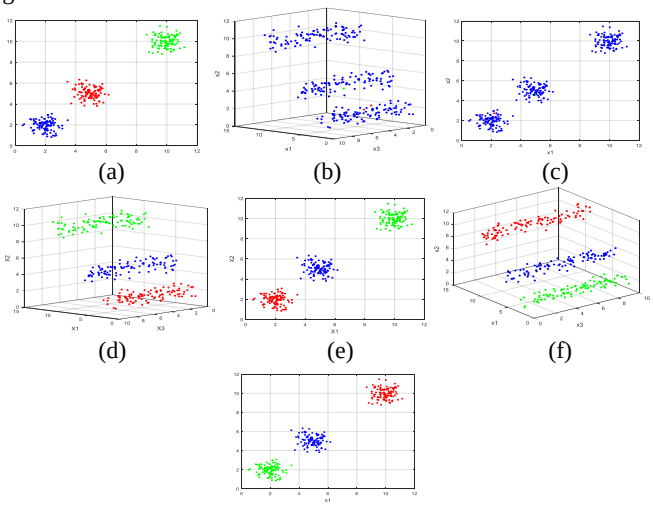
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.2968879, IEEE Transactions on Fuzzy Systems

(e), respectively. It shows a good clustering structure. Thus, FW-PCM can identify x_3 and x_4 as irrelevant features because of their small feature weights compared to more weights of the original features x_1 and x_2. We then apply the proposed FW-R-PCM for the data set with the same initials as FW-PCM by choosing 4 out of 100 simulations. These 4 random trials are used to show and compare the behavior of FW-PCM and FW-R-PCM. The results are shown in Table IV-1. It is seen that FW-PCM consistently identifies the irrelevant features by assigning very small weights on the features X_3 and X_4, but FW-R-PCM automatically discards these irrelevant features. Furthermore, Tables IV-2 show the results of *AR*, *RI*, *ARI*, *NMI*, *HI*, *JI*, and *FMI* from FW-R-PCM and FW-PCM. It is seen that the FW-R-PCM algorithm always has better *AR*s, *RI*s, *ARI*s, *NMI*s, *HI*s, *JI*s, and *FMI*s than FW-PCM. Also, the 3-D projection and 2-D projection of the clustering results from FW-R-PCM are shown in Figs. 2(f)-(g), respectively. They show good clustering structures.



(a) (b) (c)

Fig. 1 (a) The 3-cluster data set generated from a Gaussian mixture model; (b) Three features $\{x_1, x_2, x_3\}$ where x_1 and x_2 are the same in (a) and x_3 is generated from a uniform distribution; (c) Three features $\{x_1, x_2, x_3\}$ where x_1 and x_2 are the same in (a) and x_4 is generated from a uniform distribution.



(a) (b) (c) (d) (e) (f) (g)

Fig. 2 (a) PCM result using x_1 and x_2; (b) 3-D PCM result using 4 features; (c) 2-D PCM using 4 features; (d) 3-D FW-PCM using 4 features; (e) 2-D FW-PCM using 4 features; (f) 3-D FW-R-PCM using 4 features; (g) 2-D FW-R-PCM using 4 features

TABLE I

SCATTER PLOTS FOR ALL POSSIBLE COMBINATIONS OF ANY TWO FEATURES WHERE x_1 AND x_2 ARE GENERATED FROM THREE-COMPONENT GAUSSIAN MIXTURE DISTRIBUTION, WHILE x_3 AND x_4 ARE GENERATED FROM UNIFORM DISTRIBUTION

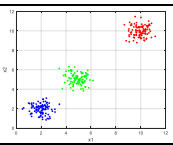
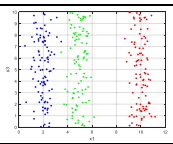
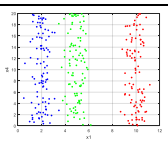
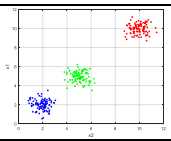
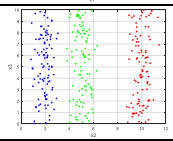
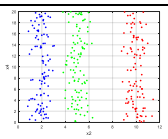
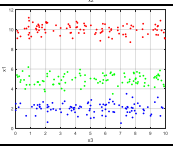
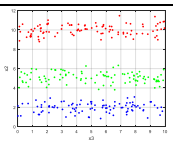
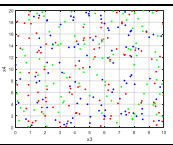
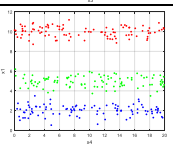
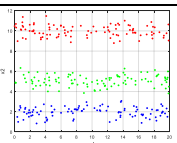
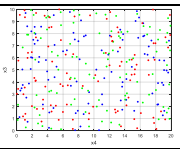
y-axis / x-axis	x_1	x_2	x_3	x_4
x_1	-			
x_2		-		
x_3			-	
x_4				-

TABLE II

FEATURE WEIGHTS, *AR*, *RI*, *ARI*, *NMI*, *HI*, *JI* AND *FMI* FROM FW-PCM FOR DIFFERENT ITERATIONS

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.2968879, IEEE Transactions on Fuzzy Systems

Iteration No.	Feature weights from FW-PCM				FW-PCM						
	X_1	X_2	X_3	X_4	*AR*	*RI*	*ARI*	*NMI*	*HI*	*JI*	*FMI*
1	0.55	0.44	0.01	9.0E-8	0.97	0.67	0.25	0.37	0.33	0.34	0.51
2	0.72	0.28	2.0E-11	1.7E-105	1.00	1.00	0.99	0.98	0.99	0.99	0.99
3	0.81	0.19	3.4E-12	2.4E-69	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	0.83	0.17	1.7E-12	3.1E-70	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5	0.84	0.16	1.4E-12	1.9E-70	1.00	1.00	1.00	1.00	1.00	1.00	1.00

TABLE III
FINAL FEATURE WEIGHTS, *AR, RI, ARI, NMI, HI, JI* AND *FMI* FROM FW-PCM WITH INITIAL EQUAL WEIGHTS AND DIFFERENT RANDOMLY GENERATED CLUSTER CENTERS

Random trials	Final feature weights from FW-PCM				FW-PCM						
	X_1	X_2	X_3	X_4	*AR*	*RI*	*ARI*	*NMI*	*HI*	*JI*	*FMI*
1	0.84	0.16	1.2E-12	1.6E-70	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2	0.92	0.08	8.5E-14	1.4E-80	0.99	0.78	0.57	0.76	0.55	0.60	0.77
3	0.84	0.16	1.2E-12	1.6E-70	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	0.92	0.08	8.5E-14	1.2E-80	0.99	0.75	0.50	0.59	0.49	0.54	0.72

TABLE IV-1
COMPARISON OF FINAL FEATURE WEIGHTS BETWEEN FW-PCM AND FW-R-PCM

Random Trials	Final feature weights from FW- PCM				Final feature weights from FW-R-PCM			
	X_1	X_2	X_3	X_4	X_1	X_2	X_3	X_4
1	0.84	0.16	1.2E-12	1.6E-70	0.48	0.52	-	-
2	0.92	0.08	8.5E-14	1.3E-80	0.46	0.54	-	-
3	0.67	0.33	1.63E-11	2.4E-60	0.80	0.20	-	-
4	0.84	0.16	1.2E-12	1.6E-70	0.27	0.73	-	-

TABLE IV-2
COMPARISON OF *AR, RI, ARI, NMI, HI, JI* AND *FMI* BETWEEN FW-PCM AND FW-R-PCM BASED ON THE TRIALS OF TABLE IV-1

	FW-PCM				FW-R-PCM			
	Trial 1	Trial 2	Trial 3	Trial 4	Trial 1	Trial 2	Trial 3	Trial 4
AR	**1.00**	**0.99**	0.99	**1.00**	**1.00**	**0.99**	**0.99**	**1.00**
RI	**1.00**	0.78	0.75	**1.00**	**1.00**	**0.91**	**1.00**	**1.00**
ARI	**1.00**	0.57	0.50	**1.00**	**1.00**	**0.79**	**0.99**	**1.00**
NMI	**1.00**	0.76	0.59	**1.00**	**1.00**	**0.82**	**0.98**	**1.00**
HI	**1.00**	0.55	0.49	**1.00**	**1.00**	**0.81**	**0.99**	**1.00**
JI	**1.00**	0.60	0.54	**1.00**	**1.00**	**0.76**	**0.99**	**1.00**
FMI	**1.00**	0.77	0.72	**1.00**	**1.00**	**0.86**	**0.99**	**1.00**

TABLE V
COMPARISON OF AVERAGE/STANDARD DEVIATION FOR *AR, RI, ARI, NMI, HI, JI* AND *FMI* FROM CLUSTERING ALGORITHMS

	Clustering Algorithms							
	FW-PCM	FW-R-PCM	FRFCM	PCM	EW-KM	NW-FCM	W-PCM	SW-PCM
AR	**0.989/0.003**	**0.992/0.001**	0.976/0.013	0.991/0.004	0.973/0.010	0.965/0.004	0.965/0.003	0.965/0.002
RI	**0.846/0.079**	**0.819/0.083**	0.739/0.147	0.250/1E-05	0.587/0.134	0.620/0.024	0.675/0.038	0.650/0.029
ARI	**0.633/0.162**	**0.576/0.147**	0.339/0.375	-8E-06/1.3E-05	0.094/0.151	0.000/0.002	0.170/0.084	0.111/0.062
NMI	**0.742/0.103**	**0.702/0.097**	0.375/0.000	0.013/9.6E-05	0.134/0.175	0.003/0.002	0.192/0.092	0.132/0.072
HI	**0.691/0.157**	**0.637/0.166**	0.478/0.013	-0.499/2E-05	0.175/0.267	0.241/0.048	0.349/0.076	0.299/0.059
JI	**0.597/0.150**	**0.545/0.116**	0.400/0.285	0.249/6.3E-06	0.228/0.087	0.145/0.012	0.244/0.048	0.211/0.034
FMI	**0.744/0.105**	**0.707/0.088**	0.518/0.278	0.499/1E-05	0.378/0.109	0.2540.024	0.391/0.060	0.349/0.046

We next compare our proposed FW-PCM and FW-R-PCM algorithms with FRFCM [23], PCM [14], EW-KM [17], NW-FCM [28], W-PCM [30] and SW-PCM [29].

Example 2 In this example, we consider a data set from Yang & Nataliani [23]. The data set consists of 4 clusters with 1000 data points generated from the Gaussian mixture (GM) distribution $\sum_{k=1}^{4}\alpha_k N(u_k,\Sigma_k)$ with parameters $\alpha_k = 1/4,\ \forall k$, $u_1 = (10\quad 5)^T$, $u_2 = (10\quad 10)^T$, $u_3 = (10\quad 15)^T$, $u_4 = (10\quad 20)^T$, and $\Sigma_k = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ \forall k$, as shown in Fig. 3(a). The 2-D graph of the Gaussian mixture distribution is converted into 3-D graph, as shown in Fig. 3(b), using the mapping $(x_1, x_2) \to (x_1, x_2, x_3) = (x_1 \cos x_1, x_2, x_1 \sin x_1)$. The 3-D data set is then projected into the Cartesian plane, as shown in Fig. 3(c). We first implement PCM on the data set of Fig. 3(b) with features x_1, x_2 and x_3, where its clustering results, that are projected on the 2-D plane, are shown in Fig. 3(d). From Fig.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.2968879, IEEE Transactions on Fuzzy Systems

3(d), it gives very poor clustering, where all data are almost in one cluster. That is, PCM is actually affected by the irrelevant feature with poor clustering. We also implement the FW-PCM algorithm on the 3-D data set of features $\{x_1, x_2, x_3\}$ with the same initial as PCM and initial weights 0, 1, and 0 to features x_1, x_2 and x_3, respectively. The clustering results projected on the 2-D plane from FW-PCM are shown in Fig. 3(e). It is seen that FW-PCM produces much better results than PCM. This is because FW-PCM obtains very small weights to these irrelevant features x_1 and x_3, compared to the relevant feature x_2. Similarly, we implement the proposed FW-R-PCM and also FRFCM [23] on the 3-D data set of features $\{x_1, x_2, x_3\}$ with all the same initial as PCM and FW-PCM. The clustering results projected on the 2-D plane from FW-R-PCM and FRFCM are shown in Figs. 3(f) and 3(g), respectively. It is seen that both FW-R-PCM and FRFCM obtain good clustering results. In order to compare FW-PCM, FW-R-PCM with PCM, NW-FCM, EW-KM, SW-PCM, W-PCM and FRFCM, we take 100 different initials and then compute their average values and standard deviations of *AR*s, *RI*s, *ARI*s, *NMI*s, *HI*s, *JI*s and *FMI*s. These results are shown in Table V. It is seen that the proposed FW-PCM and FW-R-PCM algorithms always give the best *AR*, *RI*, *ARI*, *NMI*, *HI*, *JI* and *FMI*.

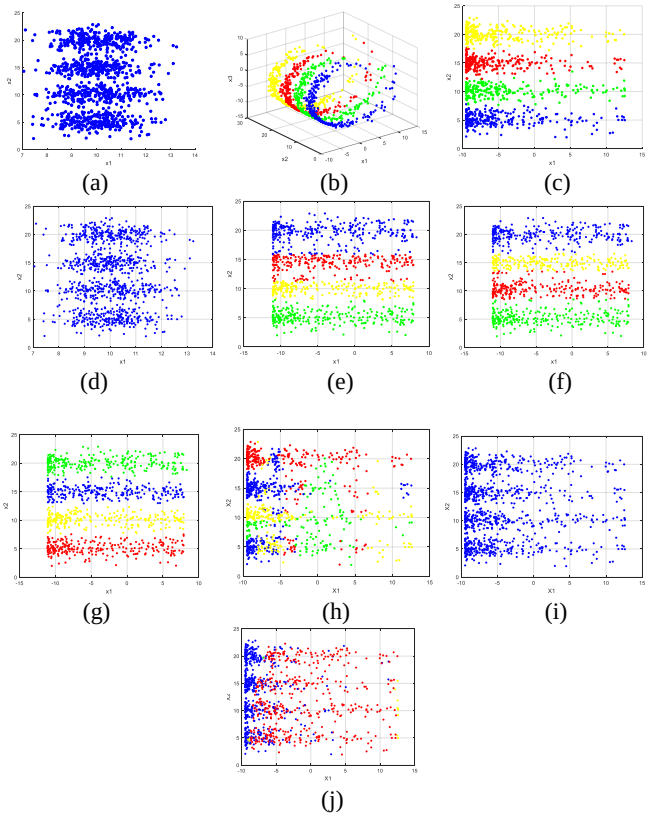


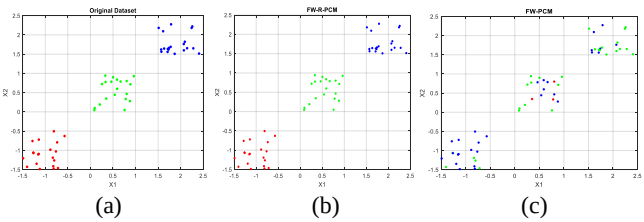
Fig. 3 (a) The 2-D data set generated from GM distribution; (b) The 3-D data set of features $\{x_1, x_2, x_3\}$; (c) The 2-D projection of 3-D data set; The 2-D projection of clustering results of the 3-D data set using (d) PCM; (e) FW-PCM; (f) FW-R-PCM; (g) FRFCM; (h) SW-PCM; (i) EW-KM and (j)NW-FCM.

We mention that SW-PCM [29] and W-PCM [30] generate the weights w_j of the membership of x_j using

$$w_j = \sum_{k=1}^{c} \exp\{-\alpha \|x_i - v_k\|\} \quad \& \quad w_j = \left(1 - \left(l_j / m\right)\right)^t \sum_{k=1}^{c} \exp\{-\alpha \|x_i - v_k\|\},$$

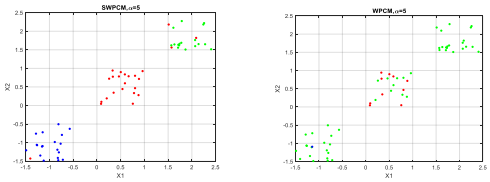
respectively. In both equations, the constant $\alpha > 0$ controls the weight values which are suitably chosen. In next example, we will compare our proposed algorithm with SW-PCM and DW-PCM for different values of α.

Example 3 In this example, we generate a 3-cluster data set using the method of Qiu et al. [44]. Let $X_{n\times d}$ with n observations and d features. The elements x_{ij} are independently and normally distributed. The observations fall into 3 classes: C_1, C_2 and C_3 which differ only with respect to the first $q\,(q < d)$ features. Three different normal distributions with a mean shift $\mu, -\mu,$ and 0, respectively, among the three classes only in the first q features. The rest of the $d-q$ features have 0 means are considered as irrelevant features. The classes are pair-wise non-intersecting and the union of any two classes is not empty. Note as well that each observation belongs and only belongs to $C_1, C_2,$ or C_3. Using the above method, we generate a dataset with $n = 60$ and $d = 25$. We set $m = 2$, mean $\mu = 1.5$ and $q = 5$. The first 5 features are relevant and the remaining 20 features are considered irrelevant. We apply FW-R-PCM and FW-PCM on this dataset. Since the weights for SW-PCM [29] and W-PCM [30] are dependent on the choice of α, we assign α with values 0.5, 1.5, 2.75, 3.5, 5, 7.5, 10 and 12.25 and apply SW-PCM and W-PCM algorithms on this dataset. The results of SW-PCM and W-PCM are compared to the results of FW-R-PCM and FW-PCM. Fig. 4 shows the graph of the original data and the clustering performance of these algorithms. In Fig. 4, we display only the graph of SW-PCM and W-PCM for $\alpha = 0.5$ chosen randomly to show their clustering performance. It can be seen from the figure that FW-R-PCM clustering result is the same as the original dataset as compared to the others. This indicates that FW-R-PCM gives a good clustering result. Fig. 5 shows a line plot comparison of the different evaluation criteria of these algorithms. It is seen from the figure that FW-R-PCM gives the best clustering performance with a value of 1.0 across all criteria for different values of α assumed for SW-PCM and W-PCM. Although FW-PCM gives some lower results compared to SW-PCM and W-PCM, SW-PCM and W-PCM do not have the capability of assigning weights to features, because weights are for membership degrees of objects.



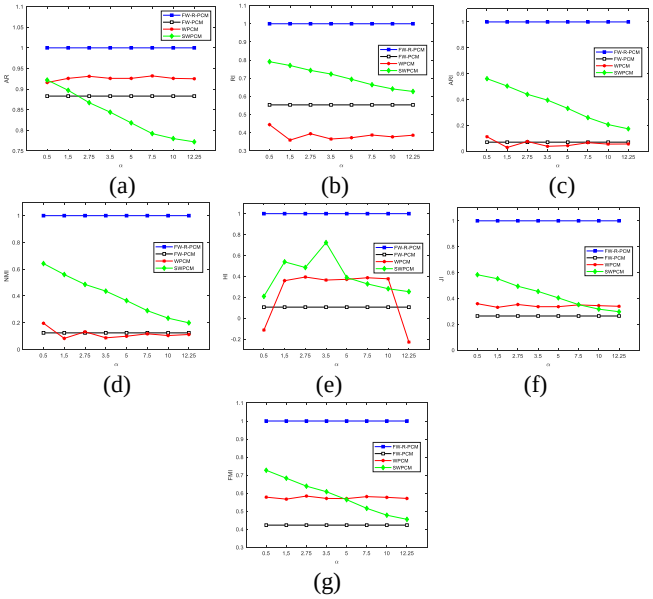
(a) (b) (c)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.2968879, IEEE Transactions on Fuzzy Systems



(d) (e)

Fig. 4 Graphical comparison: (a) Original dataset; (b) FW-R-PCM; (c) FW-PCM; (d) SW-PCM with $\alpha = 5$; (e) W-PCM with $\alpha = 5$.



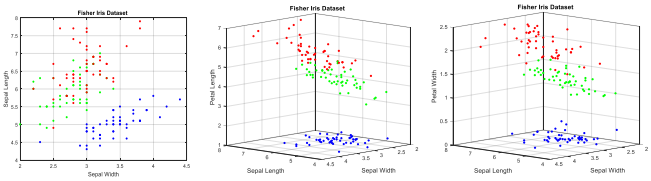
(a) (b) (c) (d) (e) (f) (g)

Fig. 5 Comparison among FW-R-PCM, FW-PCM, SW-PCM and W-PCM based on evaluation criteria.

TABLE VI
FEATURE REDUCTION FOR THE IRIS DATA SET USING FW-R-PCM

Iteration	Feature Weights of each Attribute			
	Sepal Length	Sepal Width	Petal Length	Petal Width
1	0.061	0.031	0.546	0.362
2	0.027	-	0.566	0.407
3		-	0.571	0.429
4		-	0.573	0.427
5		-	0.573	0.427
6		-	0.573	0.427

Example 4 This example uses the Iris data set. The Iris data set [45], first introduced by Fisher [46], contains 3 classes: setosa, virginica and versicolor of 50 instances each. Four features were measured from each sample: length and width of the sepals and petals in centimeters. Figs. 6(a)-(c) shows a 2D and 3D illustrations of the iris data set. The graphs show that some data points of the Iris data set tend to overlap each other. Equal weights and 100 different randomly generated cluster centers are used as initials. Using a weighting exponent $m = 2$, we apply FW-R-PCM to the Iris data set. Table VI summarizes the behavior of FW-R-PCM for each succeeding iterations. It can be observed that FW-R-PCM discards the sepal width feature after the 1st iteration and sepal length after the 2nd iteration and consistently identifies petal length and petal width as the most important features. We also compare the clustering performance of FW-PCM and FW-R-PCM with FRFCM, PCM, EW-KM, NW-FCM, W-PCM and SW-PCM. Table VII summarizes the clustering performance of these algorithms based on the evaluation criteria of *AR*, *RI*, *ARI*, *NMI*, *HI*, *JI* and *FMI*. From Table VII, it is seen that the proposed FW-R-PCM obtains the highest *AR*=0.987 and *NMI*=0.777 while EW-KM have the highest *RI*, *ARI*, *HI*, *JI* and *FMI*. The proposed FW-R-PCM gives the best *AR*.



(a) (b) (c

Fig. 6 The Iris data set (a) 2D graph using sepal length and sepal width (b) 3D graph using sepal length, sepal width and petal length (c) 3D graph using sepal length, sepal width and petal width.

Example 5 Several real data sets, aside from Iris, are considered to make the comparisons of FW-PCM and FW-R-PCM with FRFCM, PCM, EW-KM, NW-FCM, W-PCM and SW-PCM. These real data sets, as shown in Table VIII, are taken from the UCI Machine Learning Repository [45] with insects (Flea), disease data (Pima), plant data (Seeds, Soybean), power data (Solar) and chemical analysis data (Wine).

TABLE VIII
LIST OF REAL DATA SETS

Data Set	No. of points	Feature dimensions	No. of Clusters
Iris	150	4	3
Flea	74	5	3
Pima	768	8	2
Seeds	210	7	3
Solar	323	12	6
Soybean	47	21	4
Wine	129	13	3

We use these real data sets, Iris, Flea, Pima, Seeds, and Wine to compare only the values of *AR*, *RI* and *NMI* of FW-R-PCM and PCM with different values of K. These results are shown in Fig. 7. The first column, showing *AR* (accuracy rate), illustrates that FW-R-PCM gives better clustering results as compared with PCM under different values of K. The second column, depicting *RI* (Rand index), except for Pima, shows that FW-R-PCM gives better clustering results with higher *RI* as compared with PCM. In the third column, the *NMI* values demonstrate that FW-R-PCM gives better results with higher *NMI* than PCM, except for the Pima data set. Among all of these compared clustering algorithms, only FW-R-PCM and FRFCM have the capability of reducing the number of features for a data set. SW-PCM, W-PCM and NW-FCM only use feature weights to improve clustering performance. Although FW-PCM and EW-KM consider feature weights for feature selection, they do not reduce the number of features. Thus, we use these real data sets to compare *AR* and the number of remaining dimensions of FW-R-PCM and FRFCM. Both algorithms use m=2, equal weights and 100 randomly different cluster centers as initials. We give the average values and standard deviations of *AR* from FW-R-PCM and FRFCM for these 100 randomly initial cluster centers. A summary of results is shown in Table IX. Overall, FW-R-PCM performs better than FRFCM for

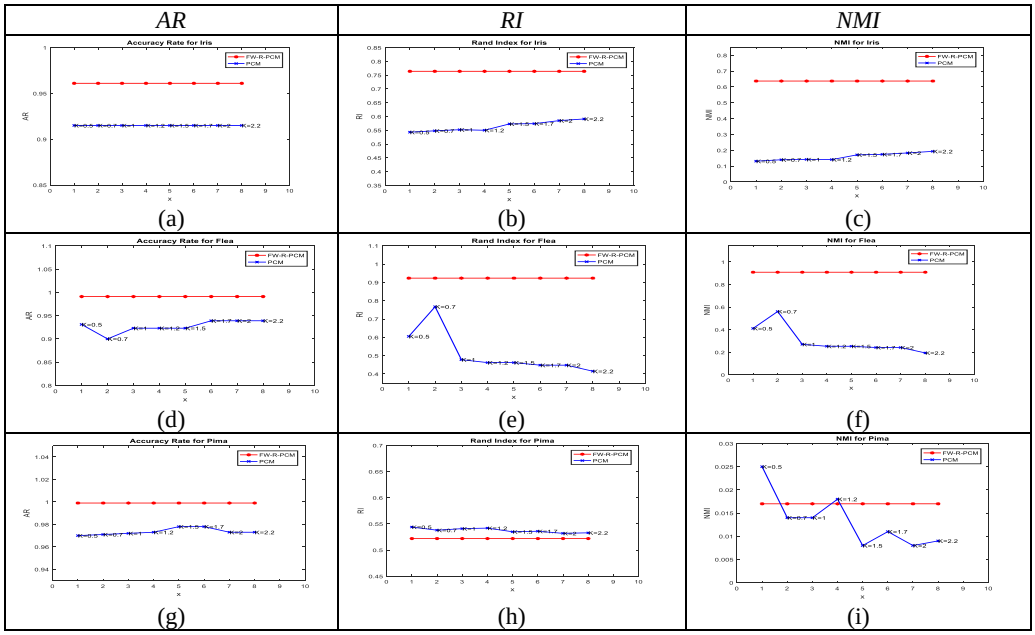
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.2968879, IEEE Transactions on Fuzzy Systems

these real data sets. Also, we compare the clustering performance of FW-R-PCM using these real data sets with other clustering algorithms using *AR, RI, ARI, NMI, HI, JI* and *FMI*. This is illustrated in Fig. 8. We observe from Fig. 8 that FW-R-PCM has the highest average values among all evaluation criteria *AR, RI, ARI, NMI, HI, JI* and *FMI* for the Flea data set with the smallest standard deviation implying that the results are consistent. This means that the data points are more clustered toward its center and gives the best clustering performance. FW-R-PCM gives also the highest average *AR* for the Iris, Solar and Soybean data sets and highest average *NMI* for the Iris data set. FW-PCM has the highest average *AR, RI, JI* and *FMI* for the Pima and Soybean data sets, highest average *HI* for the Pima data set and highest average *NMI* for the Soybean data set. EW-KM gives the highest average *RI, ARI, HI, JI* and *FMI* for the Iris data set. PCM gives the highest average *AR* for Pima and Seeds data sets. NW-FCM gives the highest average *RI, ARI, HI, JI*, and *FMI* for Solar and Seeds data sets. Furthermore, FW-R-PCM has the capability of reducing the number of features and choosing only the relevant ones of which the other clustering algorithms do not have. Overall, FW-R-PCM gives a good clustering performance for most of these real data sets.

Example 6 In this example, we compare the clustering performance of FW-R-PCM with FRFCM base on average values and standard deviations of *AR*, and the number of remaining dimensions for real data sets of high dimensions. Real data sets such as USPS, Lung, Ovarian cancer, SMK-CAN-187 and GLI-85 from the UCI Machine Learning Repository [45] are used. A summary of results from FW-R-PCM and FRFCM are shown in Table X. Both algorithms use m=2, equal weights and 100 randomly different cluster centers as initials. The average values and standard deviations of *AR* from FW-R-PCM and FRFCM are counted, as shown in Table X. It is seen that FW-R-PCM performs better than FRFCM for lung, ovarian cancer and GLI-85 data sets and the same for USPS and SMK-CAN-187. In Fig. 9, we show the behavior of running time (in seconds) for each iteration of FW-R-PCM using the data sets USPS and SMK-CAN-187. It can be observed from Fig. 9 that for all cases, the computation time for the first iteration is longer, but after that, the succeeding iterations shows a faster decrease.

TABLE VII
COMPARISON OF AVERAGE/STANDARD DEVIATION FOR *AR, RI, ARI, NMI, HI, JI* AND *FMI* FROM CLUSTERING ALGORITHMS ON THE IRIS DATA SET

	Clustering Algorithm							
	FW-PCM	FW-R-PCM	FRFCM	PCM	EW-KM	NW-FCM	W-PCM	SW-PCM
AR	0.953/0.016	**0.987/0.009**	0.968/0.018	0.932/0.030	0.971/0.014	0.964/0.010	0.964/0.018	0.937/0.020
RI	0.757/0.087	0.862/0.068	0.757/0.087	0.373/0.001	**0.881/0.020**	0.880/0.020	0.358/0.088	0.770/0.070
ARI	0.501/0.153	0.706/0.130	0.501/0.153	0.627/0.001	**0.734/0.039**	0.729/0.029	0.024/0.105	0.494/0.144
NMI	0.570/0.124	**0.777/0.078**	0.661/0.159	0.019/0.002	0.769/0.027	0.750/0.024	0.070/0.115	0.525/0.138
HI	0.515/0.173	0.725/0.137	0.515/0.173	-0.253/0.002	**0.762/0.40**	0.759/0.039	-0.284/0.177	0.540/0.140
JI	0.533/0.097	0.691/0.108	0.533/0.097	0.334/0.001	**0.701/0.033**	0.694/0.015	0.336/0.044	0.509/0.098
FMI	0.699/0.076	0.816/0.071	0.699/0.076	0.567/0.001	**0.824/0.022**	0.820/0.009	0.573/0.032	0.671/0.089



This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.2968879, IEEE Transactions on Fuzzy Systems

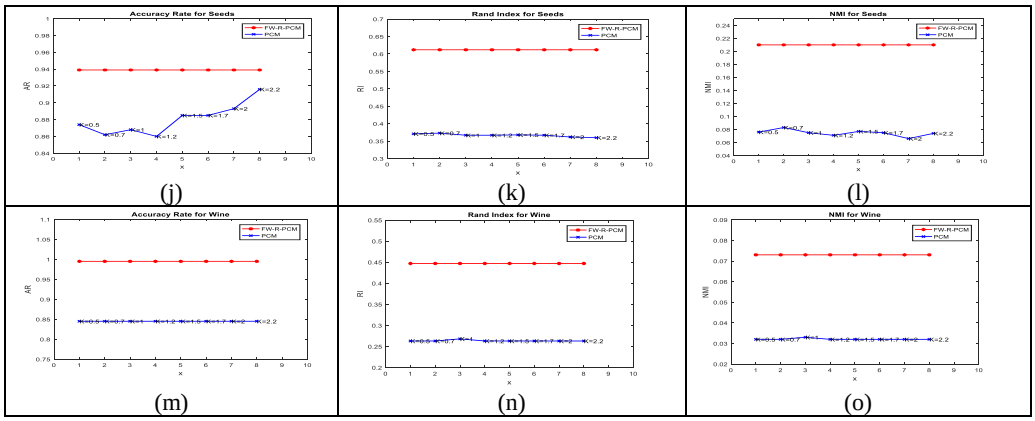
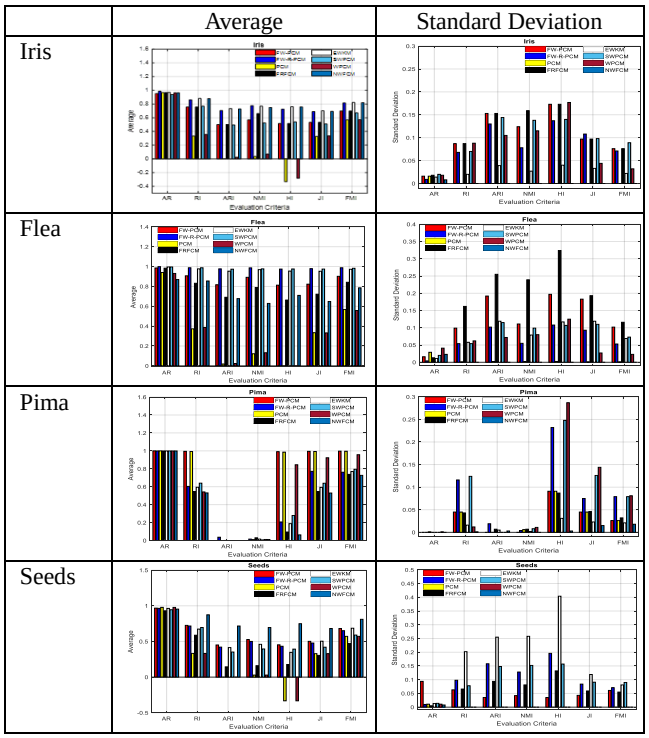


Fig. 7 Comparison of FW-R-PCM and PCM with different values of K for real data sets based on *AR*, *RI* and *NMI*; (a)-(c) Iris; (d)-(f) Flea; (g)-(i) Pima; (j)-(l)Seeds; (m)-(o) Wine.



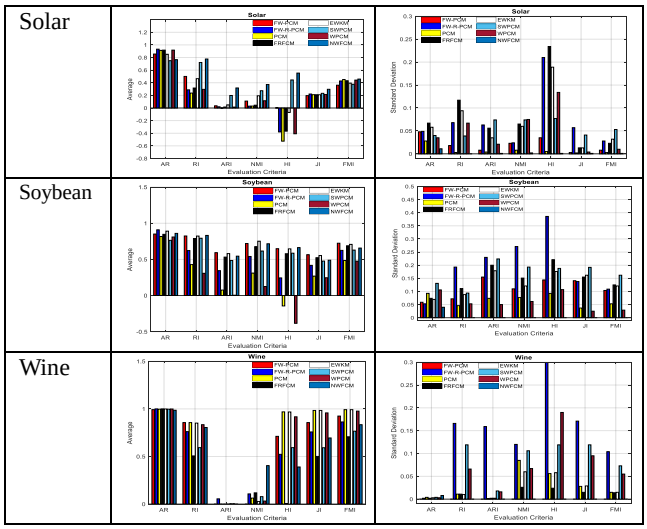


Fig. 8 Comparison of clustering performance of FW-R-PCM with FW-PCM, PCM, FRFCM, EW-KM, SW-PCM, W-PCM and NW-FCM using real data sets using evaluation criteria *AR*, *RI*, *ARI*, *NMI*, *HI*, *JI and FMI*.

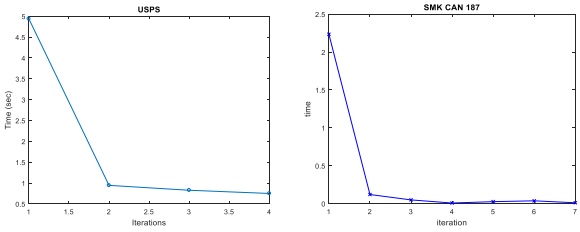
TABLE IX
COMPARISON OF FW-R-FCM AND FRFCM BASED ON REDUCED DIMENSION AND AVERAGE/STANDARD DEVIATION FOR *AR*

Real Data Set	Original dimensions	No. of remaining dimensions		Average and Standard Deviation of *AR*	
		FW-R-PCM	FRFCM	FW-R-PCM	FRFCM
Iris	4	2	2	**0.987/0.009**	0.968/0.018
Flea	5	3	2	**0.999**/0.040	0.984/**0.013**
Pima	8	1	1	**0.999/0.000**	**0.999/0.000**
Seeds	7	1	5	**0.965/0.010**	0.930/0.005
Solar	12	4	4	**0.933/0.040**	0.918/0.067
Soybean	21	11	11	**0.909/0.001**	0.847/0.074
Wine	13	2	7	**1.000/0.000**	**1.000**/0.002

TABLE X
COMPARISON OF FW-R-FCM AND FRFCM BASED ON REDUCED DIMENSION AND AVERAGE/STANDARD DEVIATION FOR *AR*

Real Data Set	No. of points	Feature dimension	No. of remaining dimensions		Average/Standard Deviation *AR*	
			FW-R-PCM	FRFCM	FW-R-PCM	FRFCM
USPS	9298	256	6	26	**0.950/0.001**	0.950/0.002

Lung	203	3312	38	75	**0.965/0.012**	0.914/0.023
Ovarian cancer	216	4000	111	65	**0.985**/0.005	0.982/**0.001**
SMK-CAN-187	187	19993	79	51	**0.979/0.000**	**0.979/0.000**
GLI-85	85	22283	37	37	**0.978**/0.012	0.955/**0.009**



(a) (b)

Fig. 9 Running time plots using FW-R-PCM per iteration for the data sets: (a) USPS (d=256); (b) SMK-CAN-187 (d=19993)

V. CONCLUSIONS AND DISCUSSION

In this paper we first proposed a new feature-weighted clustering algorithm, called a feature-weighted PCM (FW-PCM) algorithm. We then extend it to a feature-weighted reduction PCM (FW-R-PCM) clustering algorithm. FW-PCM has the characteristic of automatically identifying unimportant features by assigning small weights to it, while FW-R-PCM has the property of reducing automatically features as well as identifying those which are deemed irrelevant to produce good clustering results. In both objective functions, a feature-weighted entropy term is added for calculating a new weight for each feature. In each iteration, the possibilistic membership values and cluster centers for data are updated using these new weights. FW-R-PCM extends the idea of FW-PCM which generally improves the performance of PCM with the characteristic of selecting relevant features and discarding irrelevant features. Thus, reducing the number of features and retaining only those features which will generate better clustering results. The effectiveness of FW-PCM and FW-R-PCM was demonstrated with experimental results using synthetic and real data sets. A comparison of the algorithm with that of PCM shows that FW-PCM and FW-R-PCM produces better clustering performance. The results of FW-R-PCM is also comparable with that of FRFCM. In general, most of clustering algorithms rely on the necessity of assigning the number of clusters prior to its processing. FW-PCM and FW-R-PCM are not exempted from it. Thus, to further study FW-PCM and FW-R-PCM with automatically finding the optimal number of clusters should be our future topic. On the other hand, some real data sets may contain categorical data or mixed data types. There is still a need to extend FW-R-PCM so that it can be used for data sets with categorical or mixed data types, or even for sparse data with sparsity.

Acknowledgements

The authors would like to thank the anonymous referees for their helpful comments in improving the presentation of this paper.

REFERENCES

[1] T. Kanungo, D.M. Mount, N.S. Netanyahu, C. Piatko, R. Silverman, and A. Wu, "An efficient K-means clustering algorithm: Analysis and implementation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 881–892, 2000.

[2] F.D.A. De Carvalho, and Y. Lechevallier, "Dynamic clustering of interval-valued data based on adaptive quadratic distances," IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 39, no. 6, pp. 1295-1306, 2009.

[3] C. Ozturk, E. Hancer, and D. Karaboga, "Dynamic clustering with improved binary artificial bee colony algorithm," Applied Soft Computing, vol. 28, pp. 69–80, 2015.

[4] F.D.A. De Carvalho, and Y. Lechevallier, "Partitional clustering algorithms for symbolic interval data based on single adaptive distances," Pattern Recognition, vol. 42, no. 7, pp. 1223-1236, 2009.

[5] A.K. Jain, "Data clustering: 50 years beyond k-means," Pattern Recognition Letters, vol. 31, pp. 651-666, 2010.

[6] L.A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.

[7] E.H. Ruspini, "A new approach to clustering," *Information and Control*, vol. 15, no. 1, pp.22-32, 1969.

[8] J.C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact, well-separated clusters," *J. Cybernetics*, vol. 3, pp. 32-57, 1974.

[9] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, New York: Plenum Press, 1981.

[10] F. Hoppner, F. Klawonn, R. Kruse and T. Runkler, Fuzzy Cluster Analysis: Methods for Classification Data Analysis and Image Recognition, New York: Wiley, 1999.

[11] S.T Chang, K.P. Lu, M.S. Yang, "Fuzzy change-point algorithms for regression models," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 6, pp. 2343−2357, 2015.

[12] B.A. Pimentel, and R.M. de Souza, "A Generalized Multivariate Approach for Possibilistic Fuzzy C-Means Clustering," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 28, no. 6, pp. 893-916, 2018.

[13] E.H. Ruspini, J.C. Bezdek, J.M. Keller, "Fuzzy clustering: a historical perspective," *IEEE Computational Intelligence Magazine*, vol. 14, no. 1, pp. 45-55, 2019.

[14] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Systems*, vol. 1, pp. 98-110, 1993.

[15] A. Saha, S. Das, "Feature-weighted clustering with inner product induced norm based dissimilarity measures: an optimization perspective," *Machine Learning*, vol. 106,

pp. 951-992, 2017.

[16] J. Z. Huang, M. K. Ng, H. Rong and Z. Li, "Automated variable weighting in k-means type clustering," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 657-668, 2005.

[17] L. Jing, M. K. Ng and J. Z. Huang, "An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 8, pp. 1026-1041, 2007.

[18] D. M. Witten and R. Tibshirani, "A framework for feature selection in clustering," *Journal of the American Statistical Association*, vol. 105, no. 490, pp. 713-726, 2010.

[19] H. Frigui and O. Nasraoui, "Unsupervised learning of prototypes and attribute weights," *Pattern Recognition*, vol. 37, no. 3, pp. 567-581, 2004.

[20] W.L. Hung, M.S. Yang, D.H. Chen, "Bootstrapping approach to feature-weight selection in fuzzy c-means algorithms with an application in color image segmentation," *Pattern Recognition Letter*, vol. 29, pp. 1317–1325, 2004.

[21] X. Wang, Y. Wang and L. Wang, "Improving fuzzy c-means clustering based on feature weight learning," *Pattern Recognition Letters*, vol. 25, no. 10, pp. 1123-1132, 2004.

[22] A. Saha, S. Das, "Categorical fuzzy k-modes clustering with automated feature weight learning," *Neurocomputing*, vol. 166, pp. 422–435, 2015.

[23] M.S. Yang and Y. Nataliani, "A feature-reduction fuzzy clustering algorithm with feature-weighted entropy," *IEEE Trans. Fuzzy Systems*, vol. 26, pp 817–835, 2018.

[24] M. Barni, V. Cappellini, and A. Mecocci, "Comments on 'A Possibilistic Approach to Clustering," *IEEE Trans. Fuzzy Systems*, vol. 4, pp. 393-396, 1996.

[25] R. Krishnapuram and J. M. Keller, "The Possibilistic C-Means Algorithm: Insights and recommendations," *IEEE Trans. Fuzzy Systems*, vol. 4, pp. 385-393, 1996.

[26] M.S. Yang and C.Y Lai, "A robust automatic merging possibilistic clustering method," *IEEE Trans. Fuzzy Systems*, vol. 19, pp. 26-41, 2011.

[27] C.C. Hung, S. Kulkarni and B.C. Kuo, "A new-weighted fuzzy c-means clustering algorithm for remotely sensed image classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, 2011.

[28] B.A. Pimentel, and R.M. de Souza, "Multivariate Fuzzy c-means algorithms with weighting," *Neurocomputing*, vol. 174, pp. 946-965, 2016.

[29] Schneider, Adam (2000), "Weighted Possibilistic c-means Clustering Algorithms," Ninth IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2000 (Cat. No.00CH37063), pp. 176-180, 2000.

[30] Q. Zhang and Z. Chen, "A distributed weighted possibilistic c-means algorithm for clustering incomplete big sensor data," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 430814, 8 pages, 2014.

[31] Q. Zhang, L.T. Yang, A. Castiglione, Z. Chen, P. Li, "Secure weighted possibilistic c-means algorithm on cloud for clustering big data," *Information Sciences*, vol. 479, pp. 515-525, 2019.

[32] S. Bahrampour, B. Moshiri, K. Salahshoor, "Weighted and constrained possibilistic c-means clustering for online fault detection and isolation," *Appl. Intell.*, vol. 35, pp. 269-284, 2011.

[33] S.M. Hameed, M.B. Mohammed, "Spam filtering approach based on weighted version of possibilistic c-means," *Iraqi Journal of Science*, vol. 58, pp. 1112-1127, 2017.

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

[35] R. Xu, D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Networks*, vol. 16, no. 3, pp. 645-678, 2005.

[36] U. Fano, "Ionization yield of radiations. II. The fluctuations of the number of Ions," *Physical Review*, vol. 72, p. 26, 1947.

[37] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. American Statist. Assoc.*, vol. 66, pp. 846-850, 1971.

[38] P. Jaccard, "Distribution de la flore alpine dans le basin des Dranses et dans quelques regions voisines," *Bull. De la Soc. Vaudoise des Sci. Naturelles*, vol. 18, pp. 1008-1018, 2016.

[39] L. Hubert and P. Arabie, "Comparing partitions," *J. Classif.*, vol. 2, pp. 193–218, 1985.

[40] D. Steinley, "Properties of the Hubert-Arable adjusted Rand index," *Psychological methods*, vol. 9, no. 3, p.386, 2004.

[41] T.M. Cover, J.A. Thomas, Elements of Information Theory, New York: Wiley, 1991.

[42] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "Clustering validity checking methods: part II," SIGMOD Rec., vol. 31, no. 3, pp. 19–27, 2002.

[43] E. B. Fowlkes, C. L. Mallows, "A Method for Comparing Two Hierarchical Clusterings," *Journal of the American Statistical Association*. vol.78 (383): 553, 1983.

[44] X. Qiu, Y. Qiu, g. Feng & P. Li, "A sparse fuzzy c-means algorithm base on sparse clustering framework," *Nuerocomputing* vol. 157, 2015, pp 290-295

[45] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases, a huge collection of artificial and real-world data sets," 1998. http://archive.ics.uci.edu/ml/datasets.html

[46] R. A. Fisher "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.