IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 28, 2019, accepted August 6, 2019, date of publication August 9, 2019, date of current version August 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2934179

A Feature-Reduction Multi-View k-Means Clustering Algorithm

MIIN-SHEN YANG AND KRISTINA P. SINAGA
Department of Applied Mathematics, Chung Yuan Christian University, Chung-Li 32023, Taiwan

Corresponding author: Miin-Shen Yang (msyang@math.cycu.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 107-2118-M-033-002-MY2.

ABSTRACT The k-means clustering algorithm is the oldest and most known method in cluster analysis. It has been widely studied with various extensions and applied in a variety of substantive areas. Since internet, social network, and big data grow rapidly, multi-view data become more important. For analyzing multi-view data, various multi-view k-means clustering algorithms have been studied. However, most of multi-view k-means clustering algorithms in the literature cannot give feature reduction during clustering procedures. In general, there often exist irrelevant feature components in multi-view data sets that may cause bad performance for these clustering algorithms. There also exists high feature dimension in multi-view data sets so it is necessary to consider reducing its dimension for clustering algorithms. In this paper, a learning mechanism for the multi-view k-means algorithm to automatically compute individual feature weight is constructed. It can reduce these irrelevant feature components in each view. A new multi-view k-means objective function is firstly proposed for constructing the learning mechanism for feature weights in multi-view clustering. A schema for eliminating irrelevant feature(s) with small weight(s) is then considered for feature reduction. Therefore, a new type of multi-view k-means, called a feature-reduction multi-view k-means (FRMVK), is proposed. The computational complexity of FRMVK is also analyzed. Numerical and real data sets are used to compare FRMVK with other feature-weighted multi-view k-means algorithms. Experimental results and comparisons actually demonstrate the effectiveness and usefulness of the proposed FRMVK clustering algorithm.

INDEX TERMS Clustering, k-means, multi-view k-means, feature-reduction learning, feature-reduction multi-view k-means (FRMVK).

I. INTRODUCTION

Clustering is a useful tool for data analysis. It is a method for clustering a data set into groups with the most similarity in the same cluster and the most dissimilarity between different clusters [1], [2]. According to the statistical point of view, clustering methods may be divided as a probability model-based approach and a nonparametric approach. In nonparametric approaches, partitional methods are the most used. Partitional clustering methods suppose that the data set can be represented by finite cluster prototypes with their partitioning memberships. In partitional methods, the k-means algorithm is the oldest and most known method [3]–[6]. The k-means algorithm is generally used for (1-view) data. Co-clustering was first proposed by Dhillon [7] in 2001. However, co-clustering is used for only 2-view data. Afterward, Bickel and Scheffer [8] proposed multi-view clustering for handling multi-view data that are more than 2-views. Recently, internet, social network, and big data grow with high speeding and there are more and more multi-view data [9], [10]. In multi-view data, different views give different representations. For example, the same news can be told from different news sources. Web pages can be grouped based on both content and anchor text leading to hyperlinks. One image can be represented with different properties and different feature spaces, and one document may be translated into different languages. However, (one-view) data clustering (even, co-clustering) algorithms cannot handle these multi-view data. Extensions of clustering algorithms to multi-view clustering algorithms become important, especially for multi-view k-means clustering [11]–[15].

Multi-view learning can be divided into supervised, semisupervised and unsupervised learning approaches. For example, a supervised approach based on support vector machine was proposed in [16], and a semisupervised approach with both labeled and unlabeled training data to

The associate editor coordinating the review of this article and approving it for publication was Stavros Ntalampiras.

learn a discriminative feature representation was proposed in [17]. In this paper, we focus on the unsupervised learning approach for multi-view data analysis. However, most of multiview clustering algorithms consider all feature components of data in each view to be equally important [18], [19]. In most cases, there exist some irrelevant features that always affect clustering results and may produce incorrect clustering results. In this case, embedding feature reduction behavior in multi-view k-means for multi-view data can take advantage to improve the clustering performance. It is known that feature weights in one view are in the interval [0,1], so the more influence a feature is, the greater its weight should be. Feature-weighted techniques had been used for multi-view k-means clustering algorithms, such as simultaneous weighting on views and features (SWVF) [12] and weighted multi-view clustering with feature selection (WMCFS) [14]. Although these feature-weighted clustering algorithms may improve the performance of k-means for multi-view data, they do not consider feature reduction. In general, if there exist irrelevant features during clustering processes, the clustering algorithm must take more computational time and even yields incorrect clustering results, especially for multi-view data. Thus, a feature-reduction schema for multi-view k-means clustering algorithms becomes important.

In this paper we propose a novel feature reduction mechanism for multi-view k-means using the idea of Yang and Yessica [20]. In [20], they considered the fuzzy c-means (FCM) algorithm with a feature reduction for (1-view) data. Therefore, we propose the feature-reduction multi-view k-means (FRMVK) clustering algorithm that can automatically compute different feature weights and detects these unimportant (irrelevant) features in each view. Based on the feature-reduction mechanism in each view, the proposed FRMVK algorithm can solve the weakness in most multi-view k-means algorithms for multi-view data. The remainder of this paper is organized as follows. In Section II, we first review some related works with feature-weighted k-means for (1-view) data and then review these related multi-view k-means clustering algorithms. In Section III, we propose the FRMVK clustering algorithm with the learning schema for estimating the parameter values in the FRMVK objective function. To evaluate the performance of FRMVK, we use numerical and real data sets to compare FRMVK with the two leading algorithms: SWVF and WMCFS. These experiments and comparisons are made in Section IV. Finally, conclusions are stated in Section V.

II. RELATED WORKS

In this section, we introduce some notations and briefly review single-view and multi-view k-means clustering algorithms in the literature that use feature weights. Let $\mathbf{X} = \{x_1, \ldots, x_n\}$ be a data set in a d-dimensional Euclidean space $\mathbb{R}^d$ with $x_i = \{x_{ij}\}, i = 1, \cdots, n, j = 1, \cdots, d$ being the j-th feature component in the i-th data point. Let $U = [\mu_{ik}]_{n\times c}$, where μ_{ik} is a binary variable (i.e. $\mu_{ik} \in \{0, 1\}$) indicating if the data point x_i belongs to k-th cluster, $k = 1, \cdots, c$. Let $A = \{a_1, \ldots, a_c\}$ be the c cluster centers where $a_k = \{a_{kj}\}$, $k = 1, \cdots, c, j = 1, \cdots, d$ being the j-th feature component of the k-th cluster center. Thus, the k-means objective function is as follows:

$$J(U, A) = \sum_{i=1}^{n}\sum_{k=1}^{c}\sum_{j=1}^{d}\mu_{ik}(x_{ij} - a_{kj})^2$$

where $\|x_i - a_k\|$ is the Euclidean distance between the data point x_i and the cluster center a_k.

In Huang et al. [21], they first considered an extension of k-means by adding feature (variable) weights for data points, called the weighted k-means (WKM). Let $W = [w_{kj}]_{c\times d}$, where w_{kj} is the j-th feature weight in the k-th cluster center. The WKM objective function in Huang et al. [21] is as

$$J_{WKM}(U, A, W) = \sum_{k=1}^{c}\sum_{i=1}^{n}\sum_{j=1}^{d}\mu_{ik}(w_{kj})^{\beta}(x_{ij} - a_{kj})^2$$

where $\beta < 0$ or $\beta > 0$ is a power parameter for feature weights. They also considered to remove important variables by choosing variables with small weights for heart disease and Australian credit card data sets to obtain better results. The WKM algorithm improves the performance of the k-means algorithm with one additional step to compute feature weights during iterations. As indicated by the WKM objective function, it does depend on the exponent parameter β of feature weights. Different parameter settings will affect the WKM clustering results. On the other hand, weight discrimination ability for representing irrelevant features is not apparent in the WKM algorithm. Furthermore, Jing et al. [22] considered subspace clustering that is especially useful for high dimensional sparse data by using a feature-weighting approach. In Jing et al. [22], they proposed entropy-weighted k-means (EWKM) by adding weighted entropy term such that it can simultaneously minimize the within cluster dispersion and maximize the negative weighted entropy. Since feature weights represent the probability of a dimensional contributing to clustering results, it is used to determine subsets of important dimensions in each cluster. The EWKM objective function [22] is

$$J_{EWKM}(U, A, W) = \sum_{k=1}^{c}\sum_{i=1}^{n}\sum_{j=1}^{d}\mu_{ik}w_{kj}(x_{ij} - a_{kj})^2 + \gamma\sum_{k=1}^{c}\sum_{j=1}^{d}w_{kj}\log w_{kj}$$

where $\gamma \geq 0$ is a parameter. The parameter γ can control the size of feature weights in each cluster with the strength of feature weight entropy, but it needs to be estimated by users. If it does not have a good setting, the EWKM algorithm cannot get good clustering results. Jing et al. [22] carried out clustering on selected subspace instead of full data space by directly assigning zero weights to features with less information. They applied EWKM to high dimensional sparse data, such as text clustering and business transaction data, where many features have zero-dimension.

We know that co-clustering was first proposed by Dhillon [7] in 2001 for 2-view data sets. Bickel and Scheffer [8] in 2004 proposed multi-view clustering for handling multi-view data. Afterward, several multi-view k-means clustering algorithms had been proposed in the literature [10], [18], [23]. We next review some related works about multi-view k-means clustering. For a multi-view data set $\mathbf{X} = \{x_1, \ldots, x_n\}$ in a d-dimensional Euclidean space $\mathbb{R}^d$, let $x_i = \{x_i^h\}_{h=1}^s$ be the h-th view of the i-th data point with $x_i^h \in \mathbb{R}^{d_h}$, and let $x_i^h = \left\{x_{ij}^h\right\}_{j=1}^{d_h}$ be the j-th feature component of the h-th view in the i-th data point with $\sum_{h=1}^s d_h = d$. Let $U = [\mu_{ik}]_{n\times c}$, where μ_{ik} is a binary variable indicating if the data point x_i belongs to k-th cluster, $k = 1, \cdots, c$. Let $A = \{a_1, \ldots, a_c\}$ be the c cluster centers where $a_k = \{a_{kj}\}, j = 1, \cdots, d$ is the j-th feature component of the k-th cluster center. Let $W = [w_j]_{1\times d}$, where $w_j = \left\{w_j^h\right\}_{h=1}^s$ is the j-th feature weight in the h-th view. Let $V = [v_h]_{1\times s}$, where v_h is a weight for the h-th view. Thus, the k-means objective function for the multi-view data $\mathbf{X} = \{x_1, \ldots, x_n\}$ becomes

$$J(U, A) = \sum_{h=1}^{s}\sum_{i=1}^{n}\sum_{k=1}^{c}\sum_{j=1}^{d}\mu_{ik}(x_{ij}^h - a_{kj}^h)^2$$

For clustering multi-view data, Xu et al. [14] proposed a multi-view k-means clustering algorithm, called the weighted multi-view clustering with feature selection (WMCFS), by designing two weighting schemes for features and views such that the best view and the most representative feature subspace in each view can be selected for clustering. The WMCFS objective function in Xu et al. [14] is as follows:

$$J_{WMCFS}(U, A, W) = \sum_{h=1}^{s}(v_h)^{\alpha}\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik} \times \sum_{j=1}^{d_h}\left(w_j^h\left(x_{ij}^h - a_{kj}^h\right)\right)^2 + \beta\sum_{h}^{s}\sum_{j=1}^{d_h}\left(w_j^h\right)^2$$

where α is used to adjust the sparsity of view weights, ranging from 1 to 30. $\beta = 0.1$ is a parameter to control the sparsity of the feature weight. In their experiments, the iterations stop when the number of iterations reaches the maximum number of iterations with threshold $\varepsilon = 0.00001$. The real-world datasets they used in their experiments are Multiple Feature (MF), Reuters and Corel. Jiang et al. [12] proposed another multi-view k-means clustering algorithm via simultaneous weighting on views and features, called SWVF. In [12], they proposed weighting strategy where each feature for multi-view data is given bi-level weights to express its importance in feature level and view level, respectively. To implement the idea of simultaneous weighting, they embedded the proposed weighting method into k-means clustering algorithm to handle multi-view data. The SWVF objective function [12] is as follows:

$$J_{SWVF}(A, W, V, U) = \sum_{h=1}^{s}(v_h)^{\alpha}\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik} \times \sum_{j=1}^{d_h}(w_j)^{\beta}\left(x_{ij}^h - a_{kj}^h\right)^2$$

where α and β are two exponent parameters to respectively control view weights and feature weighs. They applied three real-world datasets to test the performance of SWVF, namely Amsterdam Library of Object Image (ALOI), Multiple Feature (MF) and 3-Sources. However, the SWVF clustering results are sensitive to the selection of parameters α and β. Different α and β will lead to different distributions of view and feature weights in which slight changes of them may yield quite different clustering results. Unlike the SWVF algorithm, the WMCFS algorithm may reduce sensitivity of parameter selection. The WMCFS makes use of the balancing parameter β to control the sparsity of feature weights in each view. The WMCFS clustering results are usually stable when the balancing parameter β is given a small value. Xu et al. [14] suggested the optimal performance can be achieved when the balancing parameter $\beta = 0.1$ for most data sets. However, the WMCFS algorithm is still sensitive to the selection of parameter α.

III. THE PROPOSED FEATURE-REDUCTION MULTI-VIEW K-MEANS ALGORITHM

In this section, we propose the feature-reduction multi-view k-means (FRMVK) clustering algorithm. Firstly, we give the FRMVK objective function, where the updating equations for FRMVK are derived by using the Lagrangian multiplier. Let $\mathbf{X} = \{x_1, \ldots, x_n\}$ be a multi-view data set in a d-dimensional Euclidean space $\mathbb{R}^d$ with $x_i = \{x_i^h\}_{h=1}^s$, $x_i^h \in \mathbb{R}^{d_h}$, and $x_i^h = \left\{x_{ij}^h\right\}_{j=1}^{d_h}$ with $\sum_{h=1}^s d_h = d$. Let $U = [\mu_{ik}]_{n\times c}$, where $\mu_{ik} = 1$ if the data point x_i belongs to the k-th cluster, and $\mu_{ik} = 0$ otherwise, i.e. $\mu_{ik} \in \{0, 1\}$. Let $A = \{a_1, \ldots, a_c\}$ be the c cluster centers with $a_k = \{a_{kj}\}$. Let $W = [w_j]_{1\times d}$, where $w_j = \left\{w_j^h\right\}_{h=1}^s$ is the j-th feature weight in the h-th view and let $V = [v_h]_{1\times s}$, where v_h is a weight for the h-th view. Since multi-view data may include some irrelevant feature components in each view, feature reduction for multi-view data is important. A novel schema with feature-weighted entropy in each view is proposed to have feature reduction for multi-view data. In this schema, each feature in each view has its own feature weight that will be updated at each iteration. After some learning procedures, feature(s) with small weights(s) in each view will be eliminated. The proposed FRMVK objective function is as follows:

$$J(U,A,V,W)=\sum_{h=1}^{s}(v_h)^{\alpha}\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\sum_{j=1}^{d_h}w_j^h\delta_j^h\left(x_{ij}^h-a_{kj}^h\right)^2+\frac{n}{d_h}\sum_{h=1}^{s}\sum_{j=1}^{d_h}w_j^h\ln\delta_j^hw_j^h \quad (1)$$

subject to $\sum_{j=1}^{d_h}w_j^h=1, w_j\in[0,1]$ and $\sum_{h=1}^{s}v_h=1$, $v_h\in[0,1]$. Note that δ_j^h is a balance parameter to control the feature weights of the k-th cluster in each view; $\alpha>0$ is the exponent for the view weights.

The FRMVK clustering algorithm is to minimize Eq. (1) with its constraints. Since we have to solve the variables a_{kj}^h, μ_{ik}, v_h and w_j^h in Eq. (1), the FRMVK updating equations can be obtained using the four minimization steps, where one of the four variables is updated by fixing the other three variables. The updating equation for $\mu_{ik}$is to minimize the FRMVK objective function $J(U,A,V,W)$ of Eq. (1) w.r.t. μ_{ik} that is equivalent to minimizing $\sum_{h=1}^{s}(v_h)^{\alpha}\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\sum_{j=1}^{d_h}w_j^h\delta_j^h\left(x_{ij}^h-a_{kj}^h\right)^2$ w.r.t. μ_{ik}. Let $d_{ik}(x_i,a_k)=\sum_{h=1}^{s}(v_h)^{\alpha}\sum_{j=1}^{d_h}w_j^h\delta_j^h\left(x_{ij}^h-a_{kj}^h\right)^2$. Thus, we can obtain the following updating equation for μ_{ik}:

$$\mu_{ik}=\begin{cases}1 \text{ if } d_{ik}(x_i,a_k)=\min_{1\le q\le c}d_{iq}(x_i,a_q)\\ 0 \text{ otherwise}\end{cases} \quad (2)$$

By taking the partial derivative of the FRMVK objective function $J(U,A,V,W)$ of Eq. (1) w.r.t. a_{kj}^h and setting them to be zero, we obtain the equation $\frac{\partial J}{\partial a_{kj}^h}=-2(v_h)^{\alpha}\sum_{i=1}^{n}\mu_{ik}w_j^h\delta_j^h\left(x_{ij}^h-a_{kj}^h\right)=0$,and then we have that $\sum_{i=1}^{n}\mu_{ik}\left(x_{ij}^h-a_{kj}^h\right)=0$. Thus, the updating equation for a_{kj}^h can be obtained as

$$a_{kj}^h=\sum_{i=1}^{n}\mu_{ik}x_{ij}^h\Big/\sum_{i=1}^{n}\mu_{ik} \quad (3)$$

where a_{kj}^h is the j-th feature component of the k-th cluster center in the h-th view.

For solving the optimization problem of the FRMVK objective function $J(U,A,V,W)$ of Eq. (1) with the constraints w.r.t. $v_h$and w_j^h, the Lagrangian multiplier needs to be used. The Lagrangian for $J(U,A,V,W)$ is given as

$$\begin{aligned}\tilde{J}&(U,A,V,W)\\&=\sum_{h=1}^{s}(v_h)^{\alpha}\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\sum_{j=1}^{d_h}w_j^h\delta_j^h\left(x_{ij}^h-a_{kj}^h\right)^2\\&\quad+\frac{n}{d_h}\sum_{h=1}^{s}\sum_{j=1}^{d_h}w_j^h\ln\delta_j^hw_j^h\\&\quad-\lambda_1\left(\sum_{j=1}^{d_h}w_j^h-1\right)-\lambda_2\left(\sum_{h=1}^{s}v_h-1\right)\end{aligned} \quad (4)$$

By taking the partial derivative of the Lagrangian $\tilde{J}$ of Eq. (4) w.r.t v_h and setting them to be zero, we obtain the equation $\frac{\partial\tilde{J}}{\partial v_h}=\alpha(v_h)^{\alpha-1}\sum_{k=1}^{c}\sum_{j=1}^{n}\mu_{ik}\sum_{j=1}^{d_h}w_j^h\delta_j^h\left(x_{ij}^h-a_{kj}^h\right)^2-\lambda_2=0$. Thus, we have $v_h=\left(\lambda_2\Big/\alpha\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\sum_{j=1}^{d_h}\delta_j^hw_j^h\left(x_{ij}^h-a_{kj}^h\right)^2\right)^{\frac{1}{\alpha-1}}$. Since $\sum_{h=1}^{s}v_h=1$, we get $\lambda_2=\left(\sum_{h'=1}^{s}\left(\alpha\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\sum_{j=1}^{d_{h'}}\delta_j^{h'}w_j^{h'}\left(x_{ij}^{h'}-a_{kj}^{h'}\right)^2\right)^{\frac{-1}{\alpha-1}}\right)^{-1}$. Thus, the updating equation for v_h can be obtained as

$$v_h=\left(\sum_{h'=1}^{s}\left(\frac{\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\sum_{j=1}^{d_h}\delta_j^hw_j^h\left(x_{ij}^h-a_{kj}^h\right)^2}{\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\sum_{j=1}^{d_{h'}}\delta_j^{h'}w_j^{h'}\left(x_{ij}^{h'}-a_{kj}^{h'}\right)^2}\right)^{\frac{1}{\alpha-1}}\right)^{-1} \quad (5)$$

By taking the partial derivative of the Lagrangian of Eq. (4) w.r.t w_j^h and setting them to be zero, we obtain the equation $\frac{\partial\tilde{J}}{\partial w_j^h}=v_h^{\alpha}\delta_j^h\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\left(x_{ij}^h-a_{kj}^h\right)^2+\frac{n}{d_h}\left(\ln\delta_j^hw_j^h+1\right)-\lambda_1=0$. Thus, we have $\ln\delta_j^hw_j^h=\frac{\left(-d_hv_h^{\alpha}\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\delta_j^h\left(x_{ij}^h-a_{kj}^h\right)^2-n+d_h\lambda_1\right)}{n}$ and $w_j^h=\frac{1\Big/\delta_j^h\exp\left(-d_hv_h^{\alpha}\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\delta_j^h\left(x_{ij}^h-a_{kj}^h\right)^2\Big/n\right)}{\exp(1-d_h\lambda_1/n)}$. Since $\sum_{j=1}^{d_h}w_j^h=1$, we get $\exp\left(n-\lambda_1/n\right)=\sum_{j'=1}^{d_h}\frac{1}{\delta_{j'}^h}\exp\left(-d_h(v_h)^{\alpha}\delta_{j'}^h\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\left(x_{ij'}^h-a_{kj'}^h\right)^2\Big/n\right)$. Thus, the updating equation for w_j^h can be obtained as

$$w_j^h=\frac{\frac{1}{\delta_j^h}\exp\left(-d_h(v_h)^{\alpha}\delta_j^h\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\left(x_{ij}^h-a_{kj}^h\right)^2\Big/n\right)}{\sum_{j'=1}^{d_h}\frac{1}{\delta_{j'}^h}\exp\left(-d_h(v_h)^{\alpha}\delta_{j'}^h\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\left(x_{ij'}^h-a_{kj'}^h\right)^2\Big/n\right)} \quad (6)$$

It is necessary to explain why we use $n/d_h$to control the effect of the entropy term $\sum_{h=1}^{s}\sum_{j=1}^{d_h}w_j^h\ln\delta_j^hw_j^h$ in the proposed FRMVK objective function, where it is also appeared in the updating equation (6) for feature weights. In Eq. (6), if the term $\exp((v_h)^{\alpha}\delta_j^h\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}(x_{ij}^h-a_{kj}^h)^2)$ is too large, then the numerator in Eq. (6) will become too small as close to zero. We need to avoid this case for preventing too many features to be discarded during updating steps. On the other hand, if the term $\exp((v_h)^{\alpha}\delta_j^h\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}(x_{ij}^h-a_{kj}^h)^2)$ is too small, then the numerator in Eq. (6) will be large as close to one so that it is difficult for the feature(s) to be discarded during updating steps. This case also needs to be avoided. In this sense, we need to put a suitable constant to control the effect. In the FRMVK clustering algorithm, one goal is to cluster the n data points into c clusters. The numbers n and d_h are the two commonly given constants. We use the constant n/d_h to control the effect of the term $\exp((v_h)^\alpha \delta_j^h \sum_{k=1}^{c}\sum_{i=1}^{n} \mu_{ik}(x_{ij}^h - a_{kj}^h)^2)$.

Another problem is how to estimate the value of δ_j^h in Eqs. (2), (5) and (6). There are two δ_j^h terms in the FRMVK objective function. The first is the sum of feature-weighted distance between data points and cluster centers in each view, which is minimized when the distance between points and centers is small. The second is the feature-weight entropy. Because the δ_j^h in the first and second terms of the FRMVK objective function are used to control the variants of feature weights in each view, the choice of δ_j^h is very important. We next propose a learning schema for estimating δ_j^h. Example 1 is used to demonstrate the proposed learning schema.

Example 1: In this example, we generate a multi-view 2-cluster data set that has 1000 data points from a 2-component Gaussian mixture distribution $\sum_{k=1}^{2}(1/2)\, N(u_k^{(h)}, \sum_k^{(h)})$, where h =1 and 2 are the two views. The means $u_k^{(1)}$ for the view 1 are $\begin{pmatrix}2 & 2\end{pmatrix}$ and $\begin{pmatrix}5 & 5\end{pmatrix}$. The means $u_k^{(2)}$ for the view 2 are $\begin{pmatrix}-6 & 6\end{pmatrix}$ and $\begin{pmatrix}2 & 2\end{pmatrix}$. The covariance matrices for the two views are $\sum_1^{(1)} = \begin{pmatrix} 0.9 & -0.0255 \\ -0.0255 & 0.9 \end{pmatrix}$, $\sum_2^{(1)} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.3 \end{pmatrix}$, $\sum_1^{(2)} = \begin{pmatrix} 1.5 & -0.4 \\ -0.4 & 1.5 \end{pmatrix}$ and $\sum_2^{(2)} = \begin{pmatrix} 1 & 0.4 \\ 0.4 & 0.7 \end{pmatrix}$. These $x_2^{(1)}$ and $x_3^{(1)}$ are the coordinates for the view 1, and $x_2^{(2)}$ and $x_3^{(2)}$ are the coordinates for the view 2, as shown in Fig. 1(a) and (b), respectively. For demonstrating the feature-reduction schema, we add a feature, coordinated by $x_1^{(1)}$, for the view 1, and another feature, coordinated by $x_1^{(2)}$, for the view 2 where they are generated from a uniform distribution over the interval $[-4, -2]$. Obviously, the uniform distribution will stretch the 2-cluster data over the interval $[-4, -2]$ in each view, as shown in Fig. 1(c) and (d). Of course, the features generated from the Gaussian mixture are important, but the features generated from the uniform distribution are unimportant.

We next use the idea of coefficient of variance (CV) in statistic that is defined as $CV = \sigma/\mu$ or $CV = \sigma/|\mu|$. The reciprocal of CV is also known as signal-to-noise ratio (SNR) that is widely used in quality engineering to evaluate the performance of a system. SNR is defined as the ratio of average received signal value to standard deviation of noise background, i.e. $SNR = \mu/\sigma$ (see [24]). Furthermore, in physics, Fano factor (FF), which can be seen as a similar CV, had been proposed and defined as $FF = \sigma^2/\mu$ (see [25]). 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 (M-V-R). However, the parameter δ_j^h

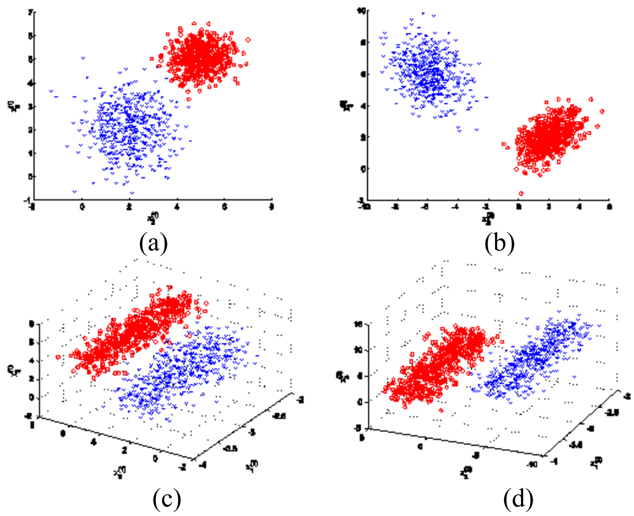


FIGURE 1. **The 2-cluster data set for (a) view 1; (b) view 2; The 2-cluster data set by adding one feature generated from a uniform distribution for (c) view 1; (d) view 2.**

must be positive. Therefore, the (absolute mean)-to-variance ratio (AM-V-R), i.e. $|\mu|/\sigma^2$, can be nature as an estimate for δ_j^h. However, to estimate δ_j^h, we have to pick a number as a better estimate for reducing unimportant features in each view. We may also take the square root of the absolute mean in the AM-V-R, i.e. $\sqrt{|\mu|}/\sigma^2$. We call it the (square root of absolute mean)-to-variance ratio (SRAM-V-R). We think that the impact of the SRAM-V-R$\sqrt{|\mu|}/\sigma^2$ on δ_j^h will give more effect for feature weights.

We use the three factors, FF, AM-V-R, and SRAM-V-R for estimating the parameter δ_j^h with their respectively obtained feature weight values of w_j^h using the data set in example 1 where the values of δ_j^h will affect the obtained feature weights. These values of δ_j^h and w_j^h are shown in Table 1. As can be seen, SRAM-V-R can produce smaller feature weights for $x_1^{(1)}$ and $x_1^{(2)}$, while FF produces smaller feature weights for $x_2^{(1)}, x_3^{(1)}, x_2^{(2)}$ and $x_3^{(2)}$ and AM-V-R produces smaller feature weights for $x_1^{(2)}$and $x_3^{(2)}$. It is found that SRAM-V-R can be fitted as an estimate of the parameter δ_j^h. Therefore, the estimator of δ_j^h is used as follows:

$$\delta_j^h = \left(\frac{\sqrt{\left| mean\left(x_j^h\right) \right|}}{var\left(x_j^h\right)} \right) \tag{7}$$

To create a feature-reduction schema in the proposed FRMVK algorithm, we need to select the irrelevant features via automatically adjust the feature weights in each view during clustering processes. In our construction, we use a threshold to determine which feature(s) will be selected and discarded. It is known that the data set has n data points that belong to the h views in which each view has d_h feature components. The data set is represented by the h different views with $x_i^h \in R^{d_h}$ where d_h is the dimension of the h-th view with

TABLE 1. Comparison of FF, AM-V-R, and SRAM-V-R for the 2-cluster data set of example 1.

		View 1			View 2		
		$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$
FF	δ_j^h	0.108	0.830	0.804	0.115	9. 367	1.298
	w_j^h	0.999	0.000	0.000	1.000	0.000	0.000
AM-V-R	δ_j^h	9.230	1.205	1.244	8.697	0.107	0.771
	w_j^h	0.167	0.442	0.392	0.000	0.999	0.000
SRAM-V-R	δ_j^h	5.333	0.641	0.657	5.024	0.078	0.390
	w_j^h	0.060	0.477	0.462	0.003	0.977	0.020

The FRMVK Algorithm

Input: Dataset $\mathbf{X} = \{x_1, \ldots, x_n\}$ with $x_i = \left\{x_i^h\right\}_{h=1}^s$ and $x_i^h = \left\{x_{ij}^h\right\}_{j=1}^{d_h}$, number c of cluster, α, and $\varepsilon > 0$.

Output: a_{kj}^h, μ_{ik}^h, v_h and w_j^h.

Initialization: Randomly generate initial U^h, initialize feature weight $W^{h^{(0)}} = [w_j^h]_{1\times d_h}$ (user may define $w_j^h = 1/d_h \forall j$), initialize view weight $V^{(0)} = [v_h]_{1\times s}$ (user may define $v_h = 1/s\forall h$), and set $t = 1$.

Step1: Calculate δ_j^h by Eq. (7).

Step 2: Compute the cluster center $A^{h^{(t)}}$ using $U^{(t-1)}$ by Eq. (3).

Step 3: Update membership matrix $U^{(t)}$ using $\delta_j^h, A^{h^{(t)}}, V^{(t-1)}$ and $\mathrm{W}^{h^{(t-1)}}$ by Eq. (2)

Step 4: Update the feature weight $W^{h^{(t)}}$ using $\delta_j^h, A^{h^{(t)}}, U^{(t)}$ and $V^{(t-1)}$ by Eq. (6)

Step 5: Update the view weight $V^{(t)}$ using $\delta_j^h, A^{h^{(t)}}, U^{(t)}$ and $W^{h^{(t)}}$ by Eq. (5)

Step 6: Discard total d_r number of these j feature components for $W^{h^{(t)}}$, if $W^{h^{(t)}} = 1/\sqrt{nd_h}$, and set $d^{new} = D - d_r$

Step 7: Adjust $W^{h^{(t)}}$ by Eq. (8).

Step 8: If $\left|\left\|W^{h^{(t)}}\right\| - \left\|W^{h^{(t-1)}}\right\|\right| < \varepsilon$, then stop;
Else set $t = t + 1$, $d = d^{(new)}$ and go back to Step 1.

$\sum_{j=1}^{d_h} w_j^h = 1$. If d_h is large, then the threshold for feature reduction in the h-th view is intuitively chosen as $1/d_h$. However, the proposed feature-reduction algorithm must fit for most multi-view data sets, even for small d_h. In this sense, the data number n should be considered as another factor. It is known that $1\Big/d_h = 1\Big/\sqrt{d_h^2} = 1/\sqrt{d_h d_h}$. For a balance between small and large d_h, we replace one d_h with n so that it becomes $1/\sqrt{nd_h}$. Therefore, we consider $1/\sqrt{nd_h}$ as a suitable threshold for discarding these irrelevant features in the h-th view. After these irrelevant features in the view h are discarded, to retain the constraint $\sum_{j=1}^{d_h} w_j^h = 1$, these feature weights w_j^h need to be adjusted by

$$(w_j^h)' = w_j^h \Big/ \sum_{p=1}^{d_h(new)} w_p^h \tag{8}$$

Thus, the proposed FRMVK algorithm is summarized as follows:

In the proposed FRMVK algorithm, we first give the number c of clusters and assign the values of exponential parameter α. We need to initialize feature weights w_j^h and view weights v_h, but simply initialized with $w_j^h = 1/d_h \forall j$ and $v_h = 1/s\forall h$. For stopping the algorithm, we set the iterative process of the weighted sum of the intra-features weight variances between the t-th and the (t-1)-th iterations as MJ(t) which can be computed as $MJ(t) = \left|\left\|W^{h^{(t)}}\right\| - \left\|W^{h^{(t-1)}}\right\|\right|$, where $W^{h^{(t)}}$ denotes the updated iteration of feature weights in each view and $W^{h^{(t-1)}}$ is the feature weights in the (t-1)-th iterations. $MJ(t) \geq 0$ means the sum of intra-feature weight variances in h-th view distances updated in each step of iterations is strictly decreasing. In our experiments, the iteration stops when the gap of the sum of intra-feature weight variances distances between the two consecutive iterations is less than the threshold ε.

IV. EXPERIMENTAL RESULTS AND COMPARISONS

In this section, three synthetic and four real data sets are used to illustrate the performance of the proposed FRMVK algorithm. Among the four real data sets, the first one is the text data set, the second is the image data set, and the last two are the images of CALTECH-101 data sets. Comparison of the proposed FRMVK algorithm with SWVF [12] and WMCFS [14] is also made. For the experimental comparisons, all algorithms use the same initial cluster center assignments, the same initial feature weights, and the same initial view weights. Accuracy rate (AR) and Rand Index (RI) [26] are used as the criteria for the performance evaluation. Note that all the three algorithms of FRMVK, SWVF, and WMCFS have the same parameter α that is used to control view weights, so the parameter α must have the same given values in all experimental comparisons.

Example 1 (Cont.): We continue Example 1 by implementing the FRMVK algorithm for the 2-cluster data set with equal feature weights and equal view weights as the initialization $W^{h^{(0)}}$. After two iterations, FRMVK clearly demonstrates $x_1^{(1)}$ and$x_1^{(2)}$ as unimportant features. This feature reduction behavior is shown in Table 2, where the clustering result from FRMVK is with $AR = 1.00$.

Example 2: In this example, we use a more complicated synthetic data set with manifold shapes that is the 2-HalfMoon+1-block, as shown in Fig. 2(a) and (b) for view 1 and view 2, respectively. The manifold data set has 3 clusters and 900 data points for which the 600 data points are for the HalfMoon pattern and the 300 data points are for 1-block shape. Fig. 2(a) is the visualization of view-1 with the coordinates $x_1^{(1)}$ and $x_2^{(1)}$, while Fig. 2(b) is the

TABLE 2. Feature reduction behavior by FRMVK for the 2-cluster data set of example 1.

	Feature weights					
	$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$
Iteration 1	0.060	0.477	0.462	0.060	0.977	0.020
Iteration 2	0.060	0.477	0.462	0.060	0.980	0.020
Iteration 3	-	0.476	0.524	-	0.943	0.057
Iteration 4	-	0.475	0.525	-	0.942	0.058
Iteration 5	-	0.475	0.525	-	0.942	0.058

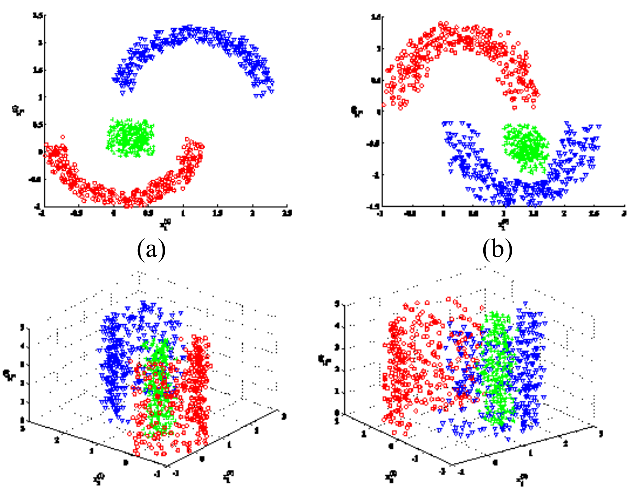


FIGURE 2. (a) 2-HalfMoon+1-block of View-1; (b) 2-HalfMoon+1-block of View-2; (c) Combination of features in Fig. 2(a) with another feature generated from a uniform distribution; (d) Combination of features in Fig. 2(b) with another feature generated from the uniform distribution.

visualization of view-2 with the coordinates $x_1^{(2)}$ and $x_2^{(2)}$. Furthermore, we design another irrelevant feature in each view with uniform distributions, where we generate $x_3^{(1)}$ and $x_3^{(2)}$ from the uniform distributions over intervals [0,5], respectively. Fig. 2(c) is the visualization of view-1 with the coordinates$x_1^{(1)}$, $x_2^{(1)}$ and $x_3^{(1)}$, while Fig. 2(d) is the visualization of view-2 with the coordinates $x_1^{(2)}$, $x_2^{(2)}$and $x_3^{(2)}$.

This complicated manifold data set is a challenge for most clustering methods. We implement the FRMVK algorithm for this data set. FRMVK is able to adaptively identify these important features of each view, in which it demonstrates that the feature weighting mechanism in FRMVK can enhance the algorithm stability. The feature reduction behavior is shown in Table 3, where the unimportant features $x_3^{(1)}$ and $x_3^{(2)}$ from uniform distributions are reduced after iteration 2. The clustering results from FRMVK in each view are shown in Fig. 3(a) and (b). Furthermore, we implement WMCFS and SWVF for the 2-Half-Moon+1-block data set. The clustering results from WMCFS in each view are respectively shown in Fig. 3(c) and (d), and the clustering results from SWVF in each view are respectively shown in Fig. 3(e) and (f). The average ARs for the FRMVK, WMCFS, and SWVF algorithms are as 0.965, 0.554 and 0.878, respectively. As can

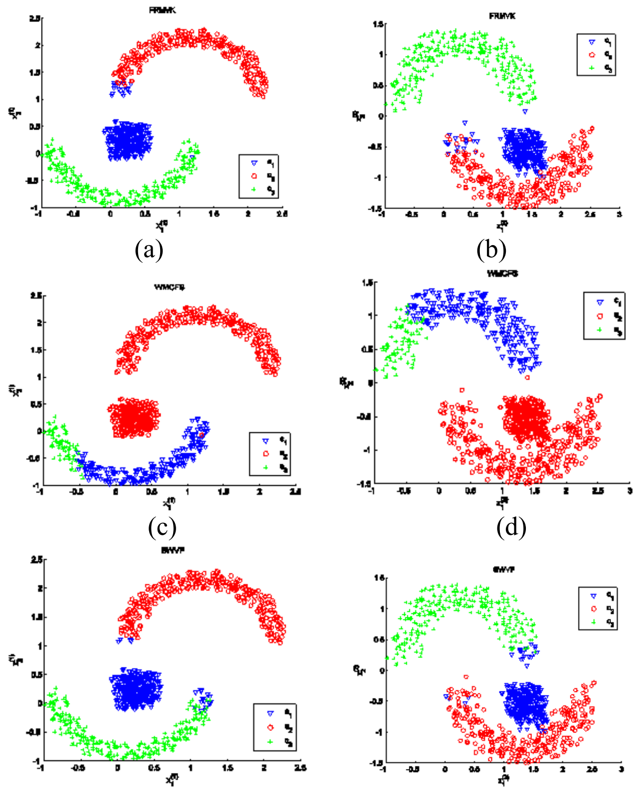


FIGURE 3. Clustering results of view 1 and view 2 in 2-HalfMoon+1-block by FRMVK; (c)-(d) Clustering results of view 1 and view 2 in 2-HalfMoon+1-block by WMCFS; (e)-(f) Clustering results of view 1 and view 2 in 2-HalfMoon+1-block by SWVF.

TABLE 3. Feature reduction behavior by FRMVK for the manifold data set of example 2.

	Feature weights					
	$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$
Iteration 1	0.386	0.573	0.041	0.095	0.851	0.055
Iteration 2	0.442	0.531	0.026	0.183	0.817	-
Iteration 3	0.210	0.790	-	0.215	0.785	-
Iteration 4	0.227	0.773	-	0.182	0.818	-
Iteration 5	0.219	0.781	-	0.160	0.840	-
Iteration 6	0.217	0.783	-	0.150	0.850	-
Iteration 7	0.221	0.779	-	0.145	0.855	-

be seen, the proposed FRMVK actually presents better clustering results than WMCFS, and SWVF.

Example 3: In this example, a numerical three-view data set with 3 clusters and 5 feature components is considered. The data points in each view are generated from a 3-component 5-variate Gaussian mixture model (GMM) where their mixing proportions, means and variance matrices are shown in Table 4. Furthermore, two irrelevant features in the view 2 with uniform distributions are added. For the feature component $x_6^{(2)}$, 307 data points generated from the uniform distribution over the interval [0, 12] are added in the first component; 332 data points generated from the

TABLE 4. Mixing proportions, means and variance matrices of the GMM for the data set of example 3.

Views	Mixing proportions	Mean values	Variance matrices
View 1	$\alpha_1 = 0.30$ $\alpha_2 = 0.35$ $\alpha_3 = 0.35$	$\mu_1^{(1)} = (1 \quad 0.1 \quad 2 \quad 1 \quad 0.5)$ $\mu_2^{(1)} = (5 \quad 3 \quad 0.5 \quad 0 \quad 2)$ $\mu_3^{(1)} = (1 \quad 0.1 \quad 5 \quad 1 \quad 0.2)$	$\Sigma_1^{(1)} = \begin{bmatrix} 4 & 0.1 & 0 & 0 & 0 \\ 0.1 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \Sigma_2^{(1)} = \begin{bmatrix} 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 \\ 0.5 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ $\Sigma_3^{(1)} = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$
View 2	$\alpha_1 = 0.30$ $\alpha_2 = 0.35$ $\alpha_3 = 0.35$	$\mu_1^{(2)} = (5 \quad 3 \quad 0.5 \quad 0 \quad 2)$ $\mu_2^{(2)} = (1 \quad 0.1 \quad 5 \quad 1 \quad 0.2)$ $\mu_3^{(2)} = (1 \quad 0.1 \quad 2 \quad 1 \quad 0.5)$	$\Sigma_1^{(2)} = \begin{bmatrix} 4 & 0.1 & 0 & 0 & 0 \\ 0.1 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \Sigma_2^{(2)} = \begin{bmatrix} 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 \\ 0.5 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ $\Sigma_3^{(2)} = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$
View 3	$\alpha_1 = 0.30$ $\alpha_3 = 0.35$ $\alpha_3 = 0.35$	$\mu_1^{(3)} = (1 \quad 0.1 \quad 5 \quad 1 \quad 0.2)$ $\mu_2^{(3)} = (1 \quad 0.1 \quad 2 \quad 1 \quad 0.5)$ $\mu_3^{(3)} = (5 \quad 3 \quad 0.5 \quad 0 \quad 2)$	$\Sigma_1^{(3)} = \begin{bmatrix} 4 & 0.1 & 0 & 0 & 0 \\ 0.1 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \Sigma_2^{(3)} = \begin{bmatrix} 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 \\ 0.5 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ $\Sigma_3^{(3)} = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$

uniform distribution over interval [0,18] are added in the second component; 361 data points generated from the uniform distribution over the interval [0,20] are added in the third component. For the feature component $x_7^{(2)}$, 307 data points generated from the uniform distribution over the interval [0, 14] are added in the first component; 332 data points generated from the uniform distribution over interval [0,16] are added in the second component; 361 data points generated from the uniform distribution over the interval [0,18] are added in the third component. We implement FRMVK, WMCFS and SWVF for the multi-view data set. It is found that FRMVK, SWVF and WMCFS get average ARs with 0.856, 0.720 and 0.444, respectively. As can be seen, the proposed FRMVK gets the best performance. On the other hand, FRMVK also performs feature reduction where the feature reduction behaviors of FRMVK are shown in Table 5. From Table 5, FRMVK clearly demonstrates that the sixth and seventh features in the view 2 are unimportant features. It is good because the sixth $x_6^{(2)}$ and seventh $x_7^{(2)}$ features are originally generated from uniform distributions that are unimportant features in the view 2. Furthermore, ARs and RIs from the proposed FRMVK after different iterations are also shown in Table 5. It is seen that the proposed FRMVK increases the values of ARs and RIs after each iteration. As can be seen, the proposed FRMVK takes feature reduction behavior that successfully detects unimportant features and also improves clustering performance.

In next example, we use four real multi-view data sets from UCI repository [27], that include Image Segmentation data set, Multiple Features data set, and two image data sets known as Caltech-7/20 [10], for investigating the performance of the proposed FRMVK algorithm. Comparisons of the proposed FRMVK with SWVF and WMCFS are also made.

Example 4: Detailed information for the real data sets, Image Segmentation (IS) [27], Multiple Features (MF) [27], and Caltech-7/20 [28] is shown in Table 6. Note that the exponent α for view weights are all appeared in the proposed FRMVK, SWVF and WMCFS where, in SWVF [12] and

TABLE 5. Feature reduction behavior by FRMVK for the data set of example 3.

	$x_i^{(h)}$	w_v					
		Initialization	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
v_1	$x_1^{(1)}$	1/5	0.091	0.091	0.147	0.112	0.105
	$x_2^{(1)}$	1/5	0.326	0.326	0.243	0.191	0.186
	$x_3^{(1)}$	1/5	0.084	0.084	0.378	0.512	0.536
	$x_4^{(1)}$	1/5	0.120	0.120	0.054	0.044	0.041
	$x_5^{(1)}$	1/5	0.380	0.380	0.178	0.142	0.132
v_2	$x_1^{(2)}$	1/7	0.061	0.061	0.128	0.084	0.068
	$x_2^{(2)}$	1/7	0.328	0.328	0.310	0.219	0.179
	$x_3^{(2)}$	1/7	0.040	0.040	0.293	0.548	0.618
	$x_4^{(2)}$	1/7	0.151	0.151	0.062	0.047	0.039
	$x_5^{(2)}$	1/7	0.419	0.420	0.208	0.102	0.096
	$x_6^{(2)}$	1/7	0.000	-	-	-	-
	$x_7^{(2)}$	1/7	0.001	-	-	-	-
v_3	$x_1^{(3)}$	1/5	0.088	0.088	0.128	0.124	0.106
	$x_2^{(3)}$	1/5	0.293	0.293	0.432	0.337	0.300
	$x_3^{(3)}$	1/5	0.097	0.097	0.222	0.409	0.484
	$x_4^{(3)}$	1/5	0.154	0.154	0.093	0.060	0.050
	$x_5^{(3)}$	1/5	0.369	0.369	0.126	0.079	0.059
AR		-	-	0.571	0.735	0.856	0.858
RI		-	-	0.707	0.746	0.843	0.846

TABLE 6. Characteristics for the real data sets.

Data	View	Composition of Each View	Dimension	Cluster	Size
IS	Shape view	9 features for the shape information of 7 images	9	7	2310
	RGB view	10 features for the RGB values of 7 images	10		
MF	Mfeat-fou view	76 Fourier coefficients of the character shapes	76	10	2000
	Mfeat-fac view	216 profile correlations	216		
	Mfeat-kar view	64 Karhunen-Love correlations	64		
	Mfeat-pix view	240 pixel averages in 2 x 3 windows	240		
	Mfeat-zer view	47 Zernike moments	47		
	Mfeat-mor view	6 morphological variables	6		
Caltech-7/20	Gabor	48 dimension Gabor feature	48	7/20	1474/2386
	WM	40 dimension wavelet moments	40		
	CENTRIST	254 dimension of census transform histogram	254		
	HOG	1984 dimension of the histogram of oriented gradients	1984		
	GIST	512 dimension of the spatial envelope of the image	512		
	LBP	928 dimension of local binary patterns	928		

WMCFS [14], they had shown that the exponent α for view weights actually has the influence on clustering performance of SWVF and WMCFS. In this example, we first explore the influence of the exponent α on clustering performance and feature reduction behavior of FRMVK. We analyze the effect of the exponent α using accuracy rate (AR) and rand index (RI). On the other hand, both WMCFS and SWVF need extra setting for the parameter β, except the exponent α for view weights. Therefore, we also make comparisons of FRMVK with both WMCFS and SWVF algorithms under

TABLE 7. Worst/Average/Best of AR and RI from algorithms with different α on IS and MF data.

Method	α	IS		MF	
		AR	RI	AR	RI
WMCFS	2	0.241/0.335/0.426	0.516/0.692/0.767	0.100/0.148/0.168	0.245/0.449/0.523
	3	0.119/0.228/0.360	0.408/0.516/0.749	0.151/0.156/0.159	0.368/0.478/0.505
	4	0.119/0.188/0.323	0.401/0.468/0.583	0.170/0.204/0.310	0.572/0.592/0.698
	5	0.119/0.190/0.360	0.403/0.473/0.762	0.328/0.456/0.569	0.681/0.756/0.839
	6	0.119/0.228/0.360	0.320/0.497/0.764	0.402/0.589/0.710	0.711/0.839/0.895
	7	0.180/0.275/0.366	0.399/0.540/0.727	0.632/0.746/0.776	0.887/0.926/0.933
	8	0.205/0.312/0.379	0.413/0.609/0.749	0.654/0.723/0.791	0.919/0.930/0.942
	9	0.209/0.346/0.439	0.394/0.678/0.766	0.664/0.718/0.795	0.927/0.935/0.948
	10	0.235/0.382/0.537	0.378/0.706/0.800	0.556/0.693/0.795	0.912/0.933/0.948
SWVF	2	0.143/0.362/0.542	0.143/0.642/0.851	0.306/0.370/0.439	0.849/0.862/0.874
	3	0.143/0.459/0.572	0.143/0.756/0.860	0.560/0.656/0.727	0.899/0.925/0.938
	4	0.143/0.482/0.613	0.143/0.800/0.864	0.701/0.740/0.852	0.940/0.944/0.965
	5	0.360/0.495/0.608	0.731/0.838/0.866	0.706/0.751/0.851	0.941/***0.948/0.966
	6	0.143/0.480/0.584	0.143/0.798/0.869	0.610/0.733/0.850	0.923/0.946/0.966
	7	0.360/0.501/0.615	0.731/0.838/0.864	0.100/0.656/0.844	0.100/0.841/0.966
	8	0.360/0.504/0.615	0.732/***0.840/0.864	0.602/0.732/0.840	0.923/0.947/0.966
	9	0.360/0.502/0.587	0.732/0.838/0.870***	0.695/0.744/0.836	0.941/0.949/0.966
	10	0.360/0.506/0.587	0.731/0.838/0.864	0.100/0.586/0.848	0.100/0.738/0.966
FRMWK	2	0.428/0.503/0.633	***0.803/0.821/0.849	0.407/0.559/0.661	0.878/0.898/0.913
	3	***0.447/***0.561/0.587	0.776/0.838/0.857	***0.778/*0.848/0.915***	***0.946/*0.958/0.968***
	4	*0.517/*0.655/0.704	*0.812/*0.867/0.887*	**0.791/***0.834/0.930**	*0.948/*0.958/0.974**
	5	**0.501/**0.598/0.709*	**0.810/**0.843/0.877**	*0.797/**0.835/0.937*	**0.947/**0.957/0.976*
	6	0.418/0.537/0.688**	0.771/0.826/0.868	0.702/0.792/0.825	0.935/***0.948/0.956
	7	0.414/0.529/0.598	0.768/0.825/0.840	0.664/0.732/0.817	0.923/0.933/0.948
	8	0.412/0.544/0.656	0.770/0.830/0.848	0.662/0.709/0.760	0.923/0.929/0.942
	9	0.408/0.547/0.626	0.781/0.829/0.851	0.659/0.703/0.754	0.922/0.930/0.941
	10	0.408/0.552/0.662***	0.781/0.832/0.854	0.656/0.700/0.755	0.929/0.922/0.940

different values for the parameter β. We run the FRMVK, WMCFS and SWVF for the four real multi-view data sets and make comparisons using different α with a fixed β value. The exponent α is searched from 2 to 10. We also make the comparisons for the four data sets using different values of β with a fixed α value. The different values of β for WMCFS are 0.0001, 0.005, 0.025, 0.05, 0.075 and 0.1, while the different values of β for SWVF are in the range of [0 30] with the step 5. The results for the IS and MF data using different α values with a fixed β value are presented in Table 7, and for the Caltech-7/20 data are presented in Table 8. While the results for the IS and MF data using different β values with a fixed α value are presented in Table 9, and for the Caltech-7/20 data are presented in Table 10. By observing these experimental results, we conclude the following results.

Result 1: As shown in Table 7 on the IS dataset, the performance of our algorithm obtains the best clustering results when the exponent $\alpha = 4$ and, at the same time, the final numbers of feature components obtained from FRMVK are stable, as shown in Table 11. For the MF dataset, FRMVK gives the optimal clustering results when $\alpha = 4$, while for Caltech-7 when $\alpha = 10$ and for Caltech-20 when $\alpha = 3$.

Result 2: For the four real world data sets, the exponent parameter α in WMCFS is sensitive. As shown in Tables 7 and 9, it is seen that the best clustering results of WMCFS can be obtained when $\alpha = 10$ and $\beta = 0.1$ on the IS data set, while with $\alpha = 10$ and $\beta = 0.05$ on the MF data set, with $\alpha = 10$ and $\beta = 0.1$ on the Caltech-7 data set, and with $\alpha = 4$ and $\beta = 0.1$ on the Caltech-20 data set.

Result 3: For the four real world data sets, the exponent$\alpha$and the parameter β in SWVF are relatively sensitive. For the IS data set, when we run SWVF with $\beta = 10$ under different α, we find that the clustering performance obtained unbalance ARs and RIs. On the MF, Caltech-7, and Caltech-20 real data sets, we also find that different α and β have big impacts on clustering performance. From Tables 7 and 8, it is seen that different α produces different clustering performance.

Result 4: From the worst, average, and the best of ARs and RIs obtained by three algorithms, the proposed FRMVK algorithm actually presents better results than other two algorithms. In FRMVK, we only need to assign the exponent parameter α, but both WMCFS and SWVF need to have another parameter β.

TABLE 8. Worst/Average/Best of AR and RI from algorithms with different α on Caltech 7 and Caltech 20 data.

Method	α	Caltech 7		Caltech 20	
		AR	RI	AR	RI
WMCFS	2	0.295/0.623/0.793	0.503/0.698/0.838	0.201/0.341/0.557	0.538/0.676/0.830
	3	0.240/0.614/0.809	0.576/0.698/0.872	0.174/0.372/0.600*	0.616/0.731/0.840
	4	0.425/0.592/0.729	0.603/0.705/0.829	0.294/0.418/0.524	0.734/0.798/0.846
	5	0.417/0.548/0.645	0.606/0.701/0.779	0.318/0.394/0.498	0.747/0.822/0.857
	6	0.395/0.520/0.602	0.631/0.710/0.772	0.260/0.374/0.486	0.774/0.830/0.863
	7	0.437/0.536/0.632	0.655/0.724/0.781	0.312/0.381/0.470	0.807/0.836/0.862
	8	0.435/0.549/0.659	0.687/0.732/0.781	0.324/0.377/0.461	0.815/0.838/0.872
	9	0.425/0.551/0.663	0.698/0.734/0.785	0.321/0.372/0.448	0.830/0.839/0.868
	10	0.422/0.550/0.672	0.701/0.734/0.785	0.302/0.369/0.456	0.818/0.840/0.863
SWVF	2	0.403/0.540/0.683	0.675/0.746/0.820	**0.352/**0.458/0.567**	*0.844/*0.867/0.900*
	3	0.400/0.539/0.682	0.676/0.746/0.819	0.342/***0.449/0.553***	**0.843/**0.864/0.895**
	4	0.399/0.544/0.685	0.681/0.748/0.822	***0.350/0.445/0.552	***0.842/0.863/0.893
	5	0.404/0.552/0.689	0.683/0.751/0.822	0.347/0.441/0.549	***0.842/**0.864/0.895**
	6	0.399/0.556/0.700	0.692/0.753/0.821	0.334/0.422/0.546	0.163/0.829/0.888
	7	0.419/0.559/0.702	0.698/0.753/0.821	0.329/0.428/0.546	0.841/0.863/0.895**
	8	0.428/0.563/0.695	0.705/0.754/0.820	0.330/0.426/0.543	0.840/0.863/0.895**
	9	0.432/0.565/0.700	0.703/0.754/0.817	0.389/0.408/0.510	0.163/0.828/0.884***
	10	0.429/0.565/0.695	0.703/0.755/0.818	0.313/0.407/0.488	0.163/0.827/0.884***
FRMWK	2	0.538/0.586/0.632	0.709/0.728/0.762	0.271/0.331/0.423	0.818/0.831/0.845
	3	***0.548/**0.594/0.670	**0.712/0.735/0.765	0.388/0.459/0.528	0.819/0.836/0.870
	4	0.519/***0.584/0.665	0.689/0.722/0.760	0.352/0.469/0.532	0.817/0.840/0.876
	5	**0.584/*0.704/0.769***	*0.727/*0.771/0.792	0.386/0.479/0.537	0.813/0.841/0.881
	6	**0.584/*0.704/0.769	*0.727/*0.771/0.792	0.410/0.480/0.544	0.813/0.840/0.885
	7	**0.584/*0.704/0.768	*0.727/*0.771/0.792	0.415/0.481/0.554	0.812/0.838/0.886
	8	**0.584/*0.704/0.768	*0.727/*0.771/0.792	0.433/0.479/0.555	0.811/0.837/0.886
	9	**0.583/*0.704/0.769****	*0.727/*0.770/0.792	0.438/0.480/0.554	0.809/0.836/0.883
	10	*0.584/*0.704/0.769***	*0.727/*0.771/0.792	0.441/0.477/0.552	0.809/0.835/0.879

TABLE 9. Worst/Average/Best of AR and RI from algorithms with different β on the IS and MF data.

	β	IS		MF	
		AR	RI	AR	RI
WMCFS	0.0001	0.119/0.151/0.287	0.356/0.388/0.484	0.625/0.704/0.783	0.915/0.928/0.940
	0.005	0.119/0.171/0.348	0.306/0.415/0.556	0.604/0.711/0.794	0.908/0.933/0.947
	0.025	0.139/0.309/0.391	0.380/0.594/0.750	0.554/0.693/0.799	0.912/0.933/0.949
	0.05	0.193/0.349/0.401	***0.388/0.688/0.776	0.554/0.707/0.799	0.912/0.935/0.949
	0.075	0.223/0.391/0.532	0.373/0.713/0.788	0.558/0.667/0.764	0.912/0.926/0.939
	0.1	***0.235/0.382/0.537	0.378/0.706/0.800	0.556/0.693/0.795	0.912/0.933/0.948
SWVF	5	0.143/0.385/0.592	0.142/0.603/0.860	0.635/***0.750/0.924***	0.924/0.940/0.971***
	10	**0.360/***0.506/0.587	**0.731/**0.838/0.864	***0.669/**0.766/0.945*	***0.932/***0.947/0.979*
	15	0.143/0.481/0.609	0.142/***0.807/0.875	0.100/0.566/0.845	0.100/0.732/0.965
	20	0.143/0.494/0.599	0.142/0.774/0.875	0.100/0.587/0.845	0.100/0.737/0.965
	25	0.143/***0.506/0.599	0.142/0.778/0.885	0.100/0.574/0.836	0.100/0.735/0.965
	30	0.143/**0.517/0.616	0.142/0.782/0.885	0.695**/0.744/0.836	**0.941/**0.949/0.966
FRMVK	-	*0.517/*0.655/*0.703	*0.832/*0.863/0.887*	*0.797/*0.835/0.937**	*0.947/*0.957/0.976**

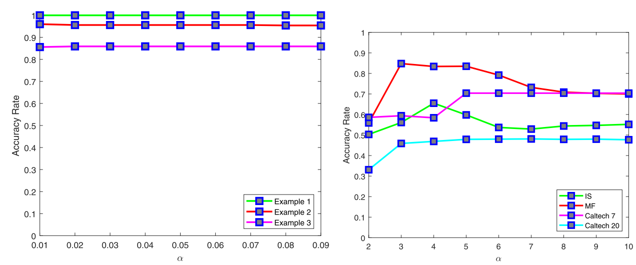
Result 5: For the four real datasets, Table 11 reports the total running time from each algorithm (in seconds). As shown in Table 11, we can see that the proposed FRMVK is much faster than WMCFS and SWVF. To further explain this phenomenon, we also present the final d obtained by FRMVK in each view. We observe that if the dataset

TABLE 10. Worst/Average/Best of AR and RI from algorithms with different β on Caltech 7 and Caltech 20 data.

	β	Caltech 7		Caltech 20	
		AR	RI	AR	RI
WMCFS	0.0001	0.321/0.581/0.803*	0.468/0.653/0.875*	0.267/0.358/0.428	0.770/0.810/0.837
	0.005	0.277/0.583/0.790**	0.504/0.673/0.865**	0.314/0.370/0.469	0.811/0.835/0.857
	0.025	0.325/**0.601/0.761	0.551/0.695/0.848***	0.329/0.376/0.458	**0.814/***0.838/0.871
	0.05	0.406/***0.596/0.741	0.583/0.703/0.836	0.329/0.377/0.460	*0.815/***0.838/0.872
	0.075	0.425/0.593/0.729	0.603/0.705/0.829	0.326/0.377/0.459	*0.815/***0.838/0.872
	0.1	**0.440/0.590/0.723	0.613/0.706/0.823	0.324/0.377/0.461	*0.815/***0.838/0.872
SWVF	5	0.356/0.530/0.698	0.643/0.740/0.813	**0.352/0.458/0.567**	**0.844/*0.867/0.900**
	10	**0.440/0.571/0.710	0.703/0.755/0.819	***0.334/0.458/0.579**	0.163/**0.841/0.907*
	15	***0.429/0.565/0.695	0.703/0.755/0.818	***0.334/**0.460/0.578***	0.163/**0.841/0.907*
	20	0.423/0.562/0.700	0.701/0.754/0.820	***0.334/***0.459/0.579**	0.163/**0.841/0.907*
	25	0.400/0.559/0.701	0.691/**0.753/0.821	***0.334/**0.460/0.579**	0.163/**0.841/0.907*
	30	0.406/0.555/0.694	0.684/***0.752/0.823	***0.334/**0.460/0.578***	0.163/**0.841/0.907*
FRMVK	-	*0.584/*0.704/0.769***	*0.727/*0.771/0.792	*0.415/*0.481/0.554	***0.812/***0.838/0.886***

has more total feature components, FRMVK will give much less running time than other algorithms. This situation occurs because FRMVK gives feature reduction behavior. The feature components will be discarded during iterations so that the number of feature components will be decreasing. This means that the computation time is also decreasing. The final number of feature components obtained from FRMVK for the four real data sets under different α are shown in Table 11.

Furthermore, we make more observations about the performance of FRMVK. We know that both WMCFS and SWVF algorithms quite depend on the parameter α and are sensitive to the parameter. The proposed FRMVK algorithm also depends on the parameter α where it is the only parameter. However, the proposed FRMVK algorithm is not so sensitive to the parameter α. We will demonstrate this behavior. To better asses how α competition has changed over clustering results, we first perform evaluation using three synthetic multi-view data sets of Examples 1, 2, and 3. Figure 4(a) shows the performance results in terms of accuracy rate of FRMVK by varying α in $\{0.01, 0.02, \ldots, 0.1\}$. We then perform evaluation using four natural multi-view data. The results, i.e., the average accuracy rate, are shown in Fig. 4(b). These results indicate that the performance of FRMVK is stable across a range of parameters $\alpha \in [0.00010.1]$ for the data sets. On the other hand, for the natural multi-view data, the results prove that FRMVK is pretty stable regardless of the choice of its parameter $\alpha > 1$. For synthetic data set with α in $\{0.01, 0.02, 0.03, 0.04, 0.05\}$, we report the weights in each view in Table 12. It can be seen that view weight distributions of each view in all three synthetic data sets are consistently stable for the given values of the parameter α. Besides, the proposed FRMVK algorithm also exploits these



(a) (b)

FIGURE 4. (a) Exponent parameter α for Examples 1, 2, and 3 by FRMVK; (b) Exponent parameter α for Example 4 by FRMVK.

important views based on the feature weight values. The greater the weight is the importance of the view is.

Finally, we further analyze more feature-reduction behaviors in the FRMVK algorithm. Usually, features with small weight values that fall below the defined threshold will be removed from the set of features during clustering processes. In other words, these features contain less information that should not be included. In the above examples, the FRMVK algorithm had been experimented and compared to the WMCFS and SWVF algorithms with synthetic and real data sets. The results show that the proposed FRMVK algorithm actually give the feature reduction behavior. However, we may ask whether these reduced features by FRMVK should be naturally discarded in data sets. From the synthetic data sets of Examples 1, 2 and 3, we had demonstrated that these reduced features by FRMVK are significantly irrelevant features in each view that need to be naturally discarded. In fact, for real data sets in Example 4, if we check these reduced features by FRMVK, we can find that these reduced

TABLE 11. Numbers of original and final features obtained from FRMVK and total running time (in seconds) for real data sets.

	α	Original d						Final d by FRMVK						Total running time		
		v_1	v_2	v_3	v_4	v_5	v_6	v_1	v_2	v_3	v_4	v_5	v_6	WMCFS	SWVF	FRMVK
	2							2	3	-	-	-	-	0.095	0.100	0.092
	3							2	8	-	-	-	-	0.097	0.106	0.097
	4							3	8	-	-	-	-	0.094	0.106	0.098
	5							4	8	-	-	-	-	0.096	0.105	0.096
IS	6	9	10	-	-	-	-	4	8	-	-	-	-	0.095	0.104	0.095
	7							4	8	-	-	-	-	0.099	0.105	0.095
	8							4	8	-	-	-	-	0.94	0.106	0.095
	9							4	8	-	-	-	-	0.098	0.106	0.094
	10							4	8	-	-	-	-	0.094	0.106	0.092
	2							12	50	63	1	29	1	0.953	1.364	0.231
	3							76	100	63	2	29	1	0.955	1.290	0.490
	4							76	104	63	4	29	1	0.968	1.271	0.533
	5							76	107	63	27	29	1	0.958	1.263	0.547
MF	6	76	216	64	240	47	6	76	108	63	54	29	1	0.960	1.325	0.587
	7							76	108	63	137	29	1	0.979	1.394	0.795
	8							76	108	63	207	29	1	1.004	1.252	0.908
	9							76	108	63	234	29	1	0.997	1.326	0.943
	10							76	108	63	236	29	1	1.035	1.254	1.017
	2							12	40	2	333	120	315	3.331	4.043	1.238
	3							6	40	2	581	259	345	3.113	3.959	1.350
	4							17	40	23	629	345	332	3.113	3.984	1.427
	5							22	40	57	629	442	375	3.262	4.003	3.709
Caltech 7	6	48	40	254	1984	512	928	24	40	67	629	442	375	3.230	3.904	3.699
	7							24	40	137	629	439	375	3.132	3.821	3.704
	8							24	40	192	629	438	375	3.113	3.826	3.711
	9							24	40	194	629	437	375	3.159	3.893	3.690
	10							24	40	208	629	436	375	3.286	3.971	3.687
	2							21	40	53	474	329	275	14.925	18.033	5.179
	3							24	40	194	1100	511	633	13.930	17.754	17.515
	4							24	40	220	1113	511	614	13.878	17.950	17.069
	5							24	40	188	1119	510	548	14.371	18.105	17.179
Caltech 20	6	48	40	254	1984	512	928	24	40	196	1115	510	527	15.857	18.151	16.926
	7							24	40	196	1116	510	523	13.930	17.735	16.931
	8							24	40	196	1116	510	521	13.882	17.789	16.958
	9							24	40	196	1116	510	521	14.214	17.806	16.989
	10							24	40	196	1116	510	520	14.077	18.338	16.929

features are less relevant compared to these reserved features. Another way is to see whether the feature reduction by FRMVK as an impact factor used in other multi-view clustering methods, such as WMCFS and SWVF, can produce better clustering results. To address this issue, we implement the WMCFS and SWVF algorithms for the data set of Example 2 and the IS (Image Segmentation) real data set of Example 4 with feature reduction (With FR) by FRMVK. Note that the chosen features in each view by FRMVK presented in Table 3 and Table 11. These clustering results (With FR) and without feature reduction (Without FR) from WMCFS and SWVF are shown in Table 13. The results show that the WMCFS and SWVF algorithms can yield better clustering quality for the data set (With FR) of Example 2 and IS (With FR) of Example 4. The average ARs and RIs of both algorithms (With FR) are increasing compared to those (Without FR), while the total running times are decreasing. It means that the feature reduction schema in the FRMVK algorithm is useful with beneficial impact on clustering results.

TABLE 12. Weight of each view.

	α	View weight		
		View 1	View 2	View 3
Example 1	0.01	0.6719	0.3281	-
	0.02	0.6748	0.3252	-
	0.03	0.6764	0.3236	-
	0.04	0.6781	0.3219	-
	0.05	0.6798	0.3202	-
Example 2	0.01	0.6505	0.3495	-
	0.02	0.6520	0.9460	-
	0.03	0.6551	0.3449	-
	0.04	0.6566	0.3434	-
	0.05	0.6586	0.3414	-
Example 3	0.01	0.1630	0.3496	0.4874
	0.02	0.1622	0.3493	0.4885
	0.03	0.1614	0.3492	0.4894
	0.04	0.1605	0.3491	0.4904
	0.05	0.1597	0.3490	0.4913

TABLE 13. Performance of WMCFS and SWVF without irrelevant features obtained by FRMVK.

		Example 2		IS	
		WMCFS	SWVF	WMCFS	SWVF
AV-AR	Without FR	0.554	0.878	0.151	0.506
	With FR	**0.598**	**0.882**	**0.420**	**0.553**
AV-RI	Without FR	0.728	0.890	0.388	0.838
	With FR	**0.747**	**0.903**	**0.712**	**0.869**
AV-TRT	Without FR	0.082	0.059	0.095	0.106
	With FR	**0.058**	**0.058**	**0.061**	**0.071**

V. CONCLUSION

In this paper, we propose a novel algorithm for clustering multi-view data, termed Feature-Reduction Multi-View K-Means (FRMVK), which can automatically reduce unimportant features in each view. The proposed FRMVK algorithm uses a learning mechanism to compute new feature weights in each view by adding a feature-weight entropy in the FRMVK objective function. These new weights are then used to update cluster centers, memberships, and view weights for the data set during iterative processes. The FRMVK algorithm is able to select important features in each view and to reduce feature dimensions by discarding unimportant features in each view. Experimental results show that the proposed FRMVK algorithm performs well for clustering multi-view data. In our future work, we will investigate the parameter selection for the view exponent α and then extend FRMVK to be suitable for different cluster shapes in multi-view data. Furthermore, there exist multi-view data with categorical or mixed data types, and our further research will try to extend FRMVK for handling these data types.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their helpful comments in improving the presentation of this paper. This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 107-2118-M-033-002-MY2.

REFERENCES

[1] C. C. Aggarwal and C. K. Reddy, *Data Clustering: Algorithms and Applications*. Boca Raton, FL, USA: CRC Press, 2013.

[2] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ, USA: Wiley, 1990.

[3] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, Berkeley, CA, USA, vol. 1, 1967, pp. 281–297.

[4] D. Pelleg and A. W. Moore, "X-means: Extending K-means with efficient estimation of the number of clusters," in *Proc. 17th Int. Conf. Mach. Learn.*, San Francisco, CA, USA, 2000, pp. 727–734.

[5] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient K-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.

[6] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[7] I. J. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, vol. 1, 2001, pp. 269–274.

[8] S. Bickel and T. Scheffer, "Multi-view clustering," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, vol. 4, 2004, pp. 19–26.

[9] X. Cai, F. Nie, and H. Huang, "Multi-view K-means clustering on big data," in *Proc. 23th Int. Joint Conf. Artif. Intell.*, 2013, pp. 2598–2604.

[10] Y. Yang and H. Wang, "Multi-view clustering: A survey," *Bid Data Mining Anal.*, vol. 1, no. 2, pp. 83–107, Jun. 2018.

[11] X. Chen, X. Xu, J. Z. Huang, and Y. Ye, "TW-K-means: Automated two-level variable weighting clustering algorithm for multiview data," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 932–944, Apr. 2013.

[12] B. Jiang, F. Qiu, and L. Wang, "Multi-view clustering via simultaneous weighting on views and features," *Appl. Soft Comput.*, vol. 47, pp. 304–315, Oct. 2016.

[13] J. Xu, J. Han, F. Nie, and X. Li, "Re-weighted discriminatively embedded *K*-means for multi-view clustering," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 3016–3027, Jun. 2017.

[14] Y.-M. Xu, C.-D. Wang, and J.-H. Lai, "Weighted multi-view clustering with feature selection," *Pattern Recognit.*, vol. 53, pp. 25–35, May 2016.

[15] G.-Y. Zhang, C.-D. Wang, D. Huang, W.-S. Zheng, and Y.-R. Zhou, "TW-Co-*K*-means: Two-level weighted collaborative *K*-means for multi-view clustering," *Knowl.-Based Syst.*, vol. 150, pp. 127–138, Jun. 2018.

[16] J. Farquhar, D. R. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmák, "Two view learning: SVM-2K, theory and practice," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 18, 2006, pp. 355–362.

[17] X. Xin, J. Wang, R. Xie, S. Zhou, W. Huang, and N. Zheng, "Semi-supervised person re-identification using multi-view clustering," *Pattern Recognit.*, vol. 88, pp. 285–297, Apr. 2019.

[18] S. Sun, "A survey of multi-view machine learning," *Neural Comput. Appl.*, vol. 23, nos. 7–8, pp. 2031–2038, 2013.

[19] X. Zhang, X. Zhang, H. Liu, and X. Liu, "Multi-task multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3324–3338, Dec. 2016.

[20] M.-S. Yang and Y. Nataliani, "A feature-reduction fuzzy clustering algorithm based on feature-weighted entropy," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 817–835, Apr. 2018.

[21] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li, "Automated variable weighting in k-means type clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 657–668, May 2005.

[22] L. Jing, M. K. Ng, and J. Z. Huang, "An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 8, pp. 1026–1041, Aug. 2007.

[23] J. Zhu, Z. Jiang, G. D. Evangelidis, C. Zhang, S. Pang, and Z. Li, "Efficient registration of multi-view point sets by *K*-means clustering," *Inf. Sci.*, vol. 488, pp. 205–218, Jul. 2019.

[24] G. Taguchi, S. Chowdhury, and Y. Wu, *Introduction to the Signal-to-Noise Ratio*. Hoboken, NJ, USA: Wiley, 2004.

[25] U. Fano, "Ionization yield of radiations. II. The fluctuations of the number of ions," *Phys. Rev.*, vol. 72, p. 26, Jul. 1947.

[26] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Statist. Assoc.*, vol. 66, no. 336, pp. 846–850, 1971.

[27] D. Dua and C. Graff, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, 2019. [Online]. Available: http://archive.ics.uci.edu/ml

[28] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Comput. Vis. Image Understand.*, vol. 106, no. 1, pp. 59–70, Jan. 2007.

MIIN-SHEN YANG received the B.S. degree in mathematics from the Chung Yuan Christian University, Chung-Li, Taiwan, in 1977, the M.S. degree in applied mathematics from the National Chiao-Tung University, Hsinchu, Taiwan, in 1980, and the Ph.D. degree in statistics from the University of South Carolina, Columbia, SC, USA, in 1989.

In 1989, he joined the faculty of the Department of Mathematics, Chung Yuan Christian University (CYCU), as an Associate Professor, where he has been a Professor, since 1994. From 1997 to 1998, he was a Visiting Professor with the Department of Industrial Engineering, University of Washington, Seattle, WA, USA. From 2001 to 2005, he was the Chairman of the Department of Applied Mathematics, CYCU. Since 2012, he has been a Distinguished Professor of the Department of Applied Mathematics and the Director of Chaplain's Office, and currently the Dean of the College of Science, CYCU. His research interests include clustering algorithms, fuzzy clustering, soft computing, pattern recognition, and machine learning. He was an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS (2005–2011), and is an Associate Editor of the *Applied Computational Intelligence and Soft Computing*.

KRISTINA P. SINAGA received the B.S. and M.S. degrees in mathematics from the University of Sumatera Utara, Indonesia. She is currently pursuing the Ph.D. degree with the Department of Applied Mathematics, Chung Yuan Christian University, Taiwan. Her research interests include clustering and pattern recognition.