IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 17, 2021, accepted May 2, 2021, date of publication May 5, 2021, date of current version May 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077622

Entropy K-Means Clustering With Feature Reduction Under Unknown Number of Clusters

KRISTINA P. SINAGA[1], ISHTIAQ HUSSAIN[2], AND MIIN-SHEN YANG[2]

[1]Department of Master in Information System Management, BINUS Graduate Program, BINUS University, Jakarta 10279, Indonesia
[2]Department of Applied Mathematics, Chung Yuan Christian University, Taoyuan City 32023, Taiwan

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

This work was supported in part by the Ministry of Science and technology (MOST) of Taiwan under Grant MOST-109-2118-M-033-001-.

ABSTRACT The k-means algorithm with its extensions is the most used clustering method in the literature. But, the k-means and its various extensions are generally affected by initializations with a given number of clusters. On the other hand, most of k-means always treat data points with equal importance for feature components. There are several feature-weighted k-means proposed in literature, but, these feature-weighted k-means do not give a feature reduction behavior. In this paper, based on several entropy-regularized terms we can construct a novel k-means clustering algorithm, called Entropy-k-means, such that it can be free of initializations without a given number of clusters, and also has a feature reduction behavior. That is, the proposed Entropy-k-means algorithm can eliminate irrelevant features with feature reduction under free of initializations with automatically finding an optimal number of clusters. Comparisons between the proposed Entropy-k-means and other methods are made. Experimental results and comparisons actually demonstrate these good aspects of the proposed Entropy-k-means with its effectiveness and usefulness in practice.

INDEX TERMS Clustering, k-means, entropy, feature weights, feature reduction, number of clusters, entropy-k-means.

I. INTRODUCTION

Clustering is a powerful tool in data analysis. It is used for discovering the cluster structure in data sets with the greatest similarity within the same cluster, but the greatest dissimilarity between different clusters. Generally, cluster analysis became a branch of statistical multivariate analysis, and it is an unsupervised learning approach to machine learning [1], [2]. In clustering, partitional methods are the most used. The simplest and popular partitional method was first proposed by MacQueen [3] in 1967, called a k-means clustering algorithm. The k-means clustering has been widely extended and applied in various areas [4]–[9]. Bai *et al.* [4] applied k-means in fast density clustering algorithm. Liu *et al.* [5] considered the extended genetic k-means. Jung *et al.* [6] gave a reinforce k-means for lowering data cost. Yu *et al.* [7] used self-paced learning to extend k-means. Han *et al.* [8] used k-means as vector quantization and Wang *et al.* [9] used k-means as fault recognition model for rotating machinery. One of extensions is to use feature weights, such as weighted k-means (WKM) [10] and entropy-weighted k-means (EWKM) [11]. Although these feature-weighted clustering algorithms may improve the performance of k-means, they do not consider a feature-reduction behavior. In general, if there exist irrelevant features during clustering processes, the clustering algorithm must take more computational time and even yields incorrect clustering results. Thus, a feature-reduction schema for k-means clustering is very important.

On the other hand, most of these k-means algorithms are usually affected by initializations with a given number of clusters a priori. However, the number of clusters is generally unknown. In this case, validity indices can be used to find a good number of clusters. Many cluster validity indices for the k-means clustering had been proposed in the literature. These are Bayesian information criterion (BIC) [12], Akaike information criterion (AIC) [13], Dunn's index (DU) [14], Davies-Bouldin index (DB) [15], Silhouette Index (SI) [16] and Calinski and Harabasz index (CH) [17]. For an efficient estimation of the number of clusters, Pelleg and Moore [18] extended k-means to X-means by using local decisions for cluster centers in each iteration of k-means with splitting

The associate editor coordinating the review of this manuscript and approving it for publication was Huiling Chen.

themselves to get better clustering. However, users need to specify a range of cluster numbers in which the true cluster number reasonably lies and then a model selection, such as BIC or AIC, is used to do the splitting process. Although these k-means clustering algorithms can find the number of clusters by cluster validity indices or X-means, they use extra iteration steps outside the clustering algorithms.

Another approach for solving the optimization problem in clustering is by considering metaheuristics algorithms such as krill herd (KH) [19]–[21] and hybrid swarm intelligence clustering ensemble (HSICE) [22]. The KH method was constructed based on the best krill individual in the population by Gandomi and Alavi [19], and then Li *et al.* [20] introduced a new version of KH with elitism strategy to improve the parameter estimation and simultaneously solve the optimum global issue in clustering problem. HSICE by Logesh *et al.* [22] combined the BrainStorm optimization algorithm and immune genetic algorithm to generate the diversified list of points of interest. Indeed, both KH and HSICE can solve the optimization problem in clustering, but their results still depend on the parameter selection and have high time-complexity. In this sense, choosing parameter issues and initialization assignments in clustering algorithms are sensitives and not guarantee an improvement for final outputs [21]. Recently, Yang and Sinaga [23] proposed an unsupervised k-means (U-k-means) clustering algorithm. The U-k-means algorithm [23] is free of initializations, parameter selection and also simultaneously find an optimal number of clusters during iteration steps.

However, most extensions of k-means, including these weighted k-means and U-k-means, do not give feature-reduction behaviors. In this paper, we extend the U-k-means algorithm such that it can eliminate irrelevant features with feature reduction under free of initialization and parameter selection with simultaneously finding the number of clusters. We call it the entropy-regularized k-means (Entropy-k-means). This is because we use several entropy-regularization terms to create learning schema with feature reduction behaviors and also automatically finding an optimal number of clusters. Totally, our approach includes the following ways. (i) First, we allocate all the data point as the cluster centers; (ii) After updating the feature weights, we decide to discard the unimportant features during clustering processes; (iii) The important features will be implemented to determine the number of clusters; (iv) After some iterations, our clustering algorithm will reduce the number of clusters by using our proposed defining criteria; (v) For the data sets in which produced some not available values for updated cluster centers in the second iteration, we replaced the not available values with the median of mean available values of that cluster centers.

The remainder of this paper is organized as follows. In Section II, we first review some related works. In Section III, we first construct the learning schema based on entropy regularization terms and then extend the U-k-means clustering algorithm to the Entropy-k-means based on a feature-weight entropy such that the proposed Entropy-k-means clustering algorithm has feature-reduction behaviors. The computational complexity of the proposed Entropy-k-means algorithm is also analyzed. In Section IV, experimental results and comparisons with some existing methods using synthetic and real data sets are used to demonstrate the effectiveness and usefulness of the proposed Entropy-k-means clustering algorithm. Finally, conclusions are stated in Section V.

II. RELATED WORKS

In this section, we give a brief review of the related works in the literature, such as k-means, weighted k-means (WKM) [10], entropy-weighted k-means (EWKM) [11], and unsupervised k-means (U-k-means) [23] algorithms. These related works will be also compared with our proposed Entropy-k-means algorithm in the experiments and comparisons section.

A. THE K-MEANS CLUSTERING ALGORITHM

In this paper, matrices are written as uppercase letters and vectors are written as lowercase letters. Let $\mathbf{X} = \{x_1, \ldots, x_n\}$ be a data set in a d-dimensional Euclidean space $\mathbb{R}^d$, $A = \{a_1, \ldots, a_c\}$ be the c cluster centers with its Euclidean norm denoted by $d_{ik} = \|x_i - a_k\|$. 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, $k = 1, \cdots, c$. The k-means algorithm is iterated through the updating equations for cluster centers and memberships by minimizing the k-means objective function $J(U, A) = \sum_{i=1}^{n}\sum_{k=1}^{c}\mu_{ik}\|x_i - a_k\|^2$ as $a_k = \sum_{i=1}^{n}\mu_{ik}x_{ij}/\sum_{i=1}^{n}\mu_{ik}$ and

$$\mu_{ik} = \begin{cases} 1 & \text{if } \|x_i - a_k\|^2 = \min\limits_{1\le k\le c} \|x_i - a_k\|^2 \\ 0, & \text{otherwise.} \end{cases}$$

B. THE WEIGHTED K-MEANS CLUSTERING ALGORITHM

Furthermore, Huang *et al.* [10] considered an extension of k-means by adding feature weights for data points, called the weighted k-means (WKM). Let $W = \left[w_{kj}\right]_{c\times d}$, where w_{kj} is the j-th feature weight in the k-th cluster center. The WKM objective function in Huang *et al.* [10] is as

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

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. Furthermore, Jing *et al.* [11] considered subspace clustering that is especially useful for high dimensional sparse data by using a feature-weighting approach. Jing *et al.* [11] 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 [11] is

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

where $\gamma \geq 0$ is a parameter to control the size of feature weights in each cluster. They applied EWKM to high dimensional sparse data, such as text clustering and business transaction data, where many attributes have zero-dimension.

C. THE U-K-MEANS CLUSTERING ALGORITHM

In general, the k-means algorithm and its extensions are always affected by initializations with a given number of clusters a priori. To solve these drawbacks, Sinaga and Yang [23] recently proposed the unsupervised k-means (U-k-means) clustering algorithm. The U-k-means algorithm extend k-means to be free of initializations with automatically finding an optimal number of clusters. In Sinaga and Yang [23], they consider the proportions α_k in which the α_k term is seen as the probability of one data point belonged to the kth class. Sinaga and Yang [23] gave the U-k-means objective function as follows:

$$\begin{aligned} J_{U-k-means}&(U,A,\alpha) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{c}\mu_{ik}\left\|x_i-a_k\right\|^2 - \beta n\sum_{k=1}^{c}\alpha_k\ln\alpha_k \\ &\quad -\gamma\sum_{i=1}^{n}\sum_{k=1}^{c}\mu_{ik}\ln\alpha_k \end{aligned} \qquad (3)$$

The U-k-means algorithm is iterated through the updating equations for cluster centers a_k, memberships μ_{ik} and proportions α_k by minimizing the U-k-means objective function $J_{U-k-means}(U,A,\alpha)$. If t denotes the iteration number in the algorithm with proportions $\alpha_k^{(t+1)}$ and $\alpha_k^{(t)}$, then β is estimated with

$$\beta^{(t+1)} = \min\left(\frac{\sum_{k=1}^{c}\exp(-\eta n\left|\alpha_k^{(t+1)}-\alpha_k^{(t)}\right|)}{c}, \frac{1-\max\limits_{1\leq k\leq c}\left(\frac{1}{n}\sum_{i=1}^{n}z_{ik}\right)}{(-\max\limits_{1\leq k\leq c}\alpha_k^{(t)}\sum_{k'=1}^{c}\ln\alpha_{k'}^{(t)})}\right)$$

and the parameter γ is set as $\gamma^{(t)} = e^{-c^{(t)}/250}$.

III. THE PROPOSED ENTROPY-K-MEANS CLUSTERING ALGORITHM

To construct the k-means clustering algorithm with free of initializations and automatically determine the number of clusters by considering the feature reduction schema, called unsupervised k-means with considering the feature reduction schema, we need to consider a penalty term with entropy concept. We first consider proportions α_k in which the α_k term is seen as the probability of one data point belonged to the kth class. Hence, we use $-\ln\alpha_k$ as the information in the point belonged to the kth class, and so $-\sum_{k=1}^{c}\alpha_k\ln\alpha_k$ becomes the average of information. In fact, the term $-\sum_{k=1}^{c}\alpha_k\ln\alpha_k$ is the entropy over proportions α_k. When $\alpha_k = 1/c$, $\forall k = 1, 2, \ldots, c$, we say that there is no information about α_k. At this point, we have the entropy achieve the maximum value. Therefore, we add this term to the k-means objective function $J(U, A, W)$ as a penalty. We then construct a schema to estimate α_k by minimizing the entropy to get the most information for α_k. To minimize $-\sum_{k=1}^{c}\alpha_k\ln\alpha_k$ is equivalent to maximizing $\sum_{k=1}^{c}\alpha_k\ln\alpha_k$. For this reason, we will add the penalty term $\sum_{k=1}^{c}\alpha_k\ln\alpha_k$ to the k-means objective function.

Furthermore, to exclude some irrelevant feature components during clustering processes, we next borrow the idea from the paper about Feature-reduction Fuzzy clustering algorithm (see Yang and Nataliani [24]). They considered $W = \left[w_j\right]_{1\times d}$ be with w_j as a feature weight of the j-*th* feature. δ_j is known as the parameter to control the feature weights. At this point, we add the feature weight entropy $(n/c)\sum_{j=1}^{d}w_j\ln\delta_j w_j$ as the third penalty term for the k-means objective function $J(U, A, \alpha, W)$. The constant n/c use to control the term. Thus, we propose the entropy-regularized k-means (Entropy-k-means) objective function as follows:

$$\begin{aligned} J(U,A,\alpha,W) = \sum_{i=1}^{n}\sum_{k=1}^{c}\mu_{ik}w_j\delta_j\left(x_{ij}-a_{kj}\right)^2 - \beta n\sum_{k=1}^{c}\alpha_k\ln\alpha_k \\ -\gamma\sum_{i=1}^{n}\sum_{k=1}^{c}\mu_{ik}\ln\alpha_k + \frac{n}{c}\sum_{j=1}^{d}w_j\ln\delta_j w_j \end{aligned} \qquad (4)$$

subject to $\sum_{k=1}^{c}\mu_{ik} = 1$, $\mu_{ik} \in \{0,1\}$, $\sum_{j=1}^{d}w_j = 1$, $w_j \in [0,1]$.

Here, α_k presents the probability of a data point belonged to the kth class. We know that, when β and γ in (4) are zero, it becomes the weighted k-means. In summary, the objective function (4) has four terms, where the first and fourth terms in (4) consist of a weighted k-means clustering. The second and third terms in (4) are known as primary terms in our scenario to reveal the number of clusters. As it can be seen, these two terms of μ_{ik} and α_k are controlled by two balancing parameters β and γ. The combination of the two parameters is essential to accelerate the proposed entropy k-means algorithm to determine the number of clusters. The fourth term is used to find the importance of feature components. The constant value of (n/c) is used to control the distribution of feature components in revealing the structure of data to significantly determine an optimal number of clusters by excluding these unimportant features during clustering processes. The

Lagrangian of (4) is

$$\begin{aligned} &J(U, A, \alpha, W, \lambda_1, \lambda_2, \lambda_3) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{c} \mu_{ik} w_j \delta_j \left(x_{ij} - a_{kj}\right)^2 \\ &\quad - \beta n \sum_{k=1}^{c} \alpha_k \ln \alpha_k - \gamma \sum_{i=1}^{n}\sum_{k=1}^{c} \mu_{ik} \ln \alpha_k + \frac{n}{c} \sum_{j=1}^{d} w_j \ln \delta_j w_j \\ &\quad + \lambda_1 \left(\sum_{k=1}^{c} \mu_{ik} - 1\right) - \lambda_2 \left(\sum_{k=1}^{c} \alpha_k - 1\right) \\ &\quad - \lambda_3 \left(\sum_{j=1}^{d} w_j - 1\right) \end{aligned} \tag{5}$$

By considering (5), the updating equations for memberships, cluster centers, and mixing proportions can be found. The updating equation for the Entropy-k-means objective function $J(U, A, \alpha, W)$ with respective to α_k is as follows:

$$a_k = \sum\nolimits_{i=1}^{n} \mu_{ik} x_{ij} \Big/ \sum\nolimits_{i=1}^{n} \mu_{ik} \tag{6}$$

By taking the partial derivative of (5) with respect to μ_{ik}, and setting them to be zero. Thus, the updating equation for μ_{ik} is obtained as follows:

$$\mu_{ik} = \begin{cases} 1, & \text{if} \displaystyle\sum_{j=1}^{d} \delta_j w_j \left\| x_i - a_k \right\|^2 - \gamma \ln \alpha_k \\ & \quad = \displaystyle\min_{1 \le k \le c} \sum_{j=1}^{d} \delta_j w_j \left\| x_i - a_k \right\|^2 - \gamma \ln \alpha_k \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Similiarly, we have $\frac{\partial \tilde{J}}{\partial \alpha_k} = -\beta n \left(\ln \alpha_k + 1\right) - \gamma \sum_{i=1}^{n} \frac{\mu_{ik}}{\alpha_k} - \lambda_2 = 0$. By multipying with α_k, we obtain

$$-\beta n \alpha_k \left(\ln \alpha_k + 1\right) - \gamma \sum\nolimits_{i=1}^{n} \mu_{ik} - \lambda_2 \alpha_k = 0 \tag{8}$$

and then $-\sum_{k=1}^{c} n\beta\alpha_k \ln \alpha_k - \sum_{k=1}^{c} n\beta\alpha_k - \gamma \sum_{k=1}^{c} \sum_{i=1}^{n} \mu_{ik} - \sum_{k=1}^{c} \lambda_2 \alpha_k = 0$. We get

$$\lambda_2 = -n\beta \sum\nolimits_{k=1}^{c} \alpha_k \ln \alpha_k - n\beta - n\gamma \tag{9}$$

By substituting (9) to (8), we have $-\beta n \alpha_k \left(\ln \alpha_k + 1\right) - \gamma \sum_{i=1}^{n} \mu_{ik} - \left(-n\beta \sum_{k=1}^{c} \alpha_k \ln \alpha_k - n\beta - n\gamma\right) \alpha_k = 0$. Thus, the updating equation for α_k can be obtained as follows:

$$\alpha_k^{(new)} = \frac{1}{n} \sum_{i=1}^{n} \mu_{ik} + \frac{\beta}{\gamma} \alpha_k^{(old)} \left(\ln \alpha_k^{(old)} - \sum_{s=1}^{c} \alpha_s^{(old)} \ln \alpha_s^{(old)} \right) \tag{10}$$

We should mention that (10) is important for our proposed Entropy-k-means clustering method in calculating the optimal number of clusters. In (10), $\sum_{s=1}^{c} \alpha_s \ln \alpha_s$ is the weighted mean of $\ln \alpha_k$ with the weights $\alpha_1, \ldots, \alpha_c$. For the kth mixing proportion $\alpha_k^{(old)}$, if $\ln \alpha_k^{(old)}$ is less than the weighted mean, then the new mixing proportion $\alpha_k^{(new)}$ will become smaller than the old $\alpha_k^{(old)}$. That is, the smaller proportion will decrease and the bigger proportion will increase in the next iteration, and then competition will occur. This situation is similar as the formula (11) in Figueiredo and Jain [25]. If $\alpha_k \le 0$ or $\alpha_k < 1/n$ for some $1 \le k \le c^{(old)}$, they are considered to be illegitimate proportions. In this situation, we discard those clusters (or set those proportions as zero) and then update the cluster number $c^{(old)}$ to be

$$c^{(new)} = c^{(old)} - \left| \left\{ \alpha_k^{(new)} \middle| \alpha_k^{(new)} < 1/(n \times (n-1)), \; k = 1, \ldots, c^{(old)} \right\} \right| \tag{11}$$

where $|\{\}|$ denotes the cardinality of the set $\{\}$. After updating the number of clusters c, the remaining mixing proportion $\alpha_{k'}$ and corresponding $\mu_{ik'}$ need to be re-normalized by

$$\alpha_{k'} = \alpha_{k'} \Big/ \sum\nolimits_{s=1}^{c^{(new)}} \alpha_s \tag{12}$$

$$\mu_{ik'} = \mu_{ik'} \Big/ \sum\nolimits_{s=1}^{c^{(new)}} \mu_{is} \tag{13}$$

A new problem is how to learn the values of the parameters γ for the penalty terms $\sum_{i=1}^{n} \sum_{k=1}^{c} \mu_{ik} \ln \alpha_k$ and $\sum_{k=1}^{c} \alpha_k \ln \alpha_k$, respectively. By considering some decreasingly learning rates, such as e^{-c^*}, $e^{-c^*/300}$, $e^{-c^*/600}$, and $e^{-c^*/900}$, we know that e^{-t} decreases faster but $e^{-t/600}$ and $e^{-t/900}$ decreases slower. Since $\sum_{i=1}^{n} \sum_{k=1}^{c} \mu_{ik} \ln \alpha_k$ has effect on membersips μ_{ik}, w_j and mixing proportions α_k, we assume that γ is not set to decrease too slow or too fast. Therefore, we set γ as

$$\gamma^{(t)} = e^{-c^*/300} \tag{14}$$

Similarly, by taking the partial derivative of (5) w.r.t w_j, we obtain the equation $\frac{\partial \tilde{J}}{\partial w_j} = \sum\limits_{k=1}^{c} \sum\limits_{i=1}^{n} \mu_{ik} \delta_j \left(x_{ij} - a_{kj}\right)^2 + \frac{n}{c} \left(\ln \delta_j w_j + 1\right) + \lambda_3 = 0$. Thus, the updating equation for w_j can be obtain as follows:

$$w_j = \frac{1}{\delta_j} \exp\left(\frac{-c \sum\limits_{k=1}^{c} \sum\limits_{i=1}^{n} \mu_{ik} \delta_j \left(x_{ij} - a_{kj}\right)^2}{n} \right) \Bigg/ \sum_{q=1}^{d} \frac{1}{\delta_q} \times \exp\left(\frac{-c \sum\limits_{k=1}^{c} \sum\limits_{i=1}^{n} \mu_{ik} \delta_q \left(x_{iq} - a_{kq}\right)^2}{n} \right) \tag{15}$$

Furthermore, in order to retain the constraint $\sum_{j'=1}^{d^{(new)}} w_{j'} = 1$, we adjust $w_{j'}$ by

$$w_{j'} = w_{j'} \Big/ \sum\nolimits_{q=1}^{d^{(new)}} w_q \tag{16}$$

Under competition schema setting, the Entropy-k-means algorithm can automatically determine the optimal number of clusters with considering the feature reduction schema. In our Entropy-k-means clustering algorithm, the parameter β can help us to control the competition. We discuss the variable β as follows. We can derive that

$$-e^{-1} \leq \alpha_k \ln \alpha_k < 0 \tag{17}$$

If $0 < \alpha_k \leq 1, \forall k = 1, 2, \ldots, c$, and let

$$E = \sum_{s=1}^{c} \alpha_s \ln \alpha_s < 0 \tag{18}$$

Then we have

$$\alpha_k E = \alpha_k \sum_{s=1}^{c} \alpha_s \ln \alpha_s < 0 \tag{19}$$

Using (17) and (19), we have that

$$-e^{-1}\beta < \beta\alpha_k(\ln \alpha_k - \sum_{s=1}^{c} \alpha_s \ln \alpha_s) < \beta(-\alpha_k E) \tag{20}$$

Under the constraint $\sum_{k=1}^{c} \alpha_k = 1$, and only when $\alpha_k < 1/2$, we can have that $(\ln \alpha_k - \sum_{s=1}^{c} \alpha_s \ln \alpha_s) < 0$. To avoid the situation where all $\alpha_k \leq 0$, the left hand of inequality (20) must be larger than $-\max\{\alpha_k | \alpha_k < 1/2, k = 1, 2, \cdots, c\}$. We now have an elementary condition of β as follows: $-e^{-1}\beta > -\max\{\alpha_k | \alpha_k < 1/2, k = 1, 2, \cdots, c\}$. Thus, we have

$$\beta < \max\{\alpha_k e | \alpha_k < 1/2, k = 1, 2, \cdots, c\} < e/2 \tag{21}$$

Therefore, to prevent β from being too big, we can use $\beta \in [0, 1]$. Furthermore, if the difference between $\alpha_k^{(new)}$ and $\alpha_k^{(old)}$ is small, then β must become large in order to enhance its competition. If the difference between $\alpha_k^{(new)}$ and $\alpha_k^{(old)}$ is large, then β will become small to maintain stability. Thus, we define an updating equation for β as

$$\beta = \frac{\sum_{k=1}^{c} \exp\{-\eta n |\alpha_k^{(new)} - \alpha_k^{(old)}|\}}{c} \tag{22}$$

where η can be set to be $\min\left\{1, t^{\lfloor t/2-1 \rfloor}\right\}$, where $\lfloor a \rfloor$ denotes the largest integer that is no more than a.

Furthermore, we need to consider the restriction of $\max\limits_{1\leq k\leq c} \alpha_k^{(new)} \leq 1$. However, $\max\limits_{1\leq k\leq c} \alpha_k^{(new)} \leq \max\limits_{1\leq k\leq c} \left(\frac{1}{n}\sum_{i=1}^{n} \mu_{ik}\right) + \frac{\beta}{\gamma} \max\limits_{1\leq k\leq c} \alpha_k^{(old)} \left(\ln \max\limits_{1\leq k\leq c} \alpha_k^{(old)} - \sum_{s=1}^{c} \alpha_s^{(old)} \ln \alpha_s^{(old)}\right)$ and $\max\limits_{1\leq k\leq c} \left(\frac{1}{n}\sum_{i=1}^{n} \mu_{ik}\right) + \frac{\beta}{\gamma} \max\limits_{1\leq k\leq c} \alpha_k^{(old)} \left(\ln \max\limits_{1\leq k\leq c} \alpha_k^{(old)} - \sum_{s=1}^{c} \alpha_s^{(old)} \ln \alpha_s^{(old)}\right) < \max\limits_{1\leq k\leq c} \left(\frac{1}{n}\sum_{i=1}^{n} \mu_{ik}\right) + \beta\left(-\left(\max\limits_{1\leq k\leq c} \alpha_k^{(old)} \sum_{s=1}^{c} \alpha_s^{(old)} \ln \alpha_s^{(old)}\right)\right)$. Thus, if $\max\limits_{1\leq k\leq c} \left(\frac{1}{n}\sum_{i=1}^{n} \mu_{ik}\right) - \beta \max\limits_{1\leq k\leq c} \alpha_k^{(old)} \sum_{s=1}^{c} \alpha_s^{(old)} \ln \alpha_s^{(old)} \leq 1$, then the restriction will be held. It follows that

$$\beta \leq \frac{\left(1 - \max\limits_{1\leq k\leq c} \left(\frac{1}{n}\sum_{i=1}^{n} \mu_{ik}\right)\right)}{\left(-\max\limits_{1\leq k\leq c} \alpha_k^{(old)} \sum_{s=1}^{c} \alpha_s^{(old)} \ln \alpha_s^{(old)}\right)} \tag{23}$$

By combining (22) and (23), we obtain

$$\beta = \min\left(\frac{\sum_{k=1}^{c} \exp(-\eta n \left|\alpha_k^{(new)} - \alpha_k^{(old)}\right|)}{c}, \frac{1 - \max\limits_{1\leq k\leq c} \left(\frac{1}{n}\sum_{i=1}^{n} \mu_{ik}\right)}{(-\max\limits_{1\leq k\leq c} \alpha_k^{(old)} \sum_{k'=1}^{c} \ln \alpha_{k'}^{(old)})}\right) \tag{24}$$

Because the β can jump at any time, we let $\beta = 0$ when the cluster number c is stable. When the cluster number c is stable, it means c is no longer decreasing. In our setting, we use all data points as initial means with $\mu_k = x_k$, i.e. $c^{initial} = n, \alpha_k = 1/c^{initial}, \forall k = 1, 2, \ldots, c^{initial}$, as initial mixing proportions, and we use $w_j = 1\big/d, \forall j = 1, \ldots, d$.

Another problem is how to estimate the value of δ_j in (4). δ_j is a measure on selecting unimportant feature of feature components. Largest values of δ_j will affect smallest feature weights, while smallest values of δ_j will affect largest feature weights. We first borrow the idea of coefficient of variance (CV) in statistic that is defined as $CV = \sigma\big/\mu$. 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. $SNR = \mu\big/\sigma$ (see [26]). Furthermore, in physics, Fano factor (FF) [27], which can be seen as a similar CV, had been proposed and defined as $FF = \sigma^2\big/\mu$. If we consider the reciprocal of Fano factor, that is similar as SNR being the reciprocal of CV, then we have $\mu\big/\sigma^2$, i.e. (mean/var). In other words, the reciprocal of FF (i.e. the ratio of mean to variance) can be used to describe the degree of clustered data. The smaller dispersion represents the data would be closer to the cluster center, while larger dispersion represents the data is far from the cluster center. For sufficiently clustering processes the larger dispersion is identified as unimportant features which can be discarded to reduce feature dimensions for more efficient clustering. To guarantee the ratio of mean to variance for a data set being always positive, its absolute value is taken. Therefore, we consider the estimate for δ_j as follows:

$$\delta_j = \left|\frac{\text{mean}(x)}{\text{var}(x)}\right|_j \tag{25}$$

To create a feature-reduction schema in our proposed Entropy-k-means algorithm, we need to select the irrelevant features via automatically adjust the feature weights 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, d dimension of features, and c number of clusters. In our Entropy-k-means schema, we consider (26) as a suitable threshold for discarding these irrelevant features in the data set.

$$w^{(t)} \leq 1\Big/\sqrt{ncd} \tag{26}$$

Otherwise, to discard those clusters in our Entropy-k-means schema, we use (27) to adjust $\alpha^{(t)}$ as

$$\alpha^{(t)} \leq \frac{1}{n\,(n-1)} \tag{27}$$

To be detailed, the γ and δ_j will be discussed in the next section by using some experimental design.

Thus, the proposed Entropy-k-means clustering algorithm can be summarized as follows:

Entropy-k-means algorithm

```
Fix $\varepsilon > 0$. Give initial $c^{(0)} = n, \alpha_k^{(0)} = 1/n, a_k^{(0)} = x_i$, $w_j = 1\big/d$ and initial learning rates $\beta^{(0)} = 1$. Set $t = 1$.
Step 1: Compute $\delta_j^{(t)}$ using data points $X$ by (25).
Step 2: Compute $\gamma^{(t)}$ by (14).
Step 3: Compute $\mu_{ik}^{(t)}$ using $a_k^{(t-1)}, \alpha_k^{(t-1)}, c^{(t-1)}, \gamma^{(t)}, \delta_j^{(t)}$, $w_j^{(t-1)}$ by (7).
Step 4: Update $w_j^{(t)}$ using $\delta_j^{(t)}$, $\mu_{ik}^{(t)}$, $c^{(t-1)}$, and $a_k^{(t-1)}$ by (15).
Step 5: Discard the total $d_r$ number of these $j$ features for $w^{(t)}$ with $w^{(t)} \leq 1\big/\sqrt{ncd}$ and set $d^{(new)} = d - d_r$.
Step 6: Adjust $w^{(t)}$ by (16).
Step 7: Update $\alpha_k^{(t)}$ with $\beta^{(t-1)}, \gamma^{(t)}$, $\mu_{ik}^{(t)}$ and $\alpha_k^{(t-1)}$ by (10).
Step 8: Compute $\beta^{(t)}$ with $\mu_{ik}^{(t)}$, $\alpha_k^{(t)}$ and $\alpha_k^{(t-1)}$ by (24).
Step 9: Update $c^{(t-1)}$ to $c^{(t)}$ by discard those clusters with $\alpha_k^{(t)} \leq 1/n\,(n-1)$ and adjust $\alpha_k^{(t)}$ and $\mu_{ik}^{(t)}$ by (12) and (13).
    IF $t \geq 60$ and $c^{(t-60)} - c^{(t)} = 0$, THEN let $\beta^{(t)} = 0$.
Step 10: Update $a_k^{(t)}$ with $c^{(t)}$ and $\mu_{ik}^{(t)}$ by (6).
Step 11: Compare $a_k^{(t)}$ and $a_k^{(t-1)}$.
    IF $\max_{1 \leq k \leq c^{(t-1)}} \left\| a_k^{(t)} - a_k^{(t-1)} \right\| < \varepsilon$, THEN Stop.
    ELSE $t = t + 1$ and return to Step 1.
```

IV. EXPERIMENTS AND COMPARISONS

In this section, we evaluate the performances of different γ and δ_j to simultaneously find the optimal number of clusters c with the feature reduction behavior by using some experiments. We firstly generating three artificial data sets in experiment 1 and simulating those three artificial data sets to see the effectiveness our proposed Entropy-k-means in improving the final clustering results. Then, eight real-world data sets in experiment 2, such as SPECTF Heart, Flea, Soybean small, Dermatology, Zoo, Soybean large, LSVT, and Yale base 64 × 64 are used in the comparison studies. In those two experiments, the performances of four validity indices are described i.e., DU [14], DB [15], SI [16] and CH [17]. We compare the proposed Entropy-k-means to four validity indices by using the original k-means. We further compare the proposed Entropy-k-means to WKM [10], EWKM [11] and U-k-means [23]. For measuring clustering performance, accuracy rate (AR) with $AR = \sum_{k=1}^{c} n\,(c_k)\big/n$ is generally used, where $n\,(c_k)$ is the number of data points that obtain correct clustering for the cluster k, and n is the total number of data points in the data set. The larger AR is the better clustering results.

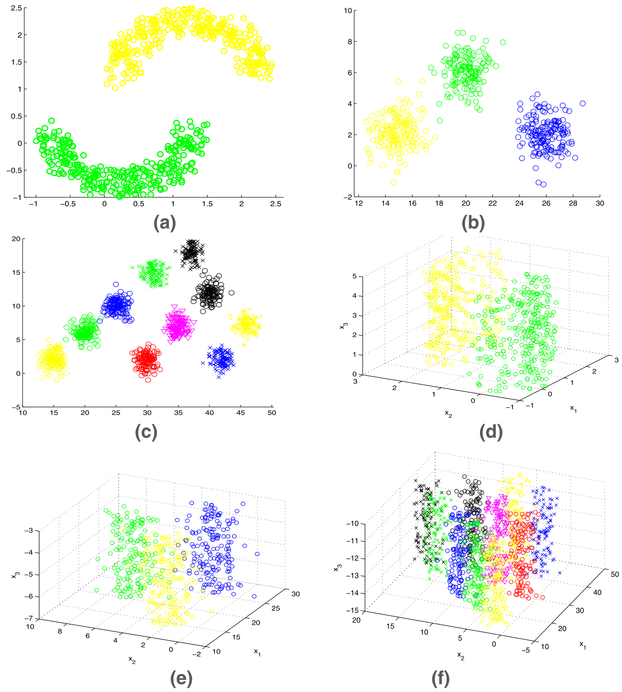

FIGURE 1. (a) A 2-D data set 1 (b) A 2-D data set 2 (c) A 2-D data set 3 (d) Data set 1 with a 2-D manifold plane and an embedded 1-D uniform (e) Data set 2 with a 2-D 3-spherical planes and an embedded 1-D (f) Data set 3 with a 2-D 10-spherical planes and an embedded 1-D uniform.

Experiment 1: In this experiment, we generated three artificial data sets as shown in Fig. 1. Fig. 1(a) has two manifold clusters and a total of 900 points, namely as data set 1. Fig. 1(b) has three spherical clusters, and a total number of points are 800, namely as data set 2. While Fig. 1(c) has ten spherical clusters, and a total number of points are 1200, namely as data set 3. To create the feature reduction scheme, we add one more dimension in each dataset by using uniform distribution. Without loss of generality, the one-dimensional generated by uniform distribution stretching the two-dimensional datasets into three-dimensional data sets, so that the additional feature component known as the unimportant feature. For data set 1, we displayed a mixture of two manifold and one-dimensional uniform distribution in Fig. 1(d). For data set 2, we displayed a mixture of three spherical clusters and one-dimensional uniform distribution in Fig. 1(e). While for data set 3, we displayed a mixture of ten spherical clusters and one-dimensional uniform distribution in Fig. 1(f). The important feature components for data set 1, data set 2, and data set 3 coordinated as x_1, and x_2, while unimportant feature coordinated as x_3.

Simulation 1 (Entropy-k-Means Under Different γ and δ_j): This simulation used to study the different γ and δ_j implementation in Entropy-k-means clustering algorithm. Table 1 summarizes the effectiveness of different γ and δ_j to cluster the data sets in Experiment 1 by using the Entropy-k-means clustering algorithm. The result shows that the AM-VR and $e^{-c^*/300}$ gives the best performance to simul-

TABLE 1. Simulation results at various and by Entropy-k-means for the data sets 1, 2, and 3.

Input δ_j		Input γ	Data set 1				Data set 2				Data set 3			
			True c	Red. d	Opt. c	AR	True c	Red. d	Opt. c	AR	True c	Red. d	Opt. c	AR
FF		e^{-c*}	2	2	N/A		3	1, 2	2		10	1, 2	N/A	
		$e^{-c*/300}$		2	5			1,2	2			1, 2	3	
		$e^{-c*/600}$		2	5			1,2	2			1, 2	3	
		$e^{-c*/900}$		2	5			1,2	2			1, 2	3	
AM-VR		e^{-c*}		3	N/A			3, 1	2			3	N/A	
		$e^{-c*/300}$		3	2	1.00		3	3	0.9960		3	10	0.9983
		$e^{-c*/600}$		3	3			3	4			3	10	0.9983
		$e^{-c*/900}$		3	2	1.00		3	7			3	10	0.9983

taneously determine the optimal number of clusters c with considering the feature reduction schema. For data sets 1, 2, and 3, FF estimates un-valid unimportant feature components in which affects the estimation number of clusters. For data set 1, by using FF in four different γ, Entropy-k-means overestimate $c^* = 5$. For the data sets 2 and 3, FF underestimated the number of clusters $c^* = 2$ and $c^* = 3$, respectively. By implemented AM-VR in different γ, we found that $\gamma = e^{-c^*/300}$ perform better in recognizing the important feature components to find the optimal number of c during the clustering processes. Thus, we set the parameter γ with $\gamma = e^{-c^*/300}$ that is used as an estimate of γ in (14).

Furthermore, we present some clustering processes by the Entropy-k-means. To be noted, every time we ran the data sets by the Entropy-k-means clustering algorithm, the data points are known as cluster centers. Figs. 2(a)-(f) shows the processes of the Entropy-k-means clustering algorithm for data set 1 in iteration 1, 3, 5, 10, 16 and 18, respectively. The number of feature components d discarding from 3 to 2. After some times, the number of c also decreases from 425 to 2. For data set 1, the Entropy-k-means clustering algorithm gives a correct number of $c = 2$ with consistently existing the two important features, as shown in Fig. 2(f). Figs. 3(a)-(f) shows the processes graphs by the Entropy-k-means for the data set 2 in iteration 1, 6, 12, 22, 51 and 56, respectively. As can be seen, the numbers of c decreases become 343, 20, 13, 8, 4, and 3. The proposed Entropy-k-means algorithm also able reduced the unimportant feature x_3 and consistently existing the two important features until it gives the correct number of $c = 3$, as shown in Fig. 3(f). Figs. 4(a)-(f) shows the clustering graphs for data set 3 in iteration 1, 9, 18, 27, 33 and 52, respectively. Similarly, as previous results, for data set 3, the proposed Entropy-k-means algorithm able to simultaneously reduced the unimportant feature x_3 and gives the correct number of $c = 10$, as shown in Fig. 4(f). The Entropy-k-means decreased the number of clusters from 1200 to 1000, 696, 258, 26, and 6, respectively. As we expected, the Entropy-k-means clustering algorithm performs well in these experiments. The proposed Entropy-k-means clustering algorithm simultaneously can reduce the unimportant feature and determine a correct number of c without depend on any initialization of cluster centers.

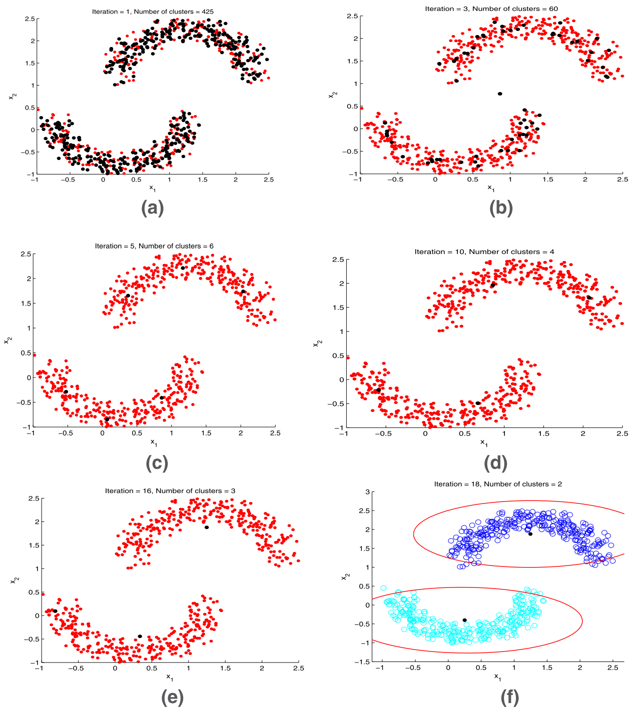

FIGURE 2. (a)-(e) The clustering result for data set 1 in iterations 1, 3, 5, 10 and 16 by Entropy-k-means (f) The final clustering result of data set 1 in iteration 18 by Entropy-k-means.

Simulation 2 (Cluster Structure): Next, we made a comparison between the proposed Entropy-k-means, k-means, WKM, and EWKM clustering algorithms. Four validity indices also made by using k-means, WKM, and EWKM. The four validity indices will be used in these comparisons are DU [14], DB [15], SI [16] and CH [17]. In order to investigate our feature reduction schema to find the optimal

TABLE 2. Experimental results on artificial data set with d = 2 and d = 3.

	Data set 1		Data set 2		Data set 3	
	3-D	2-D	3-D	2-D	3-D	2-D
k-means	0.538/0.672/0.995	1.00/1.00/1.00	0.520/0.870/0.996	0.534/0.947/0.996	0.548/0.778/0.870	0.545/0.797/0.868
WKM	0.267/0.669/1.00	0.687/0.940/1.00	0.512/0.784/0.998	0.500/0.737/0.996	0.378/0.661/0.846	0.408/0.677/0.869
EWKM	0.277/0.652/1.00	0.672/0.907/1.00	0.130/0.410/0.994	0.464/0.605/0.994	0.091/0.114/0.185	0.458/0.560/0.707
Entropy-k-means	**1.00**		**0.9960**		**0.9667**	

TABLE 3. Number of clusters obtained by the Dunn, DB, SW and CH validity indices, using the k-means, WKM, and EWKM algorithm.

	k-means				WKM				EWKM			
	DU	CH	DB	SI	DU	CH	DB	SI	DU	CH	DB	SI
Data set 1	2 (28%)	4	4	2 (28%)	2 (52%)	2 (20%)	2 (4%)	2 (20%)	2 (60%)	2 (64%)	2 (36%)	2 (40%)
Data set 2	3 (72%)	3 (72%)	3 (72%)	3 (28%)	3 (44%)	3 (56%)	3 (52%)	3 (24%)	3 (48%)	3 (44%)	3 (52%)	3 (16%)
Data set 3	5	12	10 (16%)	10 (8%)	2, 3, 12	12	2, 6, 8, 9	10 (4%)	3	2	17, 20	10 (4%)

TABLE 4. Total running time (TRT) using the k-means, WKM, EWKM, and Entropy-k-means clustering algorithms.

	TRT						
	k-means		WKM		EWKM		Entropy-k-means
	2-D	3-D	2-D	3-D	2-D	3-D	
Data set 1	0.079	0.080	0.099	0.086	0.060	0.160	**0.039**
Data set 2	0.076	0.065	0.074	0.061	0.068	0.079	**0.038**
Data set 3	0.166	0.064	0.076	0.068	0.195	0.227	**0.058**

number of clusters c, we examined the data sets in experiment 1 with $d = 2$ (2-important features) and $d = 3$ (2 important features + 1 unimportant feature) by using k-means, WKM, and EWKM clustering algorithms. We reran the k-means, WKM, and EWKM with 25 different initial random seeds. The obtained accuracy rates of these algorithms are shown in Table 2. For the clustering performances results, we show the worst, the average, and the best ARs. Bold values in the Tables indicate the clustering algorithm with best performance in terms of the accuracy rate. From Table 2, it can be seen that the k-means, WKM, and EWKM obtained the different results when it ran with $d = 2$ (2-important features) and $d = 3$ (2 important features + 1 unimportant feature). The ARs increasing when it ran in 2-D data sets. This result is to be expected, since unimportant features still exist during clustering processes will be affected the clustering result tends to be poor.

Table 3 presents the obtained number of clusters by implementing the four validity indices using the k-means, WKM, and EWKM. We show the percentage (%) for the correct number of clusters. For data set 1, four validity indices by using the WKM and EWKM has been successfully estimated c = 2, while CH and DB indices by using k-means overestimated c = 4. For data set 2, four validity indices by using the k-means, WKM, and EWKM estimated c = 3. For the data set 3, DB and SI by using k-means estimated c = 10; DU and CH overestimated c = 5 and c = 12, respectively. While for the WKM and EWKM, only one of the validity indices, namely SI estimated c = 10 (4%).

Overall, Entropy-k-means give the best accuracy rates for data set 1, data set 2, and data set 3 among these algorithms. Overall, Entropy-k-means performs better than k-means, WKM, and EWKM. To demonstrate their efficiency of algorithms, we also consider the total running times of these algorithms for data sets 1, 2, and 3. These are shown in Table 4. From Table 4, we find that the proposed Entropy-k-means have the least running time among these compared algorithms.

Experiment 2: In this clustering experiment we test the performance of our Entropy-k-means algorithm under 8 different real data sets, in which 6 of 8 data sets are collected from UCI repository [28]. These data sets namely as Single proton emission computed tomography (SPECT), Flea [29], Soybean small, Dermatology, Zoo, Soybean large, LSVT, and Yale base 64 × 64 (as shown in Fig. 5) [30]. Table 5 summarizes the data sets.

Simulation 2: We evaluate the performance of our Entropy-k-means clustering algorithm and compared it with the k-means, WKM, and E-WKM clustering algorithms.

TABLE 5. Total running time (TRT) and class wise distribution using the k-means, WKM, EWKM, and Entropy-k-means clustering algorithms.

Data sets	Type	Cluster number	Data number	Number of data used in experiment	Feature number	Number of features used in experiment	Class-wise distribution
SPECTF Heart	C	2	267	187	44	22	172, 15
Flea		3	74	74	5	5	21, 22, 31
Soybean Small	C	4	47	47	35	21	10, 10, 10, 17
Dermatology	M	6	366	358	32	32	116, 60, 71, 47, 52, 17
Zoo	M	7	101	101	16	16	4, 5, 8, 10, 13, 20, 41
Soybean large	C	15	307	266	35	35	10, 10, 10, 10, 10, 10, 10, 10, 10, 16, 20, 20, 40, 40, 40
	Data with more number of features						
LSVT	N	2	126	126	309	309	42, 84
Yale database 64x64	Image	15	165	75	4096	4096	5, 5, 5, 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5

N-Numerical; C-Categorical; M-Mixed

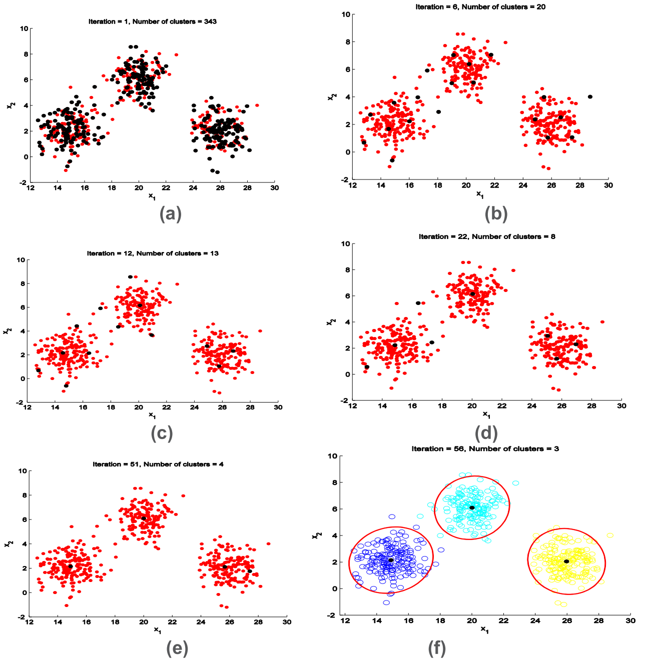

FIGURE 3. (a)-(e) The clustering result for data set 2 in iterations 1, 6, 12, 22 and 51 by Entropy-k-means (f) The final clustering result of data set 2 in iteration 56 by Entropy-k-means.

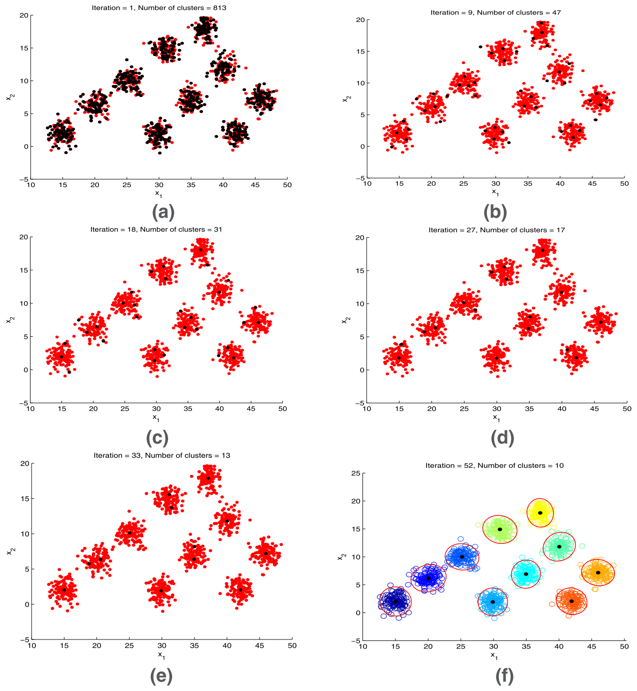

FIGURE 4. (a)-(e) The clustering result for data set 3 in iterations 1, 9, 18, 27 and 33 by Entropy-k-means (f) The final clustering result of data set 3 in iteration 52 by Entropy-k-means.

The clustering results is evaluated based on the accuracy rate (AR). Four validity indices also present by using the k-means, WKM, and EWKM clustering algorithms. Unlike k-means, WKM, and EWKM, our proposed Entropy-k-means algorithm is free from initial cluster centers. Our Entropy-k-means algorithm first initialize the number of clusters equal to the number of data points. After some iteration, our Entropy-k-means recognizing those features with large dispersion and discarding it. The clustering processes of Entropy-k-means only demonstrated the important features to find the optimal number of clusters. As we know, WKM

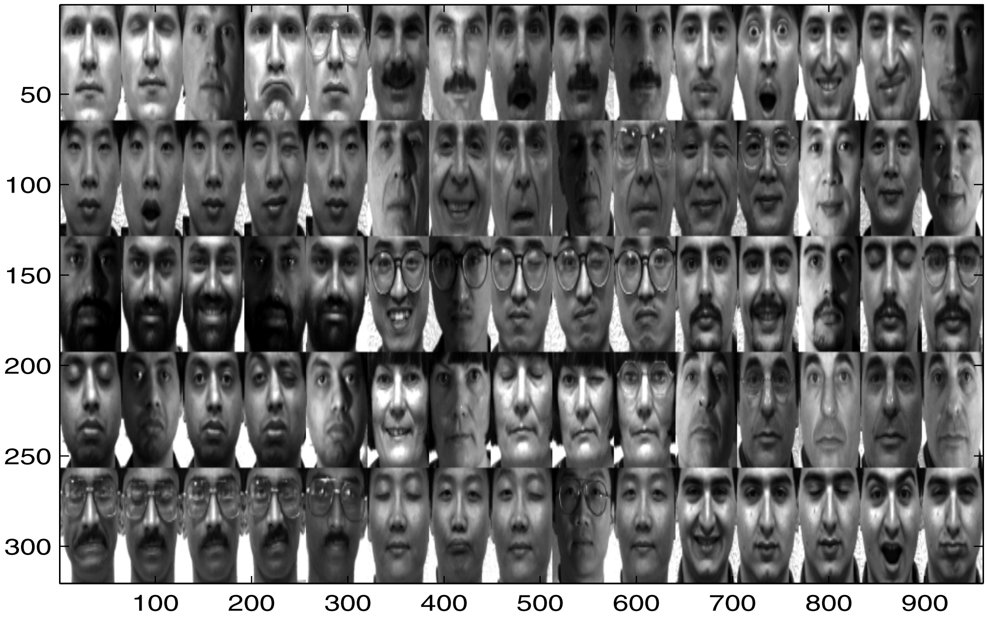

FIGURE 5. 75 face images of Yale database.

TABLE 6. Average of performance results of the k-means, WKM, EWKM, and Entropy-k-means algorithms.

Data sets	k-means		WKM		EWKM		Entropy-k-means	
	AR	Time	AR	Time	AR	Time	AR	Time
SPECTF Heart	0.5254	0.0686	0.6059	0.0560	0.6038	0.0596	**0.9198**	**0.0401**
Flea	0.7358	0.0638	0.6412	0.0523	0.7885	0.0583	**1.00**	**0.0403**
Soybean small	0.6678	0.0651	0.5804	0.0536	0.6501	0.0563	**0.9574**	**0.0392**
Dermatology	0.2061	0.0963	0.2137	0.0562	0.4219	0.0758	**0.4777**	**0.0706**
Zoo	0.6326	0.0643	0.5886	0.0543	0.6309	0.0607	**0.6733**	**0.0472**
Soybean large	0.5322	0.1044	0.4175	0.0570	0.4361	0.0936	**0.5338**	**0.0500**

and EWKM algorithms also based on one user define parameter. As comparisons, for WKM algorithm, we set $\beta = 10$. For EWKM, we set $\gamma = 10$. Table 6 compares the accuracy rate performance of our Entropy-k-means algorithm with k-means, WKM, and EWKM algorithms. From Table 6, we clearly indicated that our Entropy-k-means clustering algorithm performed the best accuracy rate, compared with other clustering algorithms. Our Entropy-k-means improved the accuracy rates over eight data sets and successfully identified the correct number of clusters c with considering the feature reduction schema (see Table 7). The k-means, WKM, and EWKM simulated over 25 different cluster centers initializations.

To be more detailed, Table 8 and Table 9 presents the behavior of our Entropy-k-means clustering algorithm in each iteration. As can be seen, for the LSVT data set, the number of features decreasing rapidly from originally 309 into 69, 35, 19, 12, 5. This clustering processing performed a different amount of features in each iteration. In other words, our Entropy-k-means clustering algorithm always tries to estimate the unimportant features from the remaining features in each iteration. So that the number of dimensions that will be evaluated in the next iteration affecting the different number of clusters. For the LSVT data set, our Entropy-k-means clustering algorithm reduced the number of c from initially 126 into 95, 31, 5, 2, 2, 2. As can be seen in Table 8, our Entropy-k-means was able to detect the correct number of c

TABLE 7. The class wise distribution by Entropy-k-means clustering algorithms.

Data sets	Fin. d	Opt. c	Class-wise distribution
SPECTF Heart	4	2	185, 2
Flea	3	3	21, 31, 22
Soybean small	5	4	10, 10, 8, 19
Dermatology	3	6	268, 22, 13, 51, 2, 2
Zoo	5	7	23, 39, 6, 7, 4, 20, 2
Soybean large	11	15	10, 10, 4, 26, 16, 4, 10, 10, 10, 6, 3, 3, 8, 118, 27
LSVT	5	2	125, 1
Yale base	314	15	3, 9, 5, 8, 5, 5, 7, 1, 5, 3, 3, 4, 7, 6, 4

TABLE 8. The details of performance of Entropy-k-means for LSVT data set in each iteration.

No. of iterations	Dimensionality behavior		Cluster number behavior		AR	Class-wise distribution by Entropy-k-means
	Originally dimension	Dimension reduced	c	Number of c reduced		
Initialization	309	-	126	-	-	
Iteration 1	309	240	126	31	-	
Iteration 2	69	34	95	64	-	
Iteration 3	35	16	31	26	-	
Iteration 4	19	7	5	3	0.5000	95, 28
Iteration 5	12	7	2	-	0.5952	114, 12
Iteration 6	5	-	2	-	0.6587	125, 1

TABLE 9. The details performance of Entropy-k-means for Yale base 64 × 64 data set in each iteration.

No. of iterations	Dimensionality behavior		Cluster number behavior		AR	Class-wise distribution by Entropy-k-means
	Originally dimension	Dimension reduced	c	Number of c reduced		
Initialization	4096	-	75			
Iteration 1	4096	3637	75	30	-	
Iteration 2	459	16	45	2	-	
Iteration 3	443	2	43	3	-	
Iteration 4	441	1	40	0	-	
Iteration 5	440	7	40	2	-	
Iteration 6	433	25	38	0	-	
Iteration 7	408	5	38	1	-	
Iteration 8	403	2	37	1	-	
Iteration 9	401	0	36	2	-	
Iteration 10	401	6	34	0	-	
Iteration 11	395	1	34	20	-	
Iteration 12	394	67	15	-	0.3867	3, 7, 2, 6, 5, 5, 5, 2, 2, 2, 2, 4, 3, 3, 4
Iteration 13	327	11	15	-	0.6133	4, 9, 4, 7, 5, 5, 7, 1, 6, 3, 3, 4, 7, 6, 4
Iteration 14	316	2	15	-	0.6000	3, 10, 5, 8, 4, 5, 7, 1, 5, 3, 3, 4, 7, 6, 4
Iteration 15	314	-	15		0.6133	3, 9, 5, 8, 5, 5, 7, 1, 5, 3, 3, 4, 7, 6, 4

starting from iteration 4 with AR = 0.5000. The Entropy-k-means clustering algorithm is also increasing the ARs values of LSVT data set until it reached AR = 0.6667. For Yale base 64 × 64 data sets, our Entropy-k-means reduced the number of features from originally 4096 into 459, 443, 441, 440, 433, 408, 403, 401, 401, 395, 394, 327, 316, 315. At the same time, by demonstrating those feature components during the clustering processes, the number of c also decreasing from originally 75 into 45, 43, 40, 40, 38, 38, 37, 36, 34, 34, 15, 15, 15, 15. As can be seen in Table 9, our Entropy-k-means was able to detect the correct number of c by iteration 12 with AR = 0.3867. The Entropy-k-means clustering algorithm also reduces some unimportant feature components. The Entropy-k-means clustering algorithm is increasing the ARs value of Yale base data set until its AR = 0.6133. The experiment results of K-means, WKM, EWKM, and Entropy-k-means in terms of clustering performances for LSVT and Yale data sets are also made, shown in Table 10. As we can see, our proposed Entropy-k-means performed the best results, showing the effectiveness of our proposed idea of reducing the uninformative features does not hurt the clustering performance but increased. Table 11 presents the validity indices with DU [14], DB [15], SI [16] and CH [17] implemented by using k-means, WKM and EWKM. For each validity index, the best result on 25 runs is taken. From all results, the proposed Entropy-k-means clustering algorithm performs better to find the correct number of clusters without initialization of cluster centers and with the feature reduction behavior.

Experiment 3: In this clustering experiment, we test the performance of our Entropy-k-means algorithm under 8 different real data sets, which 4 data sets are from the previous experiment, and four additional sets summarized in Table 12. We used these 8 real data sets to compare our proposed Entropy-k-means with k-means + DU, clustering by fast search (C-FS) [31], and U-k-means [23] clustering algorithms. The experimental results in terms of cluster number estimation are summarized in Table 13. As can be seen,

TABLE 10. The accuracy rate and total running time by using k-means, WKM, EWKM, and Entropy-k-means algorithm.

Algorithm	LSVT		Yale	
	Accuracy rate	Total running time	Accuracy rate	Total running time
k-means	0.4762/0.4845/0.5317	0.0721	0.2800/0.4406/0.5333	0.8049
WKM	0.3730/0.4576/0.5714	0.0678	0.2800/0.4053/0.4800	0.2734
EWKM	0.6429/0.6571/0.6667	0.0681	0.2800/0.4411/0.5333	1.8818
Entropy-k-means	**0.6587**	**0.0477**	**0.6133**	**0.0437**

TABLE 11. Number of clusters obtained by the Dunn, DB, SW and CH validity indices, using the k-means, WKM, and EWKM.

	k-means				WKM				EWKM			
	DU	CH	DB	SI	DU	CH	DB	SI	DU	CH	DB	SI
SPECTF	2 (32%)	2 (100%)	5	2 (84%)	2 (16%)	2 (60%)	2 (8%)	2 (68%)	2 (20%)	2 (48%)	5	2 (64%)
Chess	2 (20%)	2 (64%)	5	2 (60%)	2 (20%)	2 (64%)	2 (12%)	2 (40%)	2 (44%)	2 (44%)	2 (16%)	2 (40%)
Flea	3 (16%)	4	3 (8%)	2	3 (8%)	3 (4%)	3 (16%)	2	3 (40%)	3 (8%)	3 (4%)	3 (24%)
Soybean small	2	4 (24%)	4 (20%)	4 (12%)	4 (12%)	4 (28%)	4 (12%)	4 (20%)	4 (8%)	4 (16%)	4 (16%)	4 (16%)
Dermatology	6 (12%)	2	2, 3	2	6 (16%)	6 (4%)	3	2	6 (4%)	6 (8%)	6 (16%)	6 (4%)
Zoo	7 (4%)	2	7 (8%)	7 (12%)	7 (4%)	7 (4%)	7 (4%)	7 (12%_	7 (8%)	7 (8%)	7 (8%)	7 (16%)
Soybean large	2	2	15 (4%)	15 (4%)	2	2	2	18	4	2	15 (4%)	15 (12%)
LSVT	2 (95 %)	5	4	5	2 (10%)	2 (25%)	4	2 (25%)	2 (100%)	2 (65%)	2 (70%)	2 (70%)
Yale base 64x64	15 (19%)	15 (4%)	18	NA	15 (12 %)	15 (4%)	15 (4%)	NA	15 (12 %)	15 (4%)	18	NA

NA stands for Not Available (for example due to infinite or divide by zero issues).

TABLE 12. The characteristics of the data sets in experiment 3.

Data sets	Number of classes	Number of data	Number of data used in experiment	Number of features	Number of features used in experiment	Class-wise distribution
Fisher iris	3	150	150	4	4	50, 50, 50
Bupa	2	345	345	6	6	145, 200
PIMA	2	768	768	9	9	500, 268
Australia	2	690	690	14	14	307, 383

DU indices underestimate the number of clusters $c^* = 2$ for Fisher Iris and $c^* = 2$ for Soybean small. C-FS underestimates the number of clusters $c^* = 1$ for Bupa, $c^* = 2$ for Flea, $c^* = 3$ for Soybean small, and $c^* = 3$ for Zoo. U-k-means overestimates the number of clusters $c^* = 15$ for Flea, $c^* = 6$ for Soybean small, and underestimates the number of clusters $c^* = 3$ for Zoo. The Entropy-k-means can provide satisfactory results in estimating the correct number of clusters for these 8 data sets. Furthermore, the detailed performance of Entropy-k-means in terms of feature behavior are summarized in Table 14. The result proved the effectiveness of the proposed Entropy-k-means clustering algorithm in reducing uninformative features and still estimates the correct number of clusters. The experiment of k-means, U-k-means, and Entropy-k-means in terms of clustering performances and total running time (TRT) for Fisher Iris, Bupa, Flea, Pima, LSVT, and Australia data sets are also made. In this experiment, except AR, we also use more evaluations for clustering performance. These are RI (Rand Index) [32], FMI (Fowlkes-Mallows-Index) [33], NMI (Normalized Mutual Information) [34], and JI (Jaccard Index) [35]. Let $C = \{C_1, C_2, \cdots, C_c\}$ be the set of c clusters for the given data set and $C' = \left\{C'_1, C'_2, \cdots, C'_c\right\}$ be the set of c clusters generated by the clustering algorithm. Let

TABLE 13. Number of clusters obtained by the k-means with true c, C-FS, U-k-means, and Entropy-k-means algorithms.

	True c	k-means + DU	C-FS	U-k-means	Entropy-k-means
Fisher iris	3	2	2	**3**	**3**
Bupa	2	**2**	1	**2**	**2**
PIMA	2	**2**	**2**	**2**	**2**
Australia	2	**2**	**2**	**2**	**2**
LSVT	2	**2 (95%)**	**2**	**2**	**2**
Flea	3	**3 (16%)**	2	15	**3**
Soybean small	4	2	3	6	**4**
Zoo	7	**7 (4%)**	3	3	**7**

TABLE 14. The details performance of Entropy-k-means in terms of feature reduction behavior for the data sets in Experiment 3.

Data set	Dimensionality behavior			Class-wise distribution by Entropy-k-means
	Originally dimension	Dimension reduced	Final dimension	
Fisher iris	4	2 (1, 2)	2	50, 52, 48
Bupa	6	5 (1-4, 6)	1	305, 40
PIMA	9	8 (1-8)	1	500, 268
Australia	14	13 (1-8, 10-14)	1	295, 395
LSVT	310	305	5	125, 1
Flea	5	2 (1, 2)	3	21, 31, 22
Soybean small	21	16 (1-10, 11-14, 16-17)	5	10, 10, 8, 19
Zoo	16	11 (3-8, 10-11, 14,16)	5	23, 39, 6, 7, 4, 20, 2

$\left(X_i, X_j\right)$ 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 is the number of points if the two points belong to the same cluster in C and to two different clusters in C', and d be the number of pairs of points if the two points belong to two different clusters in C and to the same cluster in C'. RI is defined as $\mathrm{RI} = (a+d)\big/(n\,(n-1)\,/2)$ where n is the number of data points. FMI can be defined as $FMI = a/\sqrt{(a+b)\,(a+d)}$. NMI can defined as $NMI = 2\,I(X{:}Y)/\,[H(X)+H(Y)]$ where $I(X{:}Y)$ is the mutual information between the class labels $H(X)$ and the cluster labels $H(Y)$. JI is commonly used to measures the similarity between two data points and is defined as the size of the intersection divided by the size of the union of the two data points. These AR, RI, FMI, NMI, and JI ranges from 0 to 1, where 1 indicates a higher similarity between cluster solutions. We implement the k-means with true c^* over 25 different random initializations and shown the average AR, RI, FMI, NMI, and JI after 25 runs. The results are presented in Table 15. According to Table 15, Entropy-k-means is superior compare to k-means and U-k-means clustering algorithms.

Experiment 4: In this clustering experiment, we test the performance of our Entropy-k-means algorithm on digit recognizer. We use the most challenges and popular MNIST (Modified Institute of Standards and Technology) database of handwritten digits [36]. The MNIST database is collected by Yann Lecun and openly accessible at http://yann.lecun.com/exdb/mnist/index.html website. The MNIST database contains 70,000 28 × 28 black and white images representing the digits ranging from zero to nine. The data is split into two subsets, with 60,000 images belonging to the training set and 10,000 images belonging to the testing test. We subsampled 501 of 70,000 images to compose our data set, and they belong to 10 classes. Specifically, we randomly implemented 501 samples and training 100 multi-way from the MNIST database. Each digit is a gray-level image with 784 pixels in total as the features. Some examples are shown in Fig. 6.

Since the original dimensions are quite sparse, we first extract an image of dimensions 28 × 28 by conducting a pre-processing step over the samples using principal components. We thus processing the extracted features of principle components into our proposed Entropy-k-means algorithm. In this case, PCA normalizes all the grey-level pixels from the image and reduce its size to fit with 500 pixels in total as the features. A comprehensive summary of the results for Entropy-k-means and U-k-means is given in Table 16.

TABLE 15. Clustering performances and total running time (TRT) by using k-means, U-K-Means, and Entropy-k-means algorithms.

	Algorithms	Data sets					
		Fisher iris	Bupa	Flea	Pima	LSVT	Australia
AR	k-means	0.8632	**0.5113**	0.9736	**1**	0.5073	0.5551
	U-k-means	0.8400	0.4609	-	0.6510	0.4762	0.5551
	Entropy-k-means	**0.9600**	0.5043	**1**	**1**	**0.6587**	**0.7333**
RI	k-means	0.8647	**0.5039**	0.7194	**1**	0.5006	0.5066
	U-k-means	0.8368	0.4989	-	0.5458	0.4961	0.5106
	Entropy-k-means	**0.9495**	0.5026	**1**	**1**	**0.5468**	**0.6083**
FMI	k-means	0.8046	**0.6394**	0.6797	**1**	0.5833	**0.7081**
	U-k-means	0.7686	0.6000	-	0.7380	0.5833	0.7007
	Entropy-k-means	**0.9233**	0.6341	**1**	**1**	**0.7346**	0.6142
NMI	k-means	0.7389	0.0012	0.5520	**1**	0.0301	0.0287
	U-k-means	0.7224	**0.0104**	-	0.0171	**0.0448**	0.0640
	Entropy-k-means	**0.8642**	0.0027	**1**	**1**	0.0188	**0.1582**
JI	k-means	0.6739	**0.4527**	0.4106	**1**	0.4098	**0.5036**
	U-k-means	0.6223	0.4200	-	0.7380	0.4098	0.4994
	Entropy-k-means	**0.8575**	0.4483	**1**	**1**	**0.5444**	0.4432
TRT (second)	k-means	5.047	5.282	4.322	4.431	4.823	4.549
	U-k-means	0.258	0.709	**0.147**	**1.848**	0.408	2.306
	Entropy-k-means	**0.202**	**0.415**	0.148	1.883	**0.238**	**1.562**

TABLE 16. Result of Entropy-k-means and U-k-means algorithms for the 501 samples for the MNIST data base of handwritten digits.

	c*=2	c*=3	c*=4	c*=5	c*=6	c*=7	c*=8	c*=9	**c*=10**	c*=11
Entropy-k-means	0%	0%	0%	0%	0%	0%	22%	35%	**35%**	8%
U-k-means	12%	16%	33%	26%	9%	3%	1%	0%	**0%**	0%

TABLE 17. Clustering performances of Entropy-k-means and k-means with true c algorithms for the 501 samples for the MNIST data base of handwritten digits over 5 simulations.

Sim.		c^*	Final d	AR	RI	FMI	NMI	JI
1	Entropy-k-means	10	63	**0.6148**	**0.8939**	**0.4765**	**0.5558**	**0.3126**
	k-means with true c			0.5198	0.8819	0.4639	0.5546	0.3005
2	Entropy-k-means	10	71	**0.6607**	**0.8948**	**0.4923**	**0.5724**	**0.3262**
	k-means with true c			0.5351	0.8863	0.4677	0.5577	0.3051
3	Entropy-k-means	10	48	**0.6367**	**0.9036**	**0.4741**	**0.6146**	**0.3584**
	k-means with true c			0.4579	0.8759	0.4411	0.5552	0.2809
4	Entropy-k-means	10	69	**0.7305**	**0.9308**	**0.6611**	**0.7282**	**0.4934**
	k-means with true c			0.6766	0.9186	0.6205	0.7049	0.4494
5	Entropy-k-means	10	69	**0.7146**	**0.9269**	**0.6440**	**0.6904**	**0.4745**
	k-means with true c			0.6540	0.9154	0.6019	0.6843	0.4298

Table 16 shows that the proposed Entropy-k-means algorithm estimates the correct number of clusters $c^* = 10$ with 35% of 100 training for every 501 samples. Also, entropy-k-means estimates 22% of samples with $c^* = 8$, 35% of samples with $c^* = 9$, and 8% of samples with $c^* = 11$. In this experiment, we also provide the clustering performances of the proposed Entropy-k-means in terms of AR, RI [27], FMI [28], NMI [29], and JI [30]. Table 17 presents the details about the Entropy-k-means clustering performance over five different simulations. As can be seen, our proposed Entropy-k-means have reached the goals to simultaneously estimate the correct number of clusters and discard the uninformative features. To compare the proposed algorithm, we ran the k-means clustering algorithm with the correct number of clusters under 25 random initializations over 5 data simulations and reported the average in Table 17. As we can see, our proposed Entropy-k-means performed the best results, showing the effectiveness of our proposed idea of reducing the uninformative

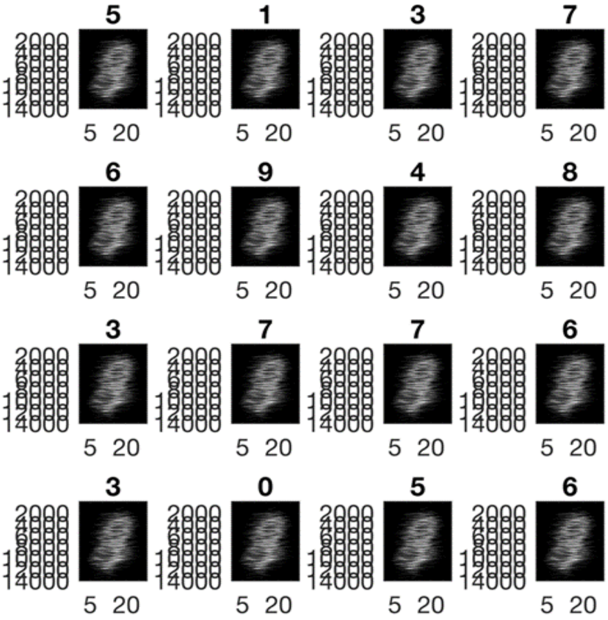

FIGURE 6. Sample images from the MNIST data base of handwritten digits.

features does not hurt the clustering performance but increased.

V. CONCLUSION

The k-means algorithm is generally the most used method in clustering. However, the k-means is always affected by initializations with equal importance for feature components under a given number of clusters. In this paper, we consider a mechanism in determining the number of clusters with feature-reduction behavior under unknown number of clusters for k-means clustering, named as Entropy-k-means. This clustering algorithm provides an alternative technique to find an optimal number of clusters with a feature reduction schema. Furthermore, the Entropy-k-means can also reduce computational times. This is due to the fact that feature reduction schema during clustering processes is successfully worked for finding the optimal number of clusters. For examining the efficiency of the proposed Entropy-k-means clustering algorithm, it is compared with the original k-means, WKM, EWKM, C-FS, and U-k-means clustering algorithms. The comparisons are also made by implementing four validity indices in the original k-means, WKM, and EWKM. The comparison results show that the proposed Entropy-k-means algorithm has better performance and can simultaneously find the optimal number of clusters with feature-reduction behaviors. However, the proposed Entropy-k-means algorithm can only handle single view data. Since internet of things (IoT), social media, and big data grow rapidly, multi-view data become more popular. Thus, extensions of clustering algorithms to multi-view clustering become important. For multi-view clustering, sharing information between different views is also essential. In our future work, we will extend the proposed Entropy-k-means algorithm for clustering multi-view data sets with sharing information between different views and also automatically finding an optimal number of clusters without any parameter selection under free of initializations. On the other hand, except the methods used in the paper, some computational intelligence algorithms can be used to solve clustering problems, such as monarch butterfly optimization (MBO) [37], earthworm optimization algorithm (EWA) [38], elephant herding optimization (EHO) [39], moth search (MS) algorithm [40], Slime mould algorithm (SMA) [41], and Harris hawks optimization (HHO) [42]. The MBO, EWA, EHO, MS, SMA, and HHO algorithms are generally used for tackling optimization issues in terms of choosing an optimal parameter via operator selection for clustering algorithms. However, they cannot automatically determine the optimal number of clusters. We will further study these computational intelligence algorithms such that they can automatically find the optimal number of clusters with free of parameter selection.

REFERENCES

[1] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.

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

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

[4] L. Bai, X. Cheng, J. Liang, H. Shen, and Y. Guo, "Fast density clustering strategies based on the *k*-means algorithm," *Pattern Recognit.*, vol. 71, pp. 375–386, Nov. 2017.

[5] Q. Liu, X. Liu, J. Wu, and Y. Li, "An improved NSGA-III algorithm using genetic *K*-means clustering algorithm," *IEEE Access*, vol. 7, pp. 185239–185249, 2019.

[6] S.-H. Jung, H. Lee, and J.-H. Huh, "A novel model on reinforce *K*-means using location division model and outlier of initial value for lowering data cost," *Entropy*, vol. 22, no. 8, p. 902, Aug. 2020.

[7] H. Yu, G. Wen, J. Gan, W. Zheng, and C. Lei, "Self-paced learning for *K*-means clustering algorithm," *Pattern Recognit. Lett.*, vol. 132, pp. 69–75, Apr. 2020.

[8] Q. Han, J. Liu, Z. Shen, J. Liu, and F. Gong, "Vector partitioning quantization utilizing *K*-means clustering for physical layer secret key generation," *Inf. Sci.*, vol. 512, pp. 137–160, Feb. 2020.

[9] Q. Wang, J. Liu, B. Wei, W. Chen, and S. Xu, "Investigating the construction, training, and verification methods of *k*-means clustering fault recognition model for rotating machinery," *IEEE Access*, vol. 8, pp. 196515–196528, 2020.

[10] 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.

[11] 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.

[12] R. E. Kass and A. E. Raftery, "Bayes factors," *J. Amer. Statist. Assoc.*, vol. 90, no. 430, pp. 773–795, 1995.

[13] H. Bozdogan, "Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions," *Psychometrika*, vol. 52, no. 3, pp. 345–370, Sep. 1987.

[14] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, Jan. 1973.

[15] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[16] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

[17] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Statist., Theory Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[18] P. Dan and A. W. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proc. 17th Int. Conf. Mach. Learn. (ICML)*, vol. 1, Jun. 2000, pp. 727–734.

[19] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, no. 12, pp. 4831–4845, Dec. 2012.

[20] Z.-Y. Li, J.-H. Yi, and G.-G. Wang, "A new swarm intelligence approach for clustering based on krill herd with elitism strategy," *Algorithms*, vol. 8, no. 4, pp. 951–964, Oct. 2015.

[21] Y. Feng, S. Deb, G.-G. Wang, and A. H. Alavi, "Monarch butterfly optimization: A comprehensive review," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114418.

[22] R. Logesh, V. Subramaniyaswamy, V. Vijayakumar, X.-Z. Gao, and G.-G. Wang, "Hybrid bio-inspired user clustering for the generation of diversified recommendations," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 2487–2506, Mar. 2019.

[23] K. P. Sinaga and M.-S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020.

[24] 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.

[25] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 381–396, Mar. 2002.

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

[27] U. Fano, "Ionization yield of radiations. II. The fluctuations of the number of ions," *Phys. Rev.*, vol. 72, no. 1, pp. 26–29, Jul. 1947.

[28] C. L. Blake and C. J. Merz. (1998). *UCI Repository of Machine Learning Databases, a Huge Collection of Artificial and Real-World Data Sets*. [Online]. Available: http://www.ics.uci.edu/ mlearn/MLRepository.html

[29] A. A. Lubischew, "On the use of discriminant functions in taxonomy," *Biometrics*, vol. 18, no. 4, pp. 455–477, Dec. 1962.

[30] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacian faces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.

[31] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.

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

[33] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *J. Amer. Stat. Assoc.*, vol. 78, no. 383, pp. 553–569, Sep. 1983.

[34] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley, 1991.

[35] P. Jaccard, "Distribution de la flore alpine dans le bassin des Dranses et dans quelques régiones voisines," *Bull. De la Soc. Vaudoise des Sci. Naturel les*, vol. 18, pp. 1008–1018, 2016.

[36] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[37] G. G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 1995–2014, 2019.

[38] G.-G. Wang, S. Deb, and L. dos S. Coelho, "Earthworm optimisation algorithm: A bio-inspired metaheuristic algorithm for global optimisation problems," *Int. J. Bio-Inspired Comput.*, vol. 12, no. 1, pp. 1–22, Jan. 2018.

[39] G. G. Wang, S. Deb, and L. D. S. Coelho, "Elephant herding optimization," in *Proc. 3rd Int. Symp. Comput. Bus. Intell. (ISCBI)*, Bali, Indonesia, 2015, pp. 1–5.

[40] G.-G. Wang, "Moth search algorithm: A bio-inspired Metaheuristic algorithm for global optimization problems," *Memetic Comput.*, vol. 10, no. 2, pp. 151–164, Jun. 2018.

[41] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020.

[42] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.

KRISTINA P. SINAGA received the B.S. and M.S. degrees in mathematics from the University of Sumatera Utara, Indonesia, and the Ph.D. degree from the Department of Applied Mathematics, Chung Yuan Christian University, Taoyuan City, Taiwan. In 2020, she joined the Department of Master in Information System Management, BINUS Graduate Program, BINUS University, Indonesia, as a Lecturer. Her research interests include clustering and pattern recognition.

ISHTIAQ HUSSAIN received the M.Sc. degree in statistics from Quaid-i-Azam University, Islamabad, Pakistan, and the M.Phil. degree in statistics from Riphah International University, Islamabad. He is currently a Ph.D. Student with the Department of Applied Mathematics, Chung Yuan Christian University, Taoyuan City, Taiwan. His research interests include clustering algorithms, fuzzy clustering, and pattern recognition.

MIIN-SHEN YANG received the B.S. degree in mathematics from Chung Yuan Christian University (CYCU), Taoyuan City, Taiwan, in 1977, the M.S. degree in applied mathematics from 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 Department of Mathematics, 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 with the Department of Applied Mathematics, and the Director of Chaplain's Office. He is currently the Dean of the College of Science, CYCU. His research interests include applications of statistics, fuzzy clustering, soft computing, pattern recognition, and machine learning. He was an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS, from 2005 to 2011. He is also an Associate Editor of the *Applied Computational Intelligence and Soft Computing*, and an Editorial Board Member of Computer Science and Engineering section in the journal *Electronics* (MDPI).