IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 5, 2020, accepted April 16, 2020, date of publication April 20, 2020, date of current version May 13, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2988796*

# Unsupervised K-Means Clustering Algorithm

**KRISTINA P. SINAGA AND MIIN-SHEN YANG**
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, Taiwan, under Grant MOST 107-2118-M-033-002-MY2.

**ABSTRACT** The k-means algorithm is generally the most known and used clustering method. There are various extensions of k-means to be proposed in the literature. Although it is an unsupervised learning to clustering in pattern recognition and machine learning, the k-means algorithm and its extensions are always influenced by initializations with a necessary number of clusters a priori. That is, the k-means algorithm is not exactly an unsupervised clustering method. In this paper, we construct an unsupervised learning schema for the k-means algorithm so that it is free of initializations without parameter selection and can also simultaneously find an optimal number of clusters. That is, we propose a novel unsupervised k-means (U-k-means) clustering algorithm with automatically finding an optimal number of clusters without giving any initialization and parameter selection. The computational complexity of the proposed U-k-means clustering algorithm is also analyzed. Comparisons between the proposed U-k-means and other existing methods are made. Experimental results and comparisons actually demonstrate these good aspects of the proposed U-k-means clustering algorithm.

**INDEX TERMS** Clustering, K-means, number of clusters, initializations, unsupervised learning schema, Unsupervised k-means (U-k-means).

## I. INTRODUCTION

Clustering is a useful tool in data science. It is a method for finding cluster structure in a data set that is characterized by the greatest similarity within the same cluster and the greatest dissimilarity between different clusters. Hierarchical clustering was the earliest clustering method used by biologists and social scientists, whereas cluster analysis became a branch of statistical multivariate analysis [1], [2]. It is also an unsupervised learning approach to machine learning. From statistical viewpoint, clustering methods are generally divided as probability model-based approaches and nonparametric approaches. The probability model-based approaches follow that the data points are from a mixture probability model so that a mixture likelihood approach to clustering is used [3]. In model-based approaches, the expectation and maximization (EM) algorithm is the most used [4], [5]. For nonparametric approaches, clustering methods are mostly based on an objective function of similarity or dissimilarity measures, and these can be divided into hierarchical and partitional methods where partitional methods are the most used [2], [6], [7].

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

In general, partitional methods suppose that the data set can be represented by finite cluster prototypes with their own objective functions. Therefore, defining the dissimilarity (or distance) between a point and a cluster prototype is essential for partition methods. It is known that the k-means algorithm is the oldest and popular partitional method [1], [8]. The k-means clustering has been widely studied with various extensions in the literature and applied in a variety of substantive areas [9], [10], [11], [12]. However, these k-means clustering algorithms are usually affected by initializations and need to be given a number of clusters a priori. In general, the cluster number is unknown. In this case, validity indices can be used to find a cluster number where they are supposed to be independent of clustering algorithms [13]. Many cluster validity indices for the k-means clustering algorithm had been proposed in the literature, such as Bayesian information criterion (BIC) [14], Akaike information criterion (AIC) [15], Dunn's index [16], Davies-Bouldin index (DB) [17], Silhouette Width (SW) [18], Calinski and Harabasz index (CH) [19], Gap statistic [20], generalized Dunn's index (DNg) [21], and modified Dunn's index (DNs) [22].

For estimation the number of clusters, Pelleg and Moore [23] extended k-means, called X-means, by making local decisions for cluster centers in each iteration of k-means

with splitting themselves to get better clustering. 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, such as cluster validity indices and X-means, they use extra iteration steps outside the clustering algorithms. As we know, no work in the literature for k-means can be free of initializations, parameter selection and also simultaneously find the number of clusters. We suppose that this is due to its difficulty for constructing this kind of the k-means algorithm.

In this paper, we first construct a learning procedure for the k-means clustering algorithm. This learning procedure can automatically find the number of clusters without any initialization and parameter selection. We first consider an entropy penalty term for adjusting bias, and then create a learning schema for finding the number of clusters. The organization of this paper is as follows. In Section II, we review some related works. In Section III, we first construct the learning schema and then propose the unsupervised k-means clustering (U-k-means) with automatically finding the number of clusters. The computational complexity of the proposed U-k-means algorithm is also analyzed. In Section IV, several experimental examples and comparisons with numerical and real data sets are provided to demonstrate the effectiveness of the proposed U-k-means clustering algorithm. Finally, conclusions are stated in Section V.

## II. RELATED WORKS

In this section, we review several works that are closely related with ours. The k-means is one of the most popular unsupervised learning algorithms that solve the well-known clustering problem. Let $\mathbf{X} = \{x_1, \ldots, x_n\}$ be a data set in a $d$-dimensional Euclidean spac $\mathbb{R}^d$. Let $A = \{a_1, \ldots, a_c\}$ be the $c$ cluster centers. Let $\mathrm{z} = [z_{ik}]_{n\times c}$, where $z_{ik}$ is a binary variable (i.e. $z_{ik} \in \{0, 1\}$) indicating if the data point $x_i$ belongs to $k$-$th$ cluster, $k = 1, \cdots, c$. The k-means objective function is $J(\mathrm{z}, A) = \sum_{i=1}^{n}\sum_{k=1}^{c} z_{ik} \|x_i - a_k\|^2$. The k-means algorithm is iterated through necessary conditions for minimizing the k-means objective function $J(\mathrm{z}, A)$ with updating equations for cluster centers and memberships, respectively, as

$$a_k = \frac{\sum_{i=1}^{n} z_{ik}x_{ij}}{\sum_{i=1}^{n} z_{ik}} \text{ and}$$

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

where $\|x_i - a_k\|$ is the Euclidean distance between the data point $x_i$ and the cluster center $a_k$. There exists a difficult problem in k-means, i.e., it needs to give a number of clusters a priori. However, the number of clusters is generally unkown in real applications. Another problem is that the k-means algorithm is always affected by initializations.

To resolve the above issue for finding the number $c$ of cluster, cluster validity issues get much more attention. There are several clustering validity indices available for estimating the number $c$ of clusters. Clustering validity indices can be grouped into two major categories: external and internal [24]. External indices are used to evaluate clustering results by comparing cluster memberships assigned by a clustering algorithm with the previously known knowledge such as externally supplied class label [25], [26]. However, internal indices are used to evaluate the goodness of cluster structure by focusing on the intrinsic information of the data itself [27] so that we consider only internal indices. In the paper, these most widely used internal indices, such as original Dunn's index (DNo) [16], Davies-Bouldin index (DB) [17], Silhouette Width (SW) [18], Calinski and Harabasz index (CH) [19], Gap statistics [20], generalized Dunn's index (DNg) [21], and modified Dunn's index (DNs) [22] are chosen for finding the number of clusters and then compared with our proposed U-k-means clustering algorithm.

The DNo [16], DNg [21], and DNs [22] are supposed to be the simplest (internal) validity index where it compares the size of clusters with the distance between clusters. The DNo, DNg, and DNs indices are computed as the ratio between the minimum distance between two clusters and the size of the largest cluster, and so we are looking for the maximum value of index values. Davies-Bouldin index (DB) [17] measures the average similarity between each cluster and its most similar one. The DB validity index attempts to maximize these between cluster distances while minimizing the distance between the cluster centroid and the other data objects. The Silhouette value [18] is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from $-1$ to $+1$, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. Thus, positive and negative large silhouette widths (SW) indicate that the corresponding object is well clustered and wrongly clustered, respectively. Any objects with the SW validity index around zero are considered not to be clearly discriminated between clusters. The Gap statistic [20] is a cluster validity measure based upon a statistical hypothesis test. The gap statistic works by comparing the change in within-cluster dispersion with that expected under an appropriate reference null distribution at each value $c$. The optimal number of clusters is the smallest $c$.

For an efficient method about the number of clusters, X-means proposed by Pelleg and Moore [23], should be the most well-known and used in the literature, such as Witten *et al.* [28], and Guo *et al.* [29]. In X-means, Pelleg and Moore [23] extended k-means by making local decisions for cluster centers in each iteration of k-means with splitting themselves to get better clustering. Users only need to specify a range of cluster numbers in which the true cluster number reasonably lies and then a model selection, such as BIC, is used to do the splitting process. Although X-means has been the most used for clustering without given a number of clusters a priori, it still needs to specify a range of cluster numbers based on a criterion, such as BIC. On the other hand,

it is still influenced by initializations of algorithm. On the other hand, Rodriguez and Laio [30] proposed an approach based on the idea that cluster centers are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities, which they called as a clustering by fast search (C-FS) and find of density peaks. To identify the cluster centers, C-FS uses the heuristic approach of a decision graph. However, the performance of C-FS highly depends on two factors, i.e., local density $\rho_i$ and cutoff distance $\delta_i$.

## III. THE UNSUPERVISED K-MEANS CLUSTERING ALGORITHM

There always exists a difficult problem in the k-means algorithm and its extensions for a long history in the literature. That is, they are affected by initializations and require a given number of clusters a priori. We mentioned that the X-means algorithm has been used for clustering without given a number of clusters a priori, but it still needs to specify a range of number of clusters based on BIC, and it is still influenced by initializations. To construct the k-means clustering algorithm with free of initializations and automatically find the number of clusters, we use the entropy concept. We borrow the idea from the EM algorithm by Yang *et al.* [31]. We first consider proportions $\alpha_k$ in which the $\alpha_k$ term is seen as the probability of one data point belonged to the $k$th class. Hence, we use $-\ln \alpha_k$ as the information in the occurrence of one data point belonged to the $k$th 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 $\alpha_k$. When $\alpha_k = 1/c, \forall k = 1, 2, \ldots, c$, we say that there is no information about $\alpha_k$. At this point, we have the entropy achieve the maximum value. Therefore, we add this term to the k-means objective function $J(\mathrm{z}, A)$ as a penalty. We then construct a schema to estimate $\alpha_k$ by minimizing the entropy to get the most information for $\alpha_k$. 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 use $\sum_{k=1}^{c} \alpha_k \ln \alpha_k$ as a penalty term for the k-means objective function $J(\mathrm{z}, A)$. Thus, we propose a novel objective function as follows: $\beta \geq 0$

$$J_{UKM_1}(\mathrm{z}, A, \alpha) = \sum_{i=1}^{n}\sum_{k=1}^{c} z_{ik} \|x_i - a_k\|^2 - \beta n \sum_{k=1}^{c} \alpha_k \ln \alpha_k \tag{1}$$

In order to determine the number of clusters, we next consider another entropy term. We combine the variables membership $z_{ik}$ and the proportion $\alpha_k$. By using the basis of entropy theory, we suggest a new term in the form of $z_{ik} \ln \alpha_k$. Thus, we propose the unsupervised k-means (U-k-means) objective function as follows:

$$J_{U-k-means}(\mathrm{z}, A, \alpha) = \sum_{i=1}^{n}\sum_{k=1}^{c} z_{ik} \|x_i - a_k\|^2 - \beta n \sum_{k=1}^{c} \alpha_k \ln \alpha_k - \gamma \sum_{i=1}^{n}\sum_{k=1}^{c} z_{ik} \ln \alpha_k \tag{2}$$

We know that, when $\beta$ and $\gamma$ in Eq. (2) are zero, it becomes the original k-means. The Lagrangian of Eq. (2) is

$$\tilde{J}(\mathrm{z}, A, \alpha, \lambda) = \sum_{i=1}^{n}\sum_{k=1}^{c} z_{ik} \|x_i - a_k\|^2 - \beta n \sum_{k=1}^{c} \alpha_k \ln \alpha_k - \gamma \sum_{i=1}^{n}\sum_{k=1}^{c} z_{ik} \ln \alpha_k - \lambda \left( \sum_{k=1}^{c} \alpha_k - 1 \right) \tag{3}$$

We first take the partial derivative of the Lagrangian (3) with respect to $z_{ik}$, and setting them to be zero. Thus, the updating equation for $z_{ik}$ is obtained as follows:

$$z_{ik} = \begin{cases} 1 \text{ if } \|x_i - a_k\|^2 - \gamma \ln \alpha_k = \min\limits_{1 \leq k \leq c} \|x_i - a_k\|^2 - \gamma \ln \alpha_k \\ 0, \text{ otherwise.} \end{cases} \tag{4}$$

The updating equation for the cluster center $a_k$ is as follows:

$$a_k = \sum_{i=1}^{n} z_{ik} x_{ij} \Big/ \sum_{i=1}^{n} z_{ik} \tag{5}$$

We next take the partial derivative of the Lagrangian with respect to $\alpha_k$, we obtain $\frac{\partial \tilde{J}}{\partial \alpha_k} = -\beta n (\ln \alpha_k + 1) - \gamma \sum_{i=1}^{n} \frac{z_{ik}}{\alpha_k} - \lambda = 0$ and $-\beta n \alpha_k (\ln \alpha_k + 1) - \gamma \sum_{i=1}^{n} z_{ik} - \lambda \alpha_k = 0$. Thus, we have $-\sum_{k=1}^{c} n\beta \alpha_k \ln \alpha_k - \sum_{k=1}^{c} n\beta \alpha_k - \gamma \sum_{k=1}^{c}\sum_{i=1}^{n} z_{ik} - \sum_{k=1}^{c} \lambda \alpha_k = 0$ with $\lambda = -n\beta \sum_{k=1}^{c} \alpha_k \ln \alpha_k - n\beta - n\gamma$. We obtain $-\beta n \alpha_k (\ln \alpha_k + 1) - \gamma \sum_{i=1}^{n} z_{ik} - (-n\beta \sum_{k=1}^{c} \alpha_k \ln \alpha_k - n\beta - n\gamma) \alpha_k = 0$ and then we get the updating equation for $\alpha_k$ as follows:

$$\alpha_k^{(t+1)} = \sum_{i=1}^{n} z_{ik}/n + (\beta/\gamma)\alpha_k^{(t)} \left( \ln \alpha_k^{(t)} - \sum_{s=1}^{c} \alpha_s^{(t)} \ln \alpha_s^{(t)} \right) \tag{6}$$

where $t$ denotes the iteration number in the algorithm.

We should mention that Eq. (6) created above is important for our proposed U-k-means clustering method. In Eq. (6), $\sum_{s=1}^{c} \alpha_s \ln \alpha_s$ is the weighted mean of $\ln \alpha_k$ with the weights $\alpha_1, \ldots, \alpha_c$. For the $k$th mixing proportion $\alpha_k^{(t)}$, if $\ln \alpha_k^{(t)}$ is less than the weighted mean, then the new mixing proportion $\alpha_k^{(t+1)}$ will become smaller than the old $\alpha_k^{(t)}$. 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 in Figueiredo and Jain [32]. If $\alpha_k \leq 0$ or $\alpha_k < 1/n$ for some $1 \leq k \leq c^{(t)}$, they are considered to be illegitimate proportions. In this situation, we discard those clusters and then update the cluster number $c^{(t)}$ to be

$$c^{(t+1)} = c^{(t)} - \left| \left\{ \alpha_k^{(t+1)} \middle| \alpha_k^{(t+1)} < 1/n, k = 1, \ldots, c^{(t)} \right\} \right| \tag{7}$$

where $|\{\}|$ denotes the cardinality of the set $\{\}$. After updating the number of clusters $c$, the remaining mixing

proportion $\alpha_k^*$ and corresponding $z_{ik}^*$ need to be re-normalized by

$$\alpha_k^* = \alpha_k^* \Big/ \sum_{s=1}^{c^{(t+1)}} \alpha_s^* \tag{8}$$

$$z_{ik}^* = z_{ik}^* \Big/ \sum_{s=1}^{c^{(t+1)}} z_{is}^* \tag{9}$$

We next concern about the parameter learning of $\gamma$ and $\beta$ for the two terms of $\sum_{i=1}^{n}\sum_{k=1}^{c} z_{ik}\ln\alpha_k$ and $\sum_{k=1}^{c}\alpha_k\ln\alpha_k$. Based on some increasingly learning rates of cluster number with $e^{-c^{(t)}/100}$, $e^{-c^{(t)}/250}$, $e^{-c^{(t)}/500}$, $e^{-c^{(t)}/750}$, and $e^{-c^{(t)}/1000}$, it is seen that $e^{-c^{(t)}/100}$ decreases faster, but $e^{-c^{(t)}/500}$, $e^{-c^{(t)}/750}$ and $e^{-c^{(t)}/1000}$ decreases slower. We suppose that the parameter $\gamma$ should not decrease too slow or too fast, and so we set the parameter $\gamma$ as

$$\gamma^{(t)} = e^{-c^{(t)}/250} \tag{10}$$

Under competition schema setting, the algorithm can automatically reduce the number of clusters, and also simultaneously gets the estimates of parameters.

Furthermore, the parameter $\beta$ can help us control the competition. We discuss the variable $\beta$ as follows. We first apply the rule $-e^{-1} \le \alpha_k\ln\alpha_k < 0$. If $0 < \alpha_k \le 1 \forall k$, and let $E = \sum_{s=1}^{c}\alpha_s\ln\alpha_s < 0$, then we have $\alpha_k E = \alpha_k\sum_{s=1}^{c}\alpha_s\ln\alpha_s < 0$. Thus, we obtain

$$-e^{-1}\beta < \beta\alpha_k(\ln\alpha_k - \sum_{s=1}^{c}\alpha_s\ln\alpha_s) < \beta(-\alpha_k E) \tag{11}$$

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 $\alpha_k \le 0$, the left hand of inequality (14) must be larger than $-\max\{\alpha_k|\alpha_k < 1/2, k = 1, 2, \cdots, c\}$. We now have an elementary condition of $\beta$ as follows: $-e^{-1}\beta > -\max\{\alpha_k|\alpha_k < 1/2, k = 1, 2, \cdots, c\}$. Thus, we have Thus, we have $\beta < \max\{\alpha_k e|\alpha_k < 1/2, k = 1, 2, \cdots, c\} < e/2$. Therefore, to prevent $\beta$ from being too big, we can use $\beta \in [0, 1]$. Furthermore, if the difference between $\alpha_k^{(t+1)}$ and $\alpha_k^{(t)}$ is small, then $\beta$ must become large in order to enhance its competition. If the difference between $\alpha_k^{(t+1)}$ and $\alpha_k^{(t)}$ is large, then $\beta$ will become small to maintain stability. Thus, we define an updating equation for $\beta$ with

$$\beta = \sum_{k=1}^{c}\exp\{-\eta n|\alpha_k^{(t+1)} - \alpha_k^{(t)}|\}/c \tag{12}$$

where $\eta = \min\left\{1, 1/t^{\lfloor d/2-1\rfloor}\right\}$ and $\lfloor a\rfloor$ represents the largest integer that is no more than $a$ and $t$ denotes the iteration number in the algorithm.

On the other hand, we consider the inequations

$$\max_{1\le k\le c}\alpha_k^{(t+1)} \le \max_{1\le k\le c}\left(\tfrac{1}{n}\sum_{i=1}^{n} z_{ik}\right) + \tfrac{\beta}{\gamma}\max_{1\le k\le c}\alpha_k^{(t)}\left(\ln\max_{1\le k\le c}\alpha_k^{(t)} - \sum_{s=1}^{c}\alpha_s^{(t)}\ln\alpha_s^{(t)}\right)$$

and

$$\max_{1\le k\le c}\left(\tfrac{1}{n}\sum_{i=1}^{n} z_{ik}\right) + \tfrac{\beta}{\gamma}\max_{1\le k\le c}\alpha_k^{(t)} \times \left(\ln\max_{1\le k\le c}\alpha_k^{(t)} - \sum_{s=1}^{c}\alpha_s^{(t)}\ln\alpha_s^{(t)}\right) < \max_{1\le k\le c}\left(\tfrac{1}{n}\sum_{i=1}^{n} z_{ik}\right) + \beta\left(-\left(\max_{1\le k\le c}\alpha_k^{(t)}\sum_{s=1}^{c}\alpha_s^{(t)}\ln\alpha_s^{(t)}\right)\right).$$

If

$$\max_{1\le k\le c}\left(\tfrac{1}{n}\sum_{i=1}^{n} z_{ik}\right) - \beta\max_{1\le k\le c}\alpha_k^{(t)}\sum_{s=1}^{c}\alpha_s^{(t)}\ln\alpha_s^{(t)} \le 1,$$

then the restriction of $\max_{1\le k\le c}\alpha_k^{(t+1)} \le 1$ is held, and then we obtain

$$\beta \le \left(1 - \max_{1\le k\le c}\left(\tfrac{1}{n}\sum_{i=1}^{n} z_{ik}\right)\right) \Big/ \left(-\max_{1\le k\le c}\alpha_k^{(t)}\sum_{s=1}^{c}\alpha_s^{(t)}\ln\alpha_s^{(t)}\right) \tag{13}$$

According to Eqs. (12) and (13), we can get

$$\beta^{(t+1)} = \min\left(\frac{\sum_{k=1}^{c}\exp(-\eta n\left|\alpha_k^{(t+1)} - \alpha_k^{(t)}\right|)}{c}, \frac{1 - \max_{1\le k\le c}\left(\tfrac{1}{n}\sum_{i=1}^{n} z_{ik}\right)}{(-\max_{1\le k\le c}\alpha_k^{(t)}\sum_{k'=1}^{c}\ln\alpha_{k'}^{(t)})}\right) \tag{14}$$

Because the $\beta$ 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 $a_k = x_k$, i.e. $c^{initial} = n$, and we use $\alpha_k = 1/c^{initial}, \forall k = 1, 2, ..., c^{initial}$ as initial mixing proportions. Thus, the proposed U-k-means clustering algorithm can be summarized as follows:

**U-k-means clustering algorithm**

Step 1: Fix $\varepsilon > 0$. Give initial $c^{(0)} = n$, $\alpha_k^{(0)} = 1/n$, $a_k^{(0)} = x_i$, and initial learning rates $\gamma^{(0)} = \beta^{(0)} = 1$. Set $t = 0$.

Step 2: Compute $z_{ik}^{(t+1)}$ using $a_k^{(t)}, \alpha_k^{(t)}, c^{(t)}, \gamma^{(t)}, \beta^{(t)}$ by (4).

Step 3: Compute $\gamma^{(t+1)}$ by (10).

Step 4: Update $\alpha_k^{(t+1)}$ with $z_{ik}^{(t+1)}$ and $\alpha_k^{(t)}$ by (6).

Step 5: Compute $\beta^{(t+1)}$ with $\alpha^{(t+1)}$ and $\alpha^{(t)}$ by (14).

Step 6: Update $c^{(t)}$ to $c^{(t+1)}$ by discard those clusters with $\alpha_k^{(t+1)} \le 1/n$ and adjust $\alpha_k^{(t+1)}$ and $z_{ik}^{(t+1)}$ by (8) and (9).
IF $t \ge 60$ and $c^{(t-60)} - c^{(t)} = 0$, THEN let $\beta^{(t+1)} = 0$.

Step 7: Update $a_k^{(t+1)}$ with $c^{(t+1)}$ and $z_{ik}^{(t+1)}$ by (5).

Step 8: Compare $a_k^{(t+1)}$ and $a_k^{(t)}$.
IF $\max_{1\le k\le c^{(t)}}\left\|a_k^{(t+1)} - a_k^{(t)}\right\| < \varepsilon$, THEN Stop.
ELSE $t = t+1$ and return to Step 2.

Before we analyze the computational complexity for the proposed U-k-means algorithm, we give a brief review

of another clustering algorithm that had also used the idea from the EM algorithm by Yang *et al.* [31]. This is the robust-learning fuzzy c-means (RL-FCM) proposed by Yang and Nataliani [33]. In Yang and Nataliani [33], they gave the RL-FCM objective function $J(\mathbf{U}, \alpha, \mathrm{A}) = \sum_{i=1}^{n}\sum_{k=1}^{c}\mu_{ik}\|x_i - a_k\|^2 - r_1\sum_{i=1}^{n}\sum_{k=1}^{c}\mu_{ik}\ln\alpha_k + r_2\sum_{i=1}^{n}\sum_{k=1}^{c}\mu_{ik}\ln\mu_{ik} - r_3 n\sum_{k=1}^{c}\alpha_k\ln\alpha_k$ with $\mu_{ik}$, not binary variables, but fuzzy c-memberships with $0 \le \mu_{ik} \le 1$ and $\sum_{k=1}^{c}\mu_{ik} = 1$ to indicate fuzzy memberships for the data point $x_i$ belonging to *k-th* cluster. If we compare the proposed U-k-means objective function $J_{U-k-means}(\mathrm{z}, A, \alpha)$ with the RL-FCM objective function $J(\mathbf{U}, \alpha, \mathrm{A})$, we find that, except $\mu_{ik}$ and $z_{ik}$ with different membership representations, the RL-FCM objective function $J(\mathbf{U}, \alpha, \mathrm{A})$ in Yang and Nataliani [33] gave more extra terms and parameters and so the RL-FCM algorithm is more complicated than the proposed U-k-means algorithm with more running time. For experimental results and comparisons in the next section, we make more comparisons of the proposed U-k-means algorithm with the RL-FCM algorithm. We also analyze the computational complexity for the U-k-means algorithm. In fact, the U-k-means algorithm can be divided into three parts: (1) Compute the hard membership partition $z_{ik}$ with $O(ncd)$; (2) Compute the mixing proportion $\alpha_k$ with $O(nc)$; (3) Update the cluster center $a_k$ with $O(n)$. The total computational complexity for the U-k-means algorithm is $O(ncd)$, where $n$ is the number of data points, $c$ is the number of clusters, and $d$ is the dimension of data points. Compared with the RL-FCM algorithm [33], the RL-FCM has the total computational complexity fwith $O\left(nc^2d\right)$.

## IV. EXPERIMENTAL RESULTS AND COMPARISONS

In this section we give some examples with numerical and real data sets to demonstrate the performance of the proposed U-k-means algorithm. We show these unsupervised learning behaviors to get the best number $c^*$ of clusters for the U-k-means algorithm. Generally, most clustering algorithms, including k-means, are employed to give different numbers of clusters with associated cluster memberships, and then these clustering results are evaluated by multiple validity measures to determine the most practically plausible clustering results with the estimated number of clusters [13]. Thus, we will first compare the U-k-means algorithm with the seven validity indices, DNo [16], DNg [21], DNs [22], Gap statistic (Gap-stat) [20], DB [17], SW [18] and CH [19]. Furthermore, the comparisons of the proposed U-k-means with k-means [8], robust EM [31], clustering by fast search (C-FS) [30], X-means [23], and RL-FCM [33] are also made. For measuring clustering performance, we use an accuracy rate (AR) with $AR = \sum_{k=1}^{c} n(c_k)/n$, 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. The larger AR is, the better clustering performance is.

*Example 1:* In this example, we use a data set of 400 data points generated from the 2-variate 6-component Gaussian mixture model $f(x;\alpha,\theta) = \sum_{k=1}^{c}\alpha_k f(x;\theta_k)$ with parameters $\alpha_k = 1/6, \forall k$, $\mu_1 = \begin{pmatrix}5 & 2\end{pmatrix}^T$, $\mu_2 = \begin{pmatrix}3 & 4\end{pmatrix}^T$, $\mu_3 = \begin{pmatrix}8 & 4\end{pmatrix}^T$, $\mu_4 = \begin{pmatrix}6 & 6\end{pmatrix}^T$, $\mu_5 = \begin{pmatrix}10 & 8\end{pmatrix}^T$, $\mu_6 = \begin{pmatrix}7 & 10\end{pmatrix}^T$, and $\sum_1 = \cdots = \sum_6 = \begin{pmatrix}0.4 & 0\\ 0 & 0.4\end{pmatrix}$ with 2 dimensions and 6 clusters, as shown in Fig. 1(a). We implement the proposed U-k-means clustering algorithm for the data set of Fig. 1(a) in which it obtains the correct number $c^* = 6$ of clusters with $AR$=1.00, as shown in Fig. 1(f), after 11 iterations. These validity indices of CH, SW, DB, Gap statistic, DNo, DNg, and DNs are shown in Table 1. All indices give the correct number $c^* = 6$ of clusters, except DNg.

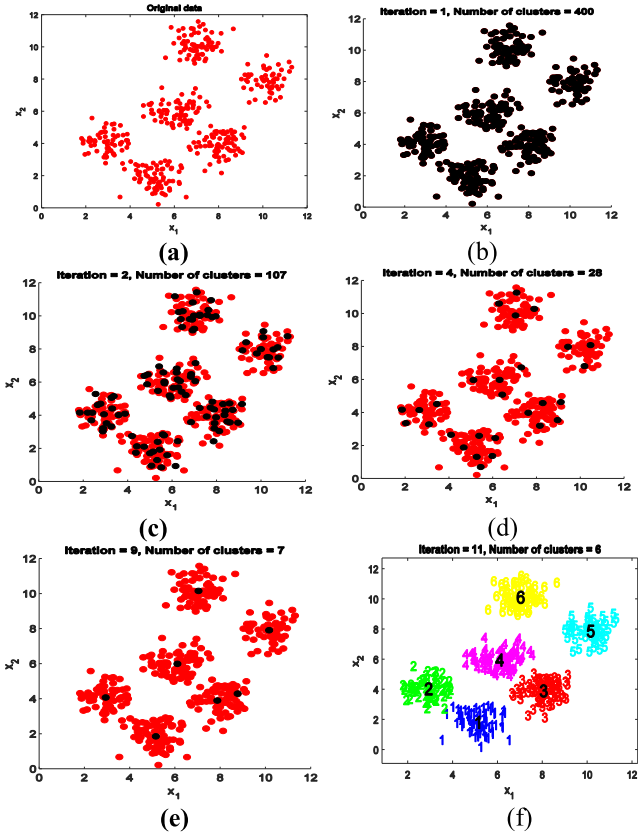
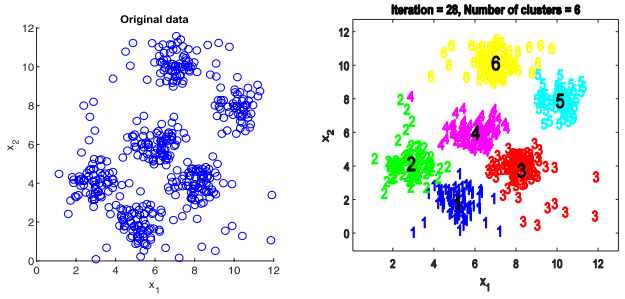



**FIGURE 1.** (a) Original data set; (b)-(e) Processes of the U-k-means after 1, 2, 4, and 9; (f) Convergent results.

Moreover, we consider the data set with noisy points to show the performance of the proposed U-k-means algorithm under noisy environment. We add 50 uniformly noisy points to the data set of Fig. 1(a), as shown in Fig. 2(a). By implementing the U-k-means algorithm on the noisy data set of Fig. 2(a), it still obtains the correct number $c^* = 6$ of clusters after 28 iterations with $AR$=1.00, as shown in Fig. 2(b). These validity index values of CH, SW, DB, Gap-stat, DNo, DNg, and DNs for the noisy data set of Fig. 2(a) are shown in Table 2. The five validity indices of CH, DB, Gap-stat, DNo and DNs give the correct number of clusters. But, SW and DNg give the incorrect numbers of clusters.

**TABLE 1.** Validity index values of CH, SW, DB, Gap-stat, DNo, DNg, and DNs for the data set of Fig. 1(a).

| c | Validity index values | | | | | | |
|---|---|---|---|---|---|---|---|
| | CH | SW | DB | Gap-stat | DNo | DNg | DNs |
| 2 | 0.511 | 0.680 | 0.772 | 0.183 | 0.008 | **6.587** | 0.001 |
| 3 | 0.553 | 0.649 | 0.866 | 0.388 | 0.047 | 2.603 | 0.019 |
| 4 | 0.605 | 0.715 | 0.700 | 0.469 | 0.040 | 1.603 | 0.016 |
| 5 | 0.754 | 0.743 | 0.571 | 0.619 | 0.041 | 4.619 | 0.020 |
| 6 | **1.277** | **0.838** | **0.483** | **1.067** | **0.102** | 4.635 | **0.048** |
| 7 | 1.155 | 0.773 | 0.634 | 0.991 | 0.060 | 0.794 | 0.022 |
| 8 | 1.054 | 0.703 | 0.808 | 0.930 | 0.030 | 0.571 | 0.004 |



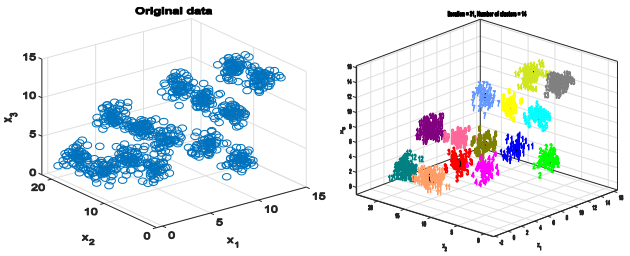

(a) (b)

**FIGURE 2.** 6-cluster dataset with 50 noisy points; (b) Final results from U-k-means.

**TABLE 2.** Validity index values of CH, SW, DB, Gap-stat, DNo, DNg, and DNs for the noisy data set.

| c | Criterion values | | | | | | |
|---|---|---|---|---|---|---|---|
| | CH | SW | DB | Gap stat | DNo | DNg | DNs |
| 2 | 508.2 | 0.662 | 0.792 | 0.827 | 0.005 | **4.547** | 0.001 |
| 3 | 523.4 | 0.615 | 0.913 | 0.835 | 0.034 | 1.828 | 0.012 |
| 4 | 526.9 | 0.655 | 0.748 | 0.719 | 0.033 | 1.752 | 0.011 |
| 5 | 637.9 | 0.697 | 0.607 | 0.914 | 0.028 | 3.397 | 0.008 |
| 6 | **902.6** | 0.771 | **0.538** | **1.237** | **0.052** | 1.502 | **0.013** |
| 7 | 864.4 | **0.783** | 0.558 | 1.173 | 0.042 | 0.797 | 0.008 |
| 8 | 837.7 | 0.766 | 0.666 | 1.143 | 0.019 | 0.497 | 0.002 |

*Example 2:* In this example, we consider a data set of 800 data points generated from a 3-variate 14-component Gaussian mixture with 800 data points with 3 dimensions and 14 clusters, as shown in Fig. 3(a). To estimate the number $c$ of clusters, we use CH, SW, DB, Gap-stat, DNo, DNg, and DNs. To create the results of the seven validity indices, we consider the k-means algorithm with 25 different initializations. These estimated numbers of clusters from CH, SW, DB, Gap statistic, DNo, DNg, and DNs with percentages are shown in Table 3. It is seen that all validity indices can give the correct number $c^* = 14$ of clusters, except DNg, where the Gap-stat index gives the highest percentage of the correct number $c^* = 14$ of clusters with 64%. We also implement the proposed U-k-means for the data set, and then compare it with the R-EM, C-FS, k-means with the true number of clusters, X-means, and RL-FCM clustering algorithms. We mention that U-k-means, R-EM, and RL-FCM are free of parameter selection, but others are dependent on parameter selection for finding the number of clusters. Table 4 shows the comparison




(a) (b)

**FIGURE 3.** 14-cluster dataset; (b) Final results from U-k-means.

**TABLE 3.** Results of the seven validity indices.

| True c | Optimal number of clusters | | | | | | |
|---|---|---|---|---|---|---|---|
| | CH | SW | DB | Gap-stat | DNo | DNg | DNs |
| **14** | 14 (60%) | 14 (60%) | 14 (60%) | **14 (64%)** | 14 (20%) | 2, 4, 5, 10, 11 | 14 (20%) |

results of the U-k-means, R-EM, C-FS, k-means with the true cluster number $c = 14$, X-means, and RL-FCM algorithms. Note that C-FS, k-means with the true number of clusters, and X-means algorithms are dependent of initials or parameter selection, and so we consider their average AR (AV-AR) under different initials or parameter selection. From Table 4, it is seen that the proposed U-k-means, R-EM, and RL-FCM clustering algorithms are able to find the correct number of clusters $c^* = 14$ with *AR=1.00*. While C-FS obtained the correct $c^* = 14$ with 96% and *AV-AR=0.9772*. The k-means with the true $c$ gave *AV-AR=0.8160*. The X-means obtained the correct $c^* = 14$ with 76% and *AV-AR=1.00*. Note that the numbers in parentheses indicate the percentage in obtaining the correct number of clusters for clustering algorithms under 25 different initial values.

*Example 3:* To examine the effectiveness of the proposed U-k-means for finding the number of clusters, we generate a data set of 900 data points from a 20-variate 6-component Gaussian mixture model. The mixing proportions, mean values and covariance matrices of the Gaussian mixture model are listed in Table 5. The validity indices of CH, SW, DB, Gap-stat, DNo, DNg, and DNs are used to estimate the number $c$ of clusters. The k-means algorithm with 25 different initializations are considered to create the results of the seven validity indices. These estimated numbers of clusters from the seven validity indices with percentages are shown in Table 6 where the parentheses are indicating the percentages of validity indices in giving the correct number of clusters under 25 different initial values. It is seen that CH, SW, and Gap-stat give the correct number $c^* = 6$ of clusters with the highest percentage. We also implemented the U-k-means and compare it with R-EM, C-FS, k-means with the true number $c$, X-means, and RL-FCM algorithms. The obtained numbers of clusters and ARs of these algorithms are shown in Table 7. As it can be seen, the proposed U-k-means, C-FS and X-means correctly find the number of clusters for the data set. The R-EM and RL-FCM underestimate the number of

**TABLE 4.** Results of U-k-means, R-EM, C-FS, k-means with the true c, X-means, and RL-FCM for the data set of Fig. 3(a).

| True c | U-k-means | | R-EM | | C-FS | | k-means with true c | X-means | | RL-FCM | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | c* | AR | c* | AR | c* | AV-AR | AV-AR | c* | AV-AR | c* | AR |
| **14** | **14** | **1.00** | **14** | **1.00** | **14 (96%)** | 0.9772 | 0.8160 | **14 (76%)** | **1.00** | **14** | **1.00** |

**TABLE 5.** Mixing proportions, mean values and covariance matrices of Example 3.

| Mixing proportions | Mean values | covariance matrix |
|---|---|---|
| $\alpha_1 = 0.2$ | $\mu_1 = (2\ \ 4\ \ 6\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 3\ \ 5\ \ 0\ \ 0\ \ 1)$ | |
| $\alpha_2 = 0.3$ | $\mu_2 = (0\ \ 1\ \ 3\ \ 5\ \ 0.1\ \ 0.1\ \ 0.5\ \ 0.5\ \ 0\ \ 0\ \ 2\ \ 4\ \ 3\ \ 1\ \ 1\ \ 1\ \ 0.25\ \ 0.5\ \ 0.7\ \ 2.5)$ | |
| $\alpha_3 = 0.1$ | $\mu_3 = (5\ \ 5\ \ 5\ \ 5\ \ 4\ \ 4\ \ 4\ \ 4\ \ 6\ \ 6\ \ 6\ \ 6\ \ 8\ \ 8\ \ 8\ \ 8\ \ 1\ \ 1\ \ 1\ \ 1)$ | |
| $\alpha_4 = 0.1$ | $\mu_4 = (2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 3\ \ 3\ \ 3\ \ 3\ \ 3\ \ 7\ \ 7\ \ 7\ \ 7\ \ 7)$ | $\sum_k = I_{[20\times 20]}$ |
| $\alpha_5 = 0.2$ | $\mu_5 = (1.25\ \ 1.3\ \ 1.45\ \ 1.5\ \ 2.25\ \ 2.3\ \ 2.45\ \ 2.5\ \ 1\ \ 1\ \ 1\ \ 1\ \ 3\ \ 3\ \ 3\ \ 3\ \ 2\ \ 2\ \ 2\ \ 2)$ | |
| $\alpha_6 = 0.1$ | $\mu_6 = (0\ \ 0\ \ 1\ \ 1\ \ 0.5\ \ 0.5\ \ 2.5\ \ 2.5\ \ 5\ \ 5\ \ 1\ \ 1\ \ 5\ \ 5\ \ 0\ \ 0\ \ 0.75\ \ 1.5\ \ 3.5\ \ 5.5)$ | |

**TABLE 6.** Results of the seven validity indices for the data set of Example 3.

| True c | Optimal number of clusters obtains by | | | | | | |
|---|---|---|---|---|---|---|---|
| | CH | SW | DB | Gap-stat | DNo | DNg | DNs |
| 6 | **6 (88%)** | **6 (88%)** | 2, 3 | **6 (88%)** | 6 (16%) | 6 (8%) | 6 (12%) |

**TABLE 7.** Results of U-k-means, R-EM, C-FS, k-means with the true c, X-means, RL-FCM for Example 3.

| True c | U-k-means | | R-EM | | C-FS | | K-means with true c | X-means | | RL-FCM | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | c* | AR | c* | AR | c* | AR | AV-AR | c* | AR | c* | AR |
| 6 | **6** | **1.00** | 3 | - | **6 (84%)** | 0.8155 | 0.7833 | **6 (100%)** | **1.00** | 3 | - |

**TABLE 8.** Results of U-k-means, R-EM, C-FS, k-means with the true c, X-means, RL-FCM for Example 4.

| True c | U-k-means | | R-EM | | C-FS | | k-means with true c | X-means | | RL-FCM | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | c* | AR | c* | AR | c* | AV-AR | AV-AR | c* | AV-AR | c* | AR |
| 9 | **9** | **1.00** | 12 | - | **9 (96%)** | 0.7641 | 0.9190 | 2 | - | 2 | - |

clusters for the data set. Both U-k-means and X-means get the best AR.

*Example 4:* In this example, we consider a synthetic data set of non-spherical shape with 3000 data points, as shown in Fig. 4(a). The U-k-means is implemented for this data set with the clustering results as shown in Figs. 4(b)-4(f). The U-k-means algorithm decreases the number of clusters from 3000 to 2132 after the iteration is implemented once. From Figs. 4(b)-4(f), it is seen that the U-k-means algorithm exhibits fast decreasing for the number of clusters. After 11 iterations, the U-k-means algorithm obtains its convergent result with $c* = 9$ and AR= 1.00, as shown in Fig. 4(f). We next compare the proposed U-k-means algorithm with R-EM, C-FS, k-means with true $c$, X-means, and RL-FCM. All the experiments are performed 25 times with parameter selection where the average AR results under the correct number of cluster are reported in Table 8. As shown in Table 8, U-k-means gives the correct number $c* = 9$ of

**TABLE 9.** Descriptions of the eight data sets used in Example 5.

| Dataset | Feature Characteristics | Number $c$ of clusters | Number $n$ of instances | Number $d$ of features |
|---|---|---|---|---|
| Iris | Real | 3 | 150 | 4 |
| Seeds | Real | 3 | 210 | 7 |
| Australian | Categorical, Integer, Real | 2 | 690 | 14 |
| Flowmeter D | Real | 4 | 180 | 43 |
| Sonar | Real | 2 | 208 | 60 |
| Wine | Integer, Real | 3 | 178 | 13 |
| Horse | Categorical, Integer, Real | 2 | 368 | 27 |
| Waveform (Version 1) | Real | 3 | 5000 | 21 |

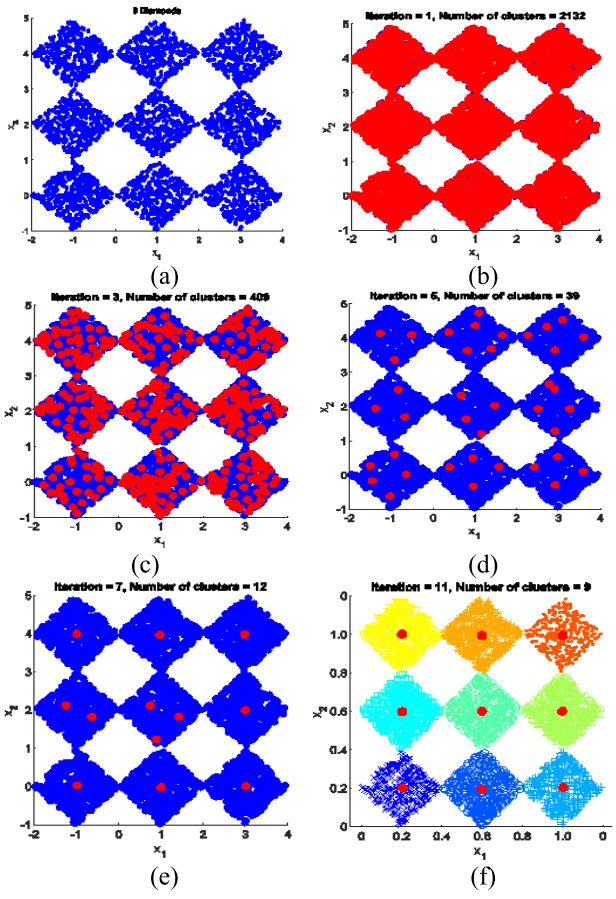



**FIGURE 4.** (a) 9-diamonds data set; (b)-(e) Results of the U-k-means after 1, 3, 5, and 7 iterations; (f) Final results of the U-k-means after 11 iterations.

clusters with $AR$=*1.00*, followed by k-means with true $c$=*9* achieves an average $AR$=*0.9190* and C-FS with $c* =9$ (96%) achieves average $AR$=*0.7641*. While R-EM overestimates the number of clusters with $c* =12$, but X-means and RL-FCM underestimate the number of clusters with $c* =2$.

We next consider real data sets. These data sets are from the UCI Machine Learning Repository [34].

*Example 5:* In this example, we use the eight real data sets from UCI Machine Learning Repository [34], known as Iris, Seeds, Australian credit approval, Flowmeter D, Sonar, Wine, Horse, and waveform (version 1). Detailed information on these data sets such as feature characteristics, the number $c$ of classes, the number $n$ of instances and the number $d$ of features is listed in Table 9. Since data features in Seeds, Flowmeter D, Wine and Waveform (version 1) are distributed in different ranges and data features in Australian (credit approval) are mixed feature types, we first preprocess data matrices using matrix factorization technique [35]. This preprocessed technique can give these data in uniform to get good quality clusters and improve accuracy rates of clustering algorithms. Clustering results from the U-k-means, R-EM, C-FS, k-means with the true $c$, k-means+Gap-stat, X-means, and RL-FCM algorithms for different real data sets are shown in Table 10, where the best results are presented in boldface. It is seen that the proposed U-k-means gives the best result in estimating the number $c$ of clusters and accuracy rate among them except for Australian data. The C-FS algorithm gives the corrected numbers of clusters for Iris, Seeds, Australian, Flowmeter D, Sonar, Wine, and Horse data sets while it underestimates the number of clusters for the waveform data set with $c* =2$. The X-means algorithm only obtains the correct number of clusters for Seeds, Wine and Horse data sets. The R-EM obtains the correct number of clusters for Iris and Seeds data sets. The k-means+Gap-stat only obtains a correct number of clusters for the Seed data set. The RL-FCM algorithm obtains the correct number of clusters for the Iris, Seeds and Waveform (version 1) data sets. Note that the results in parentheses are the percentages of algorithms to get the correct number $c$ of clusters.

*Example 6:* In this example, we use the six medical data sets from the UCI Machine Learning Repository [34], known as SPECT, Parkinsons, WPBC, Colon, Lung and Nci9. Detailed descriptions on these data sets with feature characteristics, the number $c$ of classes, the number $n$ of instances and the number $d$ of features are listed in Table 11. In this experiment, we first preprocess the SPECT, Parkinson, WPBC, Colon, and Lung data sets using the matrix

**TABLE 10.** Clustering results from various algorithms for different real data sets with the best results in boldface.

| Data set | True c | U-K-Means | | R-EM | | C-FS | | k-Means with true c | k-means + Gap-stat | | X-means | | RL-FCM | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | c* | AR | c* | AV-AR | c* | AR | | c* | AR | c* | AV-AR | c* | AV-AR |
| Iris | 3 | **3** | 0.8933 | 3 | 0.8600 | **3 (84%)** | 0.7521 | 0.7939 | 4, 5 | - | 2 | - | **3** | **0.9067** |
| Seeds | 3 | **3** | **0.9048** | 3 | 0.8476 | **3 (100%)** | 0.7944 | 0.8864 | **3 (100%)** | 0.8952 | **3 (100%)** | 0.890 | 3 | 0.8952 |
| Australian | 2 | **2** | **0.5551** | 4 | - | **2 (100%)** | **0.5551** | **0.5551** | 6 | - | 6 | - | 26 | - |
| Flowmeter D | 4 | **4** | **0.6056** | 3 | - | **4 (100%)** | 0.4338 | 0.5833 | 9, 10 | - | 10 | - | 13 | - |
| Sonar | 2 | **2** | **0.5337** | 5 | - | **2 (80%)** | 0.4791 | 0.4791 | 5, 6 | - | 3, 4 | - | 4 | - |
| Wine | 3 | **3** | **0.7022** | 2 | - | **3 (100%)** | 0.5557 | 0.6851 | 2 | - | **3 (64%)** | 0.62 | 2 | - |
| Horse | 2 | **2** | **0.6576** | 4,6,8, 10, 14 | - | **2 (100%)** | 0.6033 | 0.6055 | 3 | - | **2 (88%)** | 0.50 | 7 | - |
| Waveform (Version 1) | 3 | **3** | **0.4020** | 1 | - | 2 | - | 0.3900 | 1 | - | 8 | - | **3** | 0.3972 |

**TABLE 11.** Descriptions of the six medical data sets used in Example 6.

| Dataset | Feature Characteristics | Number $c$ of clusters | Number $n$ of instances | Number $d$ of features |
|---|---|---|---|---|
| SPECT | Categorical | 2 | 187 | 22 |
| Parkinsons | Real | 2 | 195 | 22 |
| WPBC | Real | 2 | 198 | 33 |
| Colon | Discrete, Binary | 2 | 62 | 2000 |
| Lung | Continous, Multi-class | 5 | 203 | 3312 |
| Nci9 | Discrete, Multi-class | 9 | 60 | 9712 |

**TABLE 12.** Results from various algorithms for the six medical data sets with the best results in boldface.

| DATA SET | TRUE C | U-K-MEANS | | R-EM | | C-FS | | K-MEANS WITH TRUE C | K-MEANS + GAP-STAT | | X-MEANS | | RL-FCM | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | C* | AR | C* | AV-AR | C* | AR | | C* | AR | C* | AV-AR | C* | AV-AR |
| SPECT | 2 | **2** | **0.920** | **2** | 0.562 | **2(84%)** | 0.8408 | 0.5262 | 5, 6 | - | **2** (100%) | 0.5119 | **2** | 0.588 |
| PARKINSONS | 2 | **2** | **0.754** | 1 | - | **2 (100%)** | 0.7436 | 0.5183 | **2** (100%) | 0.62 | 4, 5 | - | **2** | **0.754** |
| WPBC | 2 | **2** | **0.763** | 198 | - | **2 (100%)** | 0.7576 | 0.5927 | 4 | - | 3 | - | **2** | **0.763** |
| COLON | 2 | **2** | **0.645** | - | - | **2 (100%)** | 0.5813 | **0.4768** | 4 | - | **2** (100%) | 0.45 | 62 | - |
| LUNG | 5 | **5** | **0.788** | - | - | **5 (100%)** | 0.6859 | 0.6818 | 4, 6, 7, 8 | - | 2 | - | 9 | - |
| NCI9 | 9 | 8 | - | - | - | 2, 4 | - | 0.32 | 2 | - | 2 | - | 60 | - |

factorization technique. We also conduct experiments to compare the proposed U-k-means with R-EM, C-FS, k-means with the true c, k-means+Gap-stat, X-means, and RL-FCM. The results are shown in Table 12. For C-FS, k-means with the true c, k-means+Gap-stat and X-means, we make experiments with 25 different initializations, and report their results with the average AR (AV-AR) and the percentages of algorithms to get the correct number $c$ of clusters, as shown in Table 12. It is seen that the proposed U-k-means gets the correct number of clusters for SPECT, Parkinsons, WPBC, Colon, and Lung. While for the Nci9 data set, the U-k-means algorithm gets the number of clusters with $c* = 8$ which is very closed to the true $c$=9. In terms of AR, the U-k-means algorithm significantly performs much better than others. The R-EM algorithm estimates the correct number of clusters on SPECT. However, it underestimates the number of clusters on Parkinsons, and overestimates the number of clusters on WPBC. We also reported that the results of R-EM on Colon, Lung and Nci9 data sets are missing because the probability of one data point belonged to the $k$th class on these data sets are known as illegitimate proportions at the first iteration.

**TABLE 13.** Clustering results from various algorithms for different real data sets with the best results in boldface.

| Data set | True c | FU-k-means | | R-EM | | C-FS | | k-means with true c | X-means | | RL-FCM | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | c* | AR | c* | AR | c* | AV-AR | AV-AR | c* | AV-AR | c* | AV-AR |
| Yale Face | 15 | 16 | - | - | - | 12 | - | 0.34 | 2, 3 | - | 2 | - |

**TABLE 14.** Results of U-k-means, R-EM, C-FS, k-means with the true c, X-means, and RL-FCM for the 100 images sample of the CIFAR-10 data set.

| Data set | True c | U-k-means | | R-EM | | C-FS | | k-means with true c | X-means | | RL-FCM | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | c* | AV-AR | c* | AR | c* | AV-AR | AV-AR | c* | AV-AR | c* | AV-AR |
| CIFAR-10 | 10 | **10 (42.5%)** | **0.311** | - | - | **10 (3.03%)** | 0.295 | 0.280 | 2 | - | - | - |

The C-FS algorithm presents better than k-means+ Gap-stat and X-means. The RL-FCM algorithm estimates the correct number of clusters c for the SPECT, Parkinsons, and WPBC data sets. While RL-FCM overestimates the number of clusters on Colon, Lung and Nci9 with $c* = 62$, $c* = 9$, and $c* = 60$, respectively.

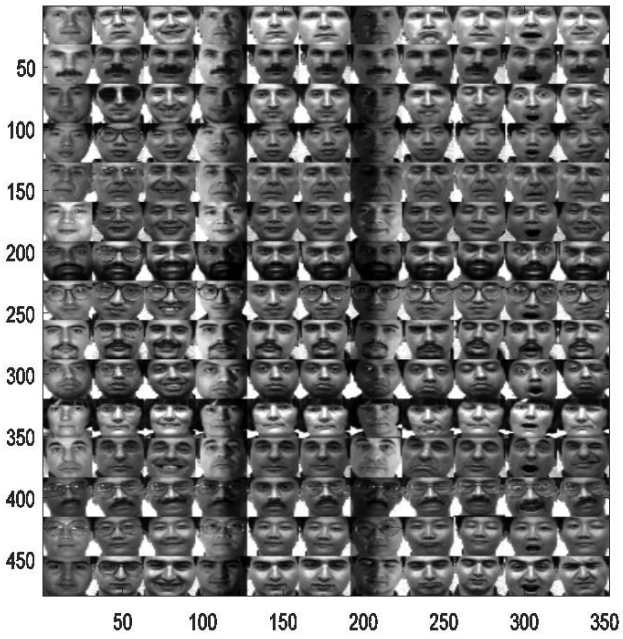



**FIGURE 5.** Yale Face 32 × 32.

**FIGURE 6.** The 100 Images Sample of CIFAR-10.

*Example 7:* In this example, we apply the U-k-means clustering algorithm for Yale Face 32 × 32 data set, as shown in Fig. 5. It has 165 grayscale images in GIF format of 15 individuals [36]. There are 11 images per subject with different facial expression or configuration: center-light, with/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. In the experiment, we use 135 images of 165 grayscale images. The results from different algorithms are shown in Table 13. From Table 13, although U-k-means cannot correctly estimate the true number $c=15$ of clusters for the Yale face data set, but it gives the number of clusters $c* = 16$ in which it is closed to the true $c=15$. The R-EM algorithm is missing because the probability of one data point belonged to the $k$th class on this data set are known as

**TABLE 15. Comparison of average running times (in seconds) of U-k-means, R-EM, C-FS, k-means with the true c, and RL-FCM for all data sets. The fastest running times are highlighted.**

| Data sets | U-k-means | R-EM | C-FS | RL-FCM |
|---|---|---|---|---|
| Synthetic Data sets | | | | |
| Example 1 | **0.3842** | 4.8921 | 5.8050 | 1.3688 |
| Example 2 | **2.9185** | 13.6157 | 7.3559 | 6.0444 |
| Example 3 | **2.1625** | 2.7938 | 10.2817 | 3.2924 |
| Example 4 | 117.2595 | 742.14 | **35.6417** | 438.047 |
| UCI Data sets | | | | |
| Iris | **0.2159** | 1.1842 | 6.31581 | 0.4184 |
| Seeds | **0.1455** | 2.0400 | 5.2702 | 0.4472 |
| Australian | **2.0434** | 5.8039 | 6.1772 | 2.3829 |
| Flowmeter D | **0.2834** | 0.6969 | 5.6230 | 0.3054 |
| Sonar | **0.1747** | 0.3148 | 5.8564 | 0.3963 |
| Wine | **0.1980** | 1.4837 | 5.8094 | 0.3060 |
| Horse | **0.6072** | 2.5989 | 5.3442 | 0.6272 |
| Waveform | 330.748 | - | **113.8162** | 474.165 |
| Medical Data sets | | | | |
| SPECFT | **0.1354** | 0.7211 | 5.9079 | 0.3411 |
| Parkinsons | **0.1487** | 0.5856 | 4.9534 | 0.3958 |
| WPBC | **0.1512** | 0.7922 | 5.2152 | 0.4036 |
| Colon | **0.1653** | - | 4.9608 | 0.2676 |
| Lung | **1.1239** | - | 5.2485 | 1.1167 |
| Nci9 | **0.6186** | - | 6.4794 | 0.5096 |
| Image Data sets | | | | |
| Yale Face 32x32 | **0.3741** | - | 5.9634 | 0.4286 |
| CIFAR-10 | **2.6561** | - | 6.4500 | - |

illegitimate proportions at the first iteration. The C-FS gives $c* = 12$ and X-means gives $c* = 2$ or $3$. The k- means clustering with the true $c=15$ gives $AV\text{-}AR=0.34$, while RL-FCM gives $c* = 2$.

*Example 8:* In this example, we apply the U-k-means clustering algorithm to the CIFAR-10 color images [37]. The CIFAR-10 data set consists of 60000 $32 \times 32$ color images in 10 classes, i.e., each pixel is an RGB triplet of unsigned bytes between 0 and 255. There are 50000 training images and 10000 test images. Each red, green, and blue channel value contains 1024 entries. The 10 classes in the data set are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Specifically, we take the first 100 color images (10 images per class) and training 40 multi-way from CIFAR-10 60K images data set for our experiment. The rest 59900 images as the retrieval database. Fig. 6 shows the 100 images sample from the CIFAR-10 images data set. The results for the number of clusters and AR are given in Table 14. From Table 14, it is seen that the proposed U-k-means and k-means with the true $c=10$ give better results on the 100 images sample of the CIFAR-10 data set. The U-k-means has the correct number $c* = 10$ of clusters with 42.5% and $AV\text{-}AR=0.28$ and k-means with $c=10$ gives the same $AV\text{-}AR=0.28$. For the C-FS, the percentage with the correct number $c* = 10$ of clusters is only 16.7% with $AV\text{-}AR=0.24$. X-means underestimates the number of clusters with $c* = 2$. The results from R-EM and RL-FCM on this data sets are missing because the probability of one data point belonged to the $k$th class on these data sets are known as illegitimate proportions at the first iteration.

We further analyze the performance of U-k-means, R-EM, C-FS, and RL-FCM by comparing their average running times of 25 runs for these algorithms, as shown in Table 15. All algorithms are implemented in MATLAB 2017b. From Table 15, it is seen that the proposed U-k-means is the fastest for all data sets among these algorithms, except that the C-FS algorithm is the fastest for the Waveform data set. Furthermore, in Section III, we had mentioned that the proposed U-k-means objective function is simpler than the RL-FCM objective function with saving running time. From Table 15, it is seen that the proposed U-k-means algorithm is actually running faster than the RL-FCM algorithm.

## V. CONCLUSION

In this paper we propose a new schema with a learning framework for the k-means clustering algorithm. We adopt the merit of entropy-type penalty terms to construct a competition schema. The proposed U-k-means algorithm uses the number of points as the initial number of clusters for solving the initialization problem. During iterations, the U-k-means algorithm will discard extra clusters, and then an optimal number of clusters can be automatically found according to the structure of data. The advantages of U-k-means are free of initializations and parameters that also robust to different cluster volumes and shapes with automatically finding the number of clusters. The proposed U-k-means algorithm was performed on several synthetic and real data sets and also compared with most existing algorithms, such as R-EM, C-FS, k-means with the true number $c$, k-means+gap, and X-means algorithms. The results actually demonstrate the superiority of the U-k-means clustering algorithm.

## 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] G. J. McLachlan and K. E. Basford, *Mixture Models: Inference and Applications to Clustering*. New York, NY, USA: Marcel Dekker, 1988.
[4] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm (with discussion)," *J. Roy. Stat. Soc., Ser. B, Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.
[5] J. Yu, C. Chaomurilige, and M.-S. Yang, "On convergence and parameter selection of the EM and DA-EM algorithms for Gaussian mixtures," *Pattern Recognit.*, vol. 77, pp. 188–203, May 2018.

[6] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.
[7] M.-S. Yang, S.-J. Chang-Chien, and Y. Nataliani, "A fully-unsupervised possibilistic C-Means clustering algorithm," *IEEE Access*, vol. 6, pp. 78308–78320, 2018.
[8] 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.
[9] M. Alhawarat and M. Hegazi, "Revisiting K-Means and topic modeling, a comparison study to cluster arabic documents," *IEEE Access*, vol. 6, pp. 42740–42749, 2018.
[10] Y. Meng, J. Liang, F. Cao, and Y. He, "A new distance with derivative information for functional k-means clustering algorithm," *Inf. Sci.*, vols. 463–464, pp. 166–185, Oct. 2018.
[11] Z. Lv, T. Liu, C. Shi, J. A. Benediktsson, and H. Du, "Novel land cover change detection method based on k-Means clustering and adaptive majority voting using bitemporal remote sensing images," *IEEE Access*, vol. 7, pp. 34425–34437, 2019.
[12] 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.
[13] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *J. Intell. Inf. Syst.*, vol. 17, no. 2, pp. 107–145, Dec. 2001.
[14] R. E. Kass and A. E. Raftery, "Bayes factors," *J. Amer. Stat. Assoc.*, vol. 90, pp. 773–795, Jan. 1995.
[15] 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.
[16] 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.
[17] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vols. PAMI–1, no. 2, pp. 224–227, Apr. 1979.
[18] 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.
[19] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Statist. Theory Methods*, vol. 3, no. 1, pp. 1–27, Jan. 1974.
[20] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *J. Roy. Stat. Soc., Ser. B, Stat. Methodol.*, vol. 63, no. 2, pp. 411–423, May 2001.
[21] N. R. Pal and J. Biswas, "Cluster validation using graph theoretic concepts," *Pattern Recognit.*, vol. 30, no. 6, pp. 847–857, Jun. 1997.
[22] N. Ilc, "Modified Dunn's cluster validity index based on graph theory," *Przeglad Elektrotechniczny (Elect. Rev.)*, vol. 88, no. 2, pp. 126–131, 2012.
[23] D. Pelleg and A. 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, pp. 727–734, 2000.
[24] E. Rendón, I. Abundez, A. Arizmendi, and E. M. Quiroz, "Internal versus external cluster validation indexes," *Int. J. Comput. Commun.*, vol. 5, no. 1, pp. 27–34, 2011.
[25] Y. Lei, J. C. Bezdek, S. Romano, N. X. Vinh, J. Chan, and J. Bailey, "Ground truth bias in external cluster validity indices," *Pattern Recognit.*, vol. 65, pp. 58–70, May 2017.
[26] J. Wu, J. Chen, H. Xiong, and M. Xie, "External validation measures for K-means clustering: A data distribution perspective," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6050–6061, Apr. 2009.
[27] L. Jegatha Deborah, R. Baskaran, and A. Kannan, "A survey on internal validity measure for cluster validation," *Int. J. Comput. Sci. Eng. Surv.*, vol. 1, no. 2, pp. 85–102, Nov. 2010.
[28] I. H. Witten, E. Frank, M. A. Hall and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2000.
[29] G. Guo, L. Chen, Y. Ye, and Q. Jiang, "Cluster validation method for determining the number of clusters in categorical sequences," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 12, pp. 2936–2948, Dec. 2017.
[30] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
[31] M.-S. Yang, C.-Y. Lai, and C.-Y. Lin, "A robust EM clustering algorithm for Gaussian mixture models," *Pattern Recognit.*, vol. 45, no. 11, pp. 3950–3961, Nov. 2012.
[32] 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.
[33] M.-S. Yang and Y. Nataliani, "Robust-learning fuzzy c-means clustering algorithm with unknown number of clusters," *Pattern Recognit.*, vol. 71, pp. 45–59, Nov. 2017.
[34] 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: https://archive.ics.uci.edu/ml/datasets.html
[35] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.
[36] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, "Learning a spatially smooth subspace for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–7.
[37] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 001, 2009, vol. 1, no. 4, p. 7.

**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.

**MIIN-SHEN YANG** received the B.S. degree in mathematics from Chung Yuan Christian University, 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, 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, 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 the Chaplain's Office, and is 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, from 2005 to 2011, and is an Associate Editor of *Applied Computational Intelligence and Soft Computing*.