Research Article

Fuzzy c -Means and Cluster Ensemble with Random Projection for Big Data Clustering

Mao Ye,^{1,2} Wenfen Liu,^{1,2} Jianghong Wei,¹ and Xuexian Hu¹

¹State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450002, China

²State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China

Correspondence should be addressed to Mao Ye; yemao119@gmail.com

Received 19 April 2016; Revised 17 June 2016; Accepted 19 June 2016

Academic Editor: Stefan Balint

Copyright © 2016 Mao Ye et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Because of its positive effects on dealing with the curse of dimensionality in big data, random projection for dimensionality reduction has become a popular method recently. In this paper, an academic analysis of influences of random projection on the variability of data set and the dependence of dimensions has been proposed. Together with the theoretical analysis, a new fuzzy c -means (FCM) clustering algorithm with random projection has been presented. Empirical results verify that the new algorithm not only preserves the accuracy of original FCM clustering, but also is more efficient than original clustering and clustering with singular value decomposition. At the same time, a new cluster ensemble approach based on FCM clustering with random projection is also proposed. The new aggregation method can efficiently compute the spectral embedding of data with cluster centers based representation which scales linearly with data size. Experimental results reveal the efficiency, effectiveness, and robustness of our algorithm compared to the state-of-the-art methods.

1. Introduction

With the rapid development of mobile Internet, cloud computing, Internet of things, social network service, and other emerging services, data is growing at an explosive rate recently. How to achieve fast and effective analyses of data and then maximize the data property's benefits has become the focus of attention. The “four Vs” model [1], variety, volume, velocity, and value, for big data has made traditional methods of data analysis unapplicable. Therefore, new techniques for big data analysis such as distributed or parallelized [2, 3], feature extraction [4, 5], and sampling [6] have been widely concerned.

Clustering is an essential method of data analysis through which the original data set can be partitioned into several data subsets according to similarities of data points. It becomes an underlying tool for outlier detection [7], biology [8], indexing [9], and so on. In the context of fuzzy clustering analysis, each object in data set no longer belongs to a single group but possibly belongs to any group. The degree of an object belonging to a group is denoted by a value in $[0, 1]$. Among various methods of fuzzy clustering, fuzzy c -means

(FCM) [10] clustering has received particular attention for its special features. In recent years, based on different sampling and extension methods, a lot of modified FCM algorithms [11–13] designed for big data analysis have been proposed. However, these algorithms are unsatisfactory in efficiency for high dimensional data, since they initially do not take the problem of “curse of dimensionality” into account.

In 1984, Johnson and Lindenstrauss [14] used the projection generated by a random orthogonal matrix to reduce the dimensionality of data. This method can preserve pairwise distances of the points within a factor of $1 \pm \epsilon$. Subsequently, [15] stated that such projection could be produced by a random Gaussian matrix. Moreover, Achlioptas investigated that even projection from a random scaled sign matrix satisfied the property of preserving pairwise distances [16]. These results laid the theoretic foundation for applying random projection to clustering analysis based on pairwise distances. Recently, Boutsidis et al. [17] designed a provably accurate dimensionality reduction method for k -means clustering based on random projection. Since the method above was analyzed for crisp partitions, the effect of random projection on FCM clustering algorithm is still unknown.

As it can combine multiple base clustering solutions of the same object set into a single consensus solution, cluster ensemble has many attractive properties such as improved quality of solution, robust clustering, and knowledge reuse [18]. Ensemble approaches of fuzzy clustering with random projection have been proposed in [19–21]. These methods were all based on multiple random projections of original data set and then integrated all fuzzy clustering results of the projected data sets. Reference [21] pointed out that their method used smaller memory and ran faster than the ones of [19, 20]. However, with respect to crisp partition solution, their method still needs computing and storing the product of membership matrices, which requires time and space complexity with quadratic data size.

Our Contribution. In this paper, our contributions can be divided into two parts: one is the analysis of impact of random projection on FCM clustering; the other is the proposition of a cluster ensemble method with random projection which is more efficient, robust, and suitable for a wider range of geometrical data sets. Concretely, the contributions are as follows:

- (i) We theoretically analyze that random projection can preserve the entire variability of data and prove the effectiveness of random projection for dimensionality reduction from the linear independence of dimensions of projected data. Together with the property of preserving pairwise distances of points, we obtain a modified FCM clustering algorithm with random projection. The accuracy and efficiency of modified algorithm have been verified through experiments on both synthetic and real data sets.
- (ii) We propose a new cluster ensemble algorithm for FCM clustering with random projection which gets spectral embedding efficiently through singular value decomposition (SVD) of the concatenation of membership matrices. The new method avoids the construction of similarity or distance matrix, so it is more efficient and space-saving than method in [21] with respect to crisp partition and methods in [19, 20] for large scale data sets. In addition, the improvements on robustness and efficiency of our approach are also verified by the experimental results on both synthetic and real data sets. At the same time, our algorithm is not only as accurate as the existing ones on Gaussian mixture data set, but also obviously more accurate than the existing ones on the real data set, which indicates that our approach is suitable for a wider range of data sets.

2. Preliminaries

In this section, we present some notations used throughout this paper, introduce the FCM clustering algorithm, and give some traditional cluster ensemble methods using random projection.

2.1. Matrix Notations. We use $\mathbf{X}$ to denote data matrix; $\mathbf{x}_i$ to denote the i th row vector of $\mathbf{X}$ and the i th point; x_{ij} to denote the (i, j) th element of $\mathbf{X}$. $E(\xi)$ means the expectation of a random variable ξ and $\Pr(A)$ denotes the probability of an event A . Let $\text{cov}(\xi, \eta)$ be the covariance of random variables ξ, η ; let $\text{var}(\xi)$ be the variance of random variable ξ .

We denote the trace of matrix by $\text{tr}()$, given $\mathbf{A} \in \mathbb{R}^{n \times n}$; then

$$\text{tr}(\mathbf{A}) = \sum_{i=1}^n a_{ii}. \quad (1)$$

For any matrix $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$, we have the following property:

$$\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA}). \quad (2)$$

Singular value decomposition is a popular dimensionality reduction method, through which one can get a projection: $f: \mathbf{X} \rightarrow \mathbb{R}^t$, with $f(\mathbf{x}_i) = \mathbf{x}_i \mathbf{V}_t$, where $\mathbf{V}_t$ contains the top t right singular vectors of matrix $\mathbf{X}$. The exact SVD of $\mathbf{X}$ takes cubic time of dimension size and quadratic time of data size.

2.2. Fuzzy c -Means Clustering Algorithm (FCM). The goal of fuzzy clustering is to get a flexible partition, where each point has membership in more than one cluster with values in $[0, 1]$. Among the various fuzzy clustering algorithms, FCM clustering algorithm is widely used in low dimensional data because of its efficiency and effectiveness [22]. We start from giving the definition of fuzzy c -means clustering problem and then describe the FCM clustering algorithm precisely.

Definition 1 (the fuzzy c -means clustering problem). Given a data set of n points with d features denoted by an $n \times d$ matrix $\mathbf{X}$, a positive integer c regarded as the number of clusters, and fuzzy constant $m > 1$, find the partition matrix $\mathbf{U}_{\text{opt}} \in \mathbb{R}^{c \times n}$ and centers of clusters $\mathbf{V}_{\text{opt}} = \{\mathbf{v}_{\text{opt},1}, \mathbf{v}_{\text{opt},2}, \dots, \mathbf{v}_{\text{opt},c}\}$, such that

$$(\mathbf{U}, \mathbf{V})_{\text{opt}} = \arg \min_{\mathbf{U}, \mathbf{V}} \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m \|\mathbf{x}_j - \mathbf{v}_i\|^2. \quad (3)$$

Here, $\|\cdot\|$ denotes norm, usually Euclidean norm; the element of partition matrix u_{ij} denotes the membership of point j in the cluster i . Moreover, for any $j \in [1, n]$, $\sum_{i=1}^c u_{ij} = 1$. The objective function is defined as $\sum_{i=1}^c \sum_{j=1}^n u_{ij}^m \|\mathbf{x}_j - \mathbf{v}_i\|^2 \triangleq \text{obj}$.

FCM clustering algorithm first computes the degree of membership through distances between points and centers of clusters and then updates the center of each cluster based on the membership degree. By means of computing cluster centers and partition matrix iteratively, a solution is obtained. It should be noted that FCM clustering can only get a locally optimal solution and the final clustering result depends on the initialization. The detailed procedure of FCM clustering is shown in Algorithm 1.

2.3. Ensemble Aggregations for Multiple Fuzzy Clustering Solutions with Random Projection. There are several algorithms

Input: data set $\mathbf{X}$ (an $n \times d$ matrix), number of clusters c , fuzzy constant m ;

Output: partition matrix $\mathbf{U}$, centers of clusters $\mathbf{V}$;

Initialize: sample $\mathbf{U}$ (or $\mathbf{V}$) randomly from proper space;

While $|\text{obj}_{\text{old}} - \text{obj}_{\text{new}}|^2 > \varepsilon$, **do**

$$u_{ij} = \left[\sum_{k=1}^c \left(\frac{\|\mathbf{x}_j - \mathbf{v}_i\|}{\|\mathbf{x}_j - \mathbf{v}_k\|} \right)^{2/(m-1)} \right]^{-1}, \quad \forall i, j$$

$$\mathbf{v}_i = \frac{\sum_{j=1}^n (u_{ij})^m \mathbf{x}_j}{\sum_{j=1}^n (u_{ij})^m}, \quad \forall i$$

$$\text{obj} = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m \|\mathbf{x}_j - \mathbf{v}_i\|^2.$$

ALGORITHM 1: FCM clustering algorithm.

proposed for aggregating the multiple fuzzy clustering results with random projection. The main strategy is to generate data membership matrices through multiple fuzzy clustering solutions on the different projected data sets and then to aggregate the resulting membership matrices. Therefore, different methods of generation and aggregation of membership matrices lead to various ensemble approaches about fuzzy clustering.

The first cluster ensemble approach using random projection was proposed in [20]. After projecting the data into low dimensional space with random projection, the membership matrices were calculated through the probabilistic model θ of c Gaussian mixture gained by EM clustering. Subsequently, the similarity of points i and j was computed as $P_{ij}^\theta = \sum_{l=1}^c P(l | i, \theta) \times P(l | j, \theta)$, where $P(l | i, \theta)$ denoted the probability of point i belonging to cluster l under model θ and P_{ij}^θ denoted the probability that points i and j belonged to the same cluster under model θ . The aggregated similarity matrix was obtained by averaging across the multiple runs, and the final clustering solution was produced by a hierarchical clustering method called complete linkage. For mixture model, the estimation for the cluster number and values of unknown parameters is often complicated [23]. In addition, this approach needs $O(n^2)$ space for storing the similarity matrix of data points.

Another approach which was used to find genes in DNA microarray data was presented in [19]. Similarly, the data was projected into a low dimensional space with random matrix. Then the method employed FCM clustering to partition the projected data and generated membership matrices $\mathbf{U}_i \in \mathbb{R}^{c \times n}$, $i = 1, 2, \dots, r$ with multiple runs r . For each run i , the similarity matrix was computed as $\mathbf{M}_i = \mathbf{U}_i^T \mathbf{U}_i$. Then the combined similarity matrix $\mathbf{M}$ was calculated by averaging as $\mathbf{M} = (1/r) \sum_{i=1}^r \mathbf{M}_i$. A distance matrix was computed by $\mathbf{D} = \mathbf{1} - \mathbf{M}$ and the final partition matrix was gained by FCM clustering on the distance matrix $\mathbf{D}$. Since this method needs to compute the product of partition matrix and its transpose, the time complexity is $O(r * cn^2)$ and the space complexity is $O(n^2)$.

Considering the large scale data set in the context of big data, [21] proposed a new method for aggregating partition

matrices from FCM clustering. They concatenated the partition matrices as $\mathbf{U}_{\text{con}} = [\mathbf{U}_1^T, \mathbf{U}_2^T, \dots]$, instead of averaging the agreement matrix. Finally, they got the ensemble result as $\mathbf{U}_f = \text{FCM}(\mathbf{U}_{\text{con}}, c)$. This algorithm avoids the products of partition matrices and is more suitable than [19] for large scale data sets. However, it still needs the multiplication of concatenated partition matrix when crisp partition result is wanted.

3. Random Projection

Dimensionality reduction is a common technique for analysis of high dimensional data. The most popular skill is SVD (or principal component analysis) where the original features are replaced by a small size of principal components in order to compress the data. But SVD takes cubic time of the number of dimensions. Recently, some literatures stated that random projection can be applied to dimensionality reduction and preserve pairwise distances within a small factor [15, 16]. Low computing complexity and preserving the metric structure make random projection receive much attention. Lemma 2 indicates that there are three kinds of simple random projection possessing the above properties.

Lemma 2 (see [15, 16]). *Let matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ be a data set of n points and d features. Given $\varepsilon, \beta > 0$, let*

$$k_0 = \frac{4 + 2\beta}{\varepsilon^2/2 - \varepsilon^3/3} \log n. \quad (4)$$

For integer $t \geq k_0$, let matrix $\mathbf{R}$ be a $d \times t$ ($t \leq d$) random matrix, wherein elements R_{ij} are independently identically distributed random variables from either one of the following three probability distributions:

$$R_{ij} \sim N(0, 1),$$

$$R_{ij} = \begin{cases} +1 & \text{with probability } \frac{1}{2}, \\ -1 & \text{with probability } \frac{1}{2}, \end{cases} \quad (5)$$

$$R_{ij} = \sqrt{3} \times \begin{cases} +1 & \text{with probability } \frac{1}{6}, \\ 0 & \text{with probability } \frac{2}{3}, \\ -1 & \text{with probability } \frac{1}{6}. \end{cases} \quad (6)$$

Let $f : \mathbb{R}^d \rightarrow \mathbb{R}^t$ with $f(\mathbf{x}_i) = (1/\sqrt{t})\mathbf{x}_i\mathbf{R}$. For any $\mathbf{u}, \mathbf{v} \in \mathbf{X}$, with probability at least $1 - n^{-\beta}$, it holds that

$$(1 - \varepsilon) \|\mathbf{u} - \mathbf{v}\|_2^2 \leq \|f(\mathbf{u}) - f(\mathbf{v})\|_2^2 \leq (1 + \varepsilon) \|\mathbf{u} - \mathbf{v}\|_2^2. \quad (7)$$

Lemma 2 implies that if the number of dimensions of data reduced by random projection is bigger than a certain bound, then pairwise Euclidean distance squares are preserved within a multiplicative factor of $1 \pm \varepsilon$.

With the above properties, researchers have checked the feasibility of applying random projection to k -means clustering in terms of theory and experiment [17, 24]. However, as membership degrees for FCM clustering and k -means clustering are defined differently, the analysis method can not be directly used for assessing the effect of random projection on FCM clustering. Motivated by the idea of principal component analysis, we draw the conclusion that the compressed data gains the whole variability of original data in probabilistic sense based on the analysis of the variance difference. Besides, variables referring to dimensions of projected data are linear independent. As a result, we can achieve dimensionality reduction via replacing original data by compressed data as “*principal components*.”

Next, we give a useful lemma for proof of the subsequent theorem.

Lemma 3. Let ξ_i ($1 \leq i \leq n$) be independently distributed random variables from one of the three probability distributions described in Lemma 2; then

$$\Pr \left\{ \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \xi_i^2 = 1 \right\} = 1. \quad (8)$$

Proof. According to the probability distribution of random variable ξ_i , it is easy to know that

$$\begin{aligned} E(\xi_i^2) &= 1 \quad (1 \leq i \leq n), \\ E\left(\frac{1}{n} \sum_{i=1}^n \xi_i^2\right) &= 1. \end{aligned} \quad (9)$$

Then, $\{\xi_i^2\}$ obeys the law of large numbers; namely,

$$\begin{aligned} \Pr \left\{ \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \xi_i^2 = E\left(\frac{1}{n} \sum_{i=1}^n \xi_i^2\right) \right\} \\ = \Pr \left\{ \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \xi_i^2 = 1 \right\} = 1. \end{aligned} \quad (10)$$

□

Since centralization of data does not change the distance of any two points and the FCM clustering algorithm is

based on pairwise distances to partition data points, we assume that expectation of the data input is 0. In practice, covariance matrix of population is likely unknown. Therefore, we investigate the effect of random projection on variability of both population and sample.

Theorem 4. Let data set $\mathbf{X} \in \mathbb{R}^{n \times d}$ be n independent samples of d -dimensional random vector $(X_1, X_2, \dots, X_d)$, and $\mathbf{S}$ denotes the sample covariance matrix of $\mathbf{X}$. The random projection induced by random matrix $\mathbf{R} \in \mathbb{R}^{d \times t}$ maps the d -dimensional random vector to t -dimensional random vector $(Y_1, Y_2, \dots, Y_t) = (1/\sqrt{t})(X_1, X_2, \dots, X_d) \cdot \mathbf{R}$, and $\mathbf{S}^*$ denotes the sample covariance matrix of projected data. If elements of random matrix $\mathbf{R}$ obey distribution demanded by Lemma 2 and are mutually independent with random vector $(X_1, X_2, \dots, X_d)$, then

- (1) dimensions of projected data are linearly independent: $\text{cov}(Y_i, Y_j) = 0, \forall i \neq j$;
- (2) random projection maintains the whole variability: $\sum_{i=1}^t \text{var}(Y_i) = \sum_{i=1}^d \text{var}(X_i)$; when $t \rightarrow \infty$, with probability 1, $\text{tr}(\mathbf{S}^*) = \text{tr}(\mathbf{S})$.

Proof. It is easy to know that the expectation of any element of random matrix $E(R_{ij}) = 0, 1 \leq i \leq d, 1 \leq j \leq t$. As elements of random matrix $\mathbf{R}$ and random vector $(X_1, X_2, \dots, X_d)$ are mutually independent, the covariance of random vector induced by random projection is

$$\begin{aligned} \text{cov}(Y_i, Y_j) &= \text{cov}\left(\frac{1}{\sqrt{t}} \cdot \sum_{k=1}^d X_k \cdot R_{ki}, \frac{1}{\sqrt{t}} \cdot \sum_{l=1}^d X_l \cdot R_{lj}\right) \\ &= \frac{1}{t} \cdot \sum_{k=1}^d \sum_{l=1}^d \text{cov}(X_k \cdot R_{ki}, X_l \cdot R_{lj}) \\ &= \frac{1}{t} \cdot \sum_{k=1}^d \sum_{l=1}^d E(X_k \cdot R_{ki} \cdot X_l \cdot R_{lj}) - \frac{1}{t} \\ &\quad \cdot \sum_{k=1}^d \sum_{l=1}^d (E(X_k \cdot R_{ki}) \cdot E(X_l \cdot R_{lj})) \quad (11) \\ &= \frac{1}{t} \cdot \sum_{k=1}^d \sum_{l=1}^d E(X_k \cdot R_{ki} \cdot X_l \cdot R_{lj}) \\ &= \frac{1}{t} \cdot \sum_{k=1}^d \sum_{l=1}^d (E(X_k \cdot X_l) \cdot E(R_{ki} \cdot R_{lj})) \\ &= \frac{1}{t} \cdot \sum_{k=1}^d (E(X_k^2) \cdot E(R_{ki} \cdot R_{kj})). \end{aligned}$$

- (1) If $i \neq j$, then

$$\begin{aligned} \text{cov}(Y_i, Y_j) &= \frac{1}{t} \cdot \left(\sum_{k=1}^d E(X_k^2) \cdot E(R_{ki}) \cdot E(R_{lj}) \right) \\ &= 0. \end{aligned} \quad (12)$$

(2) If $i = j$, then

$$\begin{aligned} \text{cov}(Y_i, Y_i) &= \text{var}(Y_i) = \frac{1}{t} \cdot \left(\sum_{k=1}^d E(X_k^2) \cdot E(R_{ki}^2) \right) \\ &= \frac{1}{t} \cdot \sum_{k=1}^d E(X_k^2). \end{aligned} \quad (13)$$

Thus, by assumption $E(X_i) = 0$ ($1 \leq i \leq d$), we can get

$$\sum_{i=1}^t \text{var}(Y_i) = \sum_{i=1}^d \text{var}(X_i). \quad (14)$$

We denote spectral decomposition of sample covariance matrix $\mathbf{S}$ by $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{V}$ is the matrix of eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix in which the diagonal elements are $\lambda_1, \lambda_2, \dots, \lambda_d$ and $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$. Supposing the data samples have been centralized, namely, their means are 0s, we can get covariance matrix $\mathbf{S} = (1/n)\mathbf{X}^T\mathbf{X}$. For convenience, we still denote a sample of random matrix by $\mathbf{R}$. Thus, projected data $\mathbf{Y} = (1/\sqrt{t})\mathbf{X}\mathbf{R}$ and sample covariance matrix of projected data $\mathbf{S}^* = (1/n)((1/\sqrt{t})\mathbf{X}\mathbf{R})^T((1/\sqrt{t})\mathbf{X}\mathbf{R}) = (1/t)\mathbf{R}^T\mathbf{S}\mathbf{R}$. Then, we can get

$$\begin{aligned} \text{tr}(\mathbf{S}^*) &= \text{tr}\left(\frac{1}{t}\mathbf{R}^T\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T\mathbf{R}\right) = \text{tr}\left(\frac{1}{t}\mathbf{R}^T\mathbf{\Lambda}\mathbf{V}\mathbf{V}^T\mathbf{R}\right) \\ &= \text{tr}\left(\frac{1}{t}\mathbf{R}^T\mathbf{\Lambda}\mathbf{R}\right) = \sum_{i=1}^d \lambda_i \cdot \left(\frac{1}{t} \cdot \sum_{j=1}^t r_{ij}^2 \right), \end{aligned} \quad (15)$$

where r_{ij} ($1 \leq i \leq d$, $1 \leq j \leq t$) is sample of element of random matrix $\mathbf{R}$.

In practice, the spectrum of a covariance often displays a distinct decay after few large eigenvalues. So we assume that there exists an integer p , limited constant $q > 0$, such that, for all $i > p$, it holds that $\lambda_i \leq q$. Then,

$$\begin{aligned} |\text{tr}(\mathbf{S}^*) - \text{tr}(\mathbf{S})| &= \left| \sum_{i=1}^d \lambda_i \cdot \left(\frac{1}{t} \sum_{j=1}^t r_{ij}^2 - 1 \right) \right| \\ &\leq \left| \sum_{i=1}^p \lambda_i \cdot \left(\frac{1}{t} \sum_{j=1}^t r_{ij}^2 - 1 \right) \right| \\ &\quad + \left| \sum_{i=p+1}^d \lambda_i \cdot \left(\frac{1}{t} \sum_{j=1}^t r_{ij}^2 - 1 \right) \right| \\ &\leq \left| \sum_{i=1}^p \lambda_i \cdot \frac{1}{t} \sum_{j=1}^t (r_{ij}^2 - 1) \right| \\ &\quad + q \left| \sum_{i=p+1}^d \left(\frac{1}{t} \sum_{j=1}^t (r_{ij}^2 - 1) \right) \right|. \end{aligned} \quad (16)$$

By Lemma 3, with probability 1,

$$\begin{aligned} \lim_{t \rightarrow \infty} \left(\frac{1}{t} \sum_{j=1}^t (r_{ij}^2 - 1) \right) &= 0, \\ \lim_{t \rightarrow \infty} \sum_{i=p+1}^d \left(\frac{1}{t} \sum_{j=1}^t (r_{ij}^2 - 1) \right) &= 0. \end{aligned} \quad (17)$$

Combining the above arguments, we achieve $\text{tr}(\mathbf{S}^*) = \text{tr}(\mathbf{S})$ with probability 1, when $t \rightarrow \infty$. $\square$

Part (1) of Theorem 4 indicates that compressed data produced by random projection can take much information with low dimensionality owing to linear independence of reduced dimensions. Part (2) manifests that sum of variances of dimensions of original data is consistent with the one of projected data, namely, random projection holds the variability of primal data. Combining results of Lemma 2 with those of Theorem 4, we consider that random projection can be employed to improve the efficiency of FCM clustering algorithm with low dimensionality, and the modified algorithm can keep the accuracy of partition approximately.

4. FCM Clustering with Random Projection and an Efficient Cluster Ensemble Approach

4.1. FCM Clustering via Random Projection. According to the results of Section 3, we design an improved FCM clustering algorithm with random projection for dimensionality reduction. The procedure of new algorithm is shown in Algorithm 2.

Algorithm 2 reduces the dimensions of input data via multiplying a random matrix. Compared with the $O(cnd^2)$ time for running each iteration in original FCM clustering, the new algorithm would imply an $O(cn(\varepsilon^{-2} \ln n)^2)$ time for each iteration. Thus, the time complexity of new algorithm decreases obviously for high dimensional data in the case $\varepsilon^{-2} \ln n \ll d$. Another common dimensionality reduction method is SVD. Compared with the $O(d^3 + nd^2)$ time of running SVD on data matrix $\mathbf{X}$, the new algorithm only needs $O(\varepsilon^{-2} d \ln n)$ time to generate random matrix $\mathbf{R}$. It indicates that random projection is a cost-effective method of dimensionality reduction for FCM clustering algorithm.

4.2. Ensemble Approach Based on Graph Partition. As different random projections may result in different clustering solutions [20], it is attractive to design the cluster ensemble framework with random projection for improved and robust clustering performance. Although it uses smaller memory and runs faster than ensemble method in [19], the cluster ensemble algorithm in [21] still needs product of concatenated partition matrix for crisp grouping, which leads to a high time and space costs under the circumstances of big data.

In this section, we propose a more efficient and effective aggregation method for multiple FCM clustering results. The overview of our new ensemble approach is presented in Figure 1. The new ensemble method is based on partition

Input: data set $\mathbf{X}$ (an $n \times d$ matrix), number of clusters c , fuzzy constant m , FCM clustering algorithm;
Output: partition matrix $\mathbf{U}$, centers of clusters $\mathbf{V}$.
(1) sample a $d \times t$ ($t \leq d$, $t = \Omega(\epsilon^{-2} \ln n)$) random projection $\mathbf{R}$ meeting the requirements of Lemma 2;
(2) compute the product $\mathbf{Y} = (1/\sqrt{t})\mathbf{X}\mathbf{R}$;
(3) run FCM algorithm on $\mathbf{Y}$, get the partition matrix $\mathbf{U}$;
(4) compute the centers of clusters through original data $\mathbf{X}$ and $\mathbf{U}$.

ALGORITHM 2: FCM clustering with random projection.

Input: data set $\mathbf{X}$ (an $n \times d$ matrix), number of clusters c , reduced dimension t , number of random projection r , FCM clustering algorithm;
Output: cluster label vector $\mathbf{u}$.
(1) at each iteration $i \in [1, r]$, run Algorithm 2, get membership matrix $\mathbf{U}_i \in \mathbb{R}^{c \times n}$;
(2) concatenate the membership matrices $\mathbf{U}_{\text{con}} = [\mathbf{U}_1^T, \dots, \mathbf{U}_r^T] \in \mathbb{R}^{n \times cr}$;
(3) compute the first c left singular vectors of $\widehat{\mathbf{U}}_{\text{con}}$, denoted by $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_c] \in \mathbb{R}^{n \times c}$, where $\widehat{\mathbf{U}}_{\text{con}} = \mathbf{U}_{\text{con}}(\mathbf{r} \cdot \widehat{\mathbf{D}})^{-1/2}$, $\widehat{\mathbf{D}}$ is a diagonal matrix and $\widehat{d}_{ii} = \sum_j u_{\text{con},ji}$;
(4) treat each row of $\mathbf{A}$ as a data point and apply k -means to obtain cluster label vector.

ALGORITHM 3: Cluster ensemble for FCM clustering with random projection.

on similarity graph. For each random projection, a new data set is generated. After performing FCM clustering on the new data sets, membership matrices are output. The elements of membership matrix are treated as the similarity measure between points and the cluster centers. Through SVD on the concatenation of membership matrices, we get the spectral embedding of data point efficiently. The detailed procedure of new cluster ensemble approach is shown in Algorithm 3.

In step (3) of the procedure in Algorithm 3, the left singular vectors of $\widehat{\mathbf{U}}_{\text{con}}$ are equivalent to the eigenvectors of $\widehat{\mathbf{U}}_{\text{con}} \widehat{\mathbf{U}}_{\text{con}}^T$. It implies that we regard the matrix product as a construction of affinity matrix of data points. This method is motivated by the research on landmark-based representation [25, 26]. In our approach, we treat the cluster centers of each FCM clustering run as landmarks and the membership matrix as landmark-based representation. Thus, the concatenation of membership matrices forms a combinational landmark-based representation matrix. In this way, the graph similarity matrix is computed as

$$\mathbf{W} = \widehat{\mathbf{U}}_{\text{con}} \widehat{\mathbf{U}}_{\text{con}}^T, \quad (18)$$

which can create spectral embedding efficiently through step (3). To normalize the graph similarity matrix, we multiply $\mathbf{U}_{\text{con}}$ by $(\mathbf{r} \cdot \widehat{\mathbf{D}})^{-1/2}$. As a result, the degree matrix of $\mathbf{W}$ is an identity matrix.

There are two perspectives to explain why our approach works. Considering the similarity measure defined by u_{ij} in FCM clustering, proposition 3 in [26] demonstrated that singular vectors of $\mathbf{U}_i$ converged to eigenvectors of $\mathbf{W}_s$ as c converges to n , where $\mathbf{W}_s$ was affinity matrix generated in standard spectral clustering. As a result, singular vectors of $\widehat{\mathbf{U}}_{\text{con}}$ converge to eigenvectors of normalized affinity matrix $\widehat{\mathbf{W}}_s$. Thus, our final output will converge to the one of standard spectral clustering as c converges to n . Another explanation is about the similarity measure defined by $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$,

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are data points. We can treat each row of $\widehat{\mathbf{U}}_{\text{con}}$ as a transformational data point. As a result, affinity matrix obtained here is the same as the one of standard spectral embedding, and our output is just the partition result of standard spectral clustering.

To facilitate comparison of different ensemble methods for FCM clustering solutions with random projection, we denote the approach of [19] by EFCM-A (average the products of membership matrices), the algorithm of [21] by EFCM-C (concatenate the membership matrices), and our new method by EFCM-S (spectral clustering on the membership matrices). In the cluster ensemble phase, the main computations of EFCM-A method are multiplications of membership matrices. Similarly, the algorithm of EFCM-C also needs the product of concatenated membership matrices in order to get the crisp partition result. Thus the above methods both need $O(n^2)$ space and $O(crn^2)$ time. However, the main computation of EFCM-S is SVD for $\widehat{\mathbf{U}}_{\text{con}}$ and k -means clustering for $\mathbf{A}$. The overall space is $O(crn)$, the SVD time is $O((cr)^2n)$, and the k -means clustering time is l^2n , where l is iteration number of k -means. Therefore, computational complexity of EFCM-S is obviously decreased compared with the ones of EFCM-A and EFCM-C considering $cr \ll n$ and $l \ll n$ in large scale data set.

5. Experiments

In this section, we present the experimental evaluations of new algorithms proposed in Section 4. We implemented the related algorithms in Matlab computing environment and conducted our experiments on a Windows-based system with the Intel Core 3.6 GHz processor and 16 GB of RAM.

5.1. Data Sets and Parameter Settings. We conducted the experiments on synthetic and real data sets which both have relatively high dimensionality. The synthetic data

FIGURE 1: Framework of the new ensemble approach based on graph partition.

set had 10000 data points with 1000 dimensions which were generated from 3 Gaussian mixtures in proportions (0.25, 0.5, 0.25). The means of components were $(2, 2, \dots, 2)_{1000}$, $(0, 0, \dots, 0)_{1000}$, and $(-2, -2, \dots, -2)_{1000}$ and the standard deviations were $(1, 1, \dots, 1)_{1000}$, $(2, 2, \dots, 2)_{1000}$, and $(3, 3, \dots, 3)_{1000}$. The real data set is the daily and sports activities data (ACT) published on UCI machine learning repository (the ACT data set can be found at <http://archive.ics.uci.edu/ml/datasets/Daily+and+Sports+Activities>). These are data of 19 activities collected by 45 motion sensors in 5 minutes at 25 Hz sampling frequency. Each activity was performed by 8 subjects in their own styles. To get high dimensional data sets, we treated 1 minute and 5 seconds of activity data as an instance, respectively. As a result, we got 760×67500 (ACT1) and 9120×5625 (ACT2) data matrices whose rows were activity instances and columns were features.

For the parameters of FCM clustering, we let $\varepsilon = 10^{-5}$, we let maximum iteration number be 100, we let fuzzy factor m be 2, and we let the number of clusters be $c = 3$ for synthetic data set and $c = 19$ for ACT data sets. We also normalized the objective function as $\text{obj}^* = \text{obj} / \|\mathbf{X}\|_F^2$, where $\|\cdot\|_F$ is Frobenius norm of matrix [27]. To minimize the influence introduced by different initializations, we present the average values of evaluation indices of 20 independent experiments.

In order to compare different dimensionality reduction methods for FCM clustering, we initialized algorithms by

choosing c points randomly as the cluster centers and made sure that every algorithm began with the same initialization. In addition, we ran Algorithm 2 with $t = 10, 20, \dots, 100$ for synthetic data set and $t = 100, 200, \dots, 1000$ for ACT1 data set. Two kinds of random projections (with random variables from (5) in Lemma 2) were both tested for verifying their feasibility. We also compared Algorithm 2 against another popular method of dimensionality reduction—SVD. What calls for special attention is that the number of eigenvectors corresponding to nonzero eigenvalues of ACT1 data is only 760, so we just took $t = 100, 200, \dots, 700$ on FCM clustering with SVD for ACT1 data set.

Among comparisons of different cluster ensemble algorithms, we set dimension number of projected data as $t = 10, 20, \dots, 100$ for both synthetic and ACT2 data sets. In order to meet $cr \ll n$ for Algorithm 3, the number of random projection r was set as 20 for the synthetic data set and 5 for the ACT2 data set, respectively.

5.2. Evaluation Criteria. For clustering algorithms, clustering validation and running time are two important indices for judging their performances. Clustering validation measures evaluate the goodness of clustering results [28] and often can be divided into two categories: external clustering validation and internal clustering validation. External validation measures use external information such as the given class labels to evaluate the goodness of solution output by a clustering

algorithm. On the contrary, internal measures are to evaluate the clustering results using feature inherited from data sets. In this paper, validity evaluation criteria used are rand index and clustering validation index based on nearest neighbors for crisp partition, together with fuzzy rand index and Xie-Beni index for fuzzy partition. Here, rand index and fuzzy rand index are external validation measures, whereas the clustering validation index based on nearest neighbors index and Xie-Beni index are internal validation measures.

(1) *Rand Index (RI)* [29]. RI describes the similarity of clustering solution and correct labels through pairs of points. It takes into account the numbers of point pairs that are in the same and different clusters. The RI is defined as

$$RI = \frac{n_{11} + n_{00}}{C_n^2}, \quad (19)$$

where n_{11} is the number of pairs of points that exist in the same cluster in both clustering result and given class labels, n_{00} is the number of pairs of points that are in different subsets for both clustering result and given class labels, and C_n^2 equals $n(n-1)/2$. The value of RI ranges from 0 to 1, and the higher value implies the better clustering solution.

(2) *Fuzzy Rand Index (FRI)* [30]. FRI is a generalization of RI with respect to soft partition. It also measures the proportion of pairs of points which exist in the same and different clusters in both clustering solution and true class labels. It needs to compute the analogous n_{11} and n_{00} through contingency table, described in [30]. Therefore, the range of FRI is also [0, 1] and the larger value means more accurate cluster solution.

(3) *Xie-Beni Index (XB)* [31]. XB takes the minimum square distance between cluster centers as the separation of the partition and the average square fuzzy deviation of data points as the compactness of the partition. XB is calculated as follows:

$$XB = \frac{\sum_{i=1}^c \sum_{j=1}^n u_{ij}^m \|\mathbf{x}_j - \mathbf{v}_i\|^2}{n \cdot \min_{i,j} \|\mathbf{v}_i - \mathbf{v}_j\|^2}, \quad (20)$$

where $\sum_{i=1}^c \sum_{j=1}^n u_{ij}^m \|\mathbf{x}_j - \mathbf{v}_i\|^2$ is just the objective function of FCM clustering and $\mathbf{v}_i$ is the center of cluster i . The smallest XB indicates the optimal cluster partition.

(4) *Clustering Validation Index Based on Nearest Neighbors (CVNN)* [32]. The separation of CVNN is about the situation of objects that have geometrical information of each cluster, and the compactness is the mean pairwise distance between objects in the same cluster. CVNN is computed as follows:

$$CVNN(c, k) = \frac{\text{Sep}(c, k)}{(\max_{c_{\min} \leq c \leq c_{\max}} \text{Sep}(c, k))} + \frac{\text{Com}(c)}{(\max_{c_{\min} \leq c \leq c_{\max}} \text{Com}(c))}, \quad (21)$$

where $\text{Sep}(c, k) = \max_{i=1,2,\dots,c} ((1/n_i) \cdot \sum_{j=1}^{n_i} (q_j/k))$ and $\text{Com}(c) = \sum_{i=1}^c ((2/n_i(n_i-1)) \cdot \sum_{x,y \in \text{Clu}_i} d(x, y))$. Here, c is the

number of clusters in partition result, $c_{\max}$ is the maximum cluster number given, $c_{\min}$ is the minimum cluster number given, k is the number of nearest neighbors, n_i is the number of objects in the i th cluster Clu_i , q_j denotes the number of nearest neighbors of Clu_i 's j th object which are not in Clu_i , and $d(x, y)$ denotes the distance between x and y . The lower CVNN value indicates the better clustering solution.

Objective function is a special evaluation criterion of validity for FCM clustering algorithm. The smaller objective function indicates that the points inside clusters are more "similar."

Running time is also an important evaluation criterion often related to the scalability of algorithm. One main target of random projection for dimensionality reduction is to decrease the runtime and enhance the applicability of algorithm in the context of big data.

5.3. Performance of FCM Clustering with Random Projection. The experimental results about FCM clustering with random projection are presented in Figure 2 where (a), (c), (e), and (g) correspond to the synthetic data set and (b), (d), (f), and (h) correspond to the ACT1 data set. The evaluation criteria used to assess proposed algorithms are FRI, (a) and (b), XB, (c) and (d), objective function, (e) and (f), and running time, (g) and (h). "SignRP" denotes the proposed algorithm with random sign matrix, "GaussRP" denotes the FCM clustering with random Gaussian matrix, "FCM" denotes the original FCM clustering algorithm, and "SVD" denotes the FCM clustering with dimensionality reduction through SVD. It should be noted that true XB value of FCM clustering in subfigure (d) is $4.03e+12$, not 0.

From Figure 2, we can see that FCM clustering with random projection is clearly more efficient than the original FCM clustering. When number of dimensions t is above certain bound, the validity indices are nearly stable and similar to the ones of naive FCM clustering for both data sets. This verifies the conclusion that "accuracy of clustering algorithm can be preserved when the dimensionality exceeds a certain bound." The effectiveness for random projection method is also verified by the small bound compared to the total dimensions (30/1000 for synthetic data and 300/67500 for ACT1 data). Besides, the two different kinds of random projection methods have the similar impact on FCM clustering because of the analogous plot.

The higher objective function values and the smaller XB indices of SVD method for synthetic data set indicate that the generated clustering solution has better separation degree between clusters. The external cluster validation indices also verify that SVD method has better clustering results for synthetic data. These observations state that SVD method is more suitable for Gaussian mixture data sets than FCM clustering with random projection and naive FCM clustering.

Although the SVD method has a higher FRI for synthetic data set, the random projection methods have analogous FRI values for ACT1 data set and better objective function values for both data sets. In addition, the random projection approaches are obviously more efficient as the SVD needs cubic time of dimensionality. Hence, these observations indicate that our algorithm is quite encouraging in practice.

(a) FRI versus number of dimensions

(b) FRI versus number of dimensions

(c) FRI versus number of dimensions

(d) FRI versus number of dimensions

(e) FRI versus number of dimensions

(f) FRI versus number of dimensions

FIGURE 2: Continued.

FIGURE 2: Performance of clustering algorithms with different dimensionality.

TABLE 1: CVNN indices for different ensemble approaches on ACT2 data.

Dimension t	10	20	30	40	50	60	70	80	90	100
EFCM-A	1.7315	1.7383	1.7449	1.7789	1.819	1.83	1.7623	1.8182	1.8685	1.8067
EFCM-C	1.7938	1.7558	1.7584	1.8351	1.8088	1.8353	1.8247	1.8385	1.8105	1.8381
EFCM-S	1.3975	1.3144	1.2736	1.2974	1.3112	1.3643	1.3533	1.409	1.3701	1.3765

5.4. Comparisons of Different Cluster Ensemble Methods. The comparisons of different cluster ensemble approaches are shown in Figure 3 and Table 1. Similarly, (a) and (c) of the figure correspond to the synthetic data set and (b) and (d) corresponds to the ACT2 data set. We use RI, (a) and (b), and running time, (c) and (d), to present the performance of ensemble methods. Meanwhile, the meanings of EFCM-A, EFCM-C, and EFCM-S are identical to the ones in Section 4.2. In order to get crisp partition for EFCM-A and EFCM-C, we used hierarchical clustering-complete linkage method after getting the distance matrix as in [21]. Since all three cluster ensemble methods get perfect partition results on synthetic data set, we only compare CVNN indices of different ensemble methods on ACT2 data set, which is presented in Table 1.

In Figure 3, running time of our algorithm is shorter for both data sets. This verifies the result of time complexity analysis for different algorithms in Section 4.2. The three cluster ensemble methods all get the perfect partition for synthetic data set, whereas our method is more accurate than the other two methods for ACT2 data set. The perfect partition results suggest that all three ensemble methods are suitable for Gaussian mixture data set. However, the almost 18% improvement on RI for ACT2 data set should be due

to the different grouping ideas. Our method is based on the graph partition such that the edges between different clusters have low weight and the edges within a cluster have high weight. This clustering way of spectral embedding is more suitable for ACT2 data set. In Table 1, the smaller values of CVNN of our new method also show that new approach has better partition results on ACT2 data set. These observations indicate that our algorithm has the advantage on efficiency and adapts to a wider range of geometries.

We also compare the stability for three ensemble methods, presented in Table 2. From the table, we can see that the standard deviation of RI about EFCM-S is a lower order of magnitude than the ones of the other methods. Hence, this result shows that our algorithm is more robust.

Aiming at the situation of unknown clusters' number, we also varied the number of clusters c in FCM clustering and spectral embedding for our new method. We denote this version of new method as EFCM-SV. Since the number of random projections was set as 5 for ACT2 data set, we changed the clusters' number from 17 to 21 as the input of FCM clustering algorithm. In addition, we set the clusters' number from 14 to 24 as the input of spectral embedding and applied CVNN to estimate the most plausible number of clusters. The experimental results are presented in Table 3.

TABLE 2: Standard deviations of RI of 20 runs with different dimensions on ACT2 data.

Dimension t	10	20	30	40	50	60	70	80	90	100
EFCM-A	0.0222	0.0174	0.018	0.0257	0.0171	0.0251	0.0188	0.0172	0.0218	0.0184
EFCM-C	0.0217	0.0189	0.0128	0.0232	0.0192	0.0200	0.0175	0.0194	0.0151	0.0214
EFCM-S	0.0044	0.0018	0.0029	0.0030	0.0028	0.0024	0.0026	0.0020	0.0024	0.0019

TABLE 3: RI values for EFCM-S and EFCM-Sv on ACT2 data.

Dimension t	10	20	30	40	50	60	70	80	90	100
EFCM-S	0.9227	0.922	0.9223	0.923	0.9215	0.9218	0.9226	0.9225	0.9231	0.9237
EFCM-SV	0.9257	0.9257	0.9165	0.9257	0.927	0.9165	0.9268	0.927	0.9105	0.9245
+CVNN	$c = 18.5$	$c = 20.7$	$c = 19.4$	$c = 19.3$	$c = 19.3$	$c = 18.2$	$c = 19.2$	$c = 18.3$	$c = 19.4$	$c = 20.2$

(a) RI versus number of dimensions

(b) RI versus number of dimensions

(c) Running time versus number of dimensions

(d) Running time versus number of dimensions

FIGURE 3: Performance of cluster ensemble approaches with different dimensionality.

In Table 3, the values with respect to “EFCM-SV” are the average RI values with the estimated clusters’ numbers for 20 individual runs. The values of “+CVNN” are the average clusters’ numbers decided by the CVNN cluster validity index. Using the estimated clusters’ numbers by CVNN, our method gets the similar results of ensemble method with correct clusters’ number. In addition, the average estimates of clusters’ number are close to the true one. This indicates that our cluster ensemble method EFCM-SV is attractive when the number of clusters is unknown.

6. Conclusion and Future Work

The “curse of dimensionality” in big data gives new challenges for clustering recently, and feature extraction for dimensionality reduction is a popular way to deal with these challenges. We studied the feature extraction method of random projection for FCM clustering. Through analyzing the effects of random projection on the entire variability of data theoretically and verification both on synthetic and real world data empirically, we designed an enhanced FCM clustering algorithm with random projection. The new algorithm can maintain nearly the same clustering solution of preliminary FCM clustering and be more efficient than feature extraction method of SVD. What is more, we also proposed a cluster ensemble approach that is more applicable to large scale data sets than existing ones. The new ensemble approach can achieve spectral embedding efficiently from SVD on the concatenation of membership matrices. The experiments showed that the new ensemble method ran faster, had more robust partition solutions, and fitted a wider range of geometrical data sets.

A future research content is to design the provably accurate feature extraction and feature selection methods for FCM clustering. Another remaining question is that how to choose proper number of random projections for cluster ensemble method in order to get a trade-off between clustering accuracy and efficiency.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported in part by the National Key Basic Research Program (973 programme) under Grant 2012CB315905 and in part by the National Nature Science Foundation of China under Grants 61502527 and 61379150 and in part by the Open Foundation of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (no. SKLNST-2013-1-06).

References

- [1] M. Chen, S. Mao, and Y. Liu, “Big data: a survey,” *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.
- [2] J. Zhang, X. Tao, and H. Wang, “Outlier detection from large distributed databases,” *World Wide Web*, vol. 17, no. 4, pp. 539–568, 2014.
- [3] C. Ordóñez, N. Mohanam, and C. Garcia-Alvarado, “PCA for large data sets with parallel data summarization,” *Distributed and Parallel Databases*, vol. 32, no. 3, pp. 377–403, 2014.
- [4] D.-S. Pham, S. Venkatesh, M. Lazarescu, and S. Budhaditya, “Anomaly detection in large-scale data stream networks,” *Data Mining and Knowledge Discovery*, vol. 28, no. 1, pp. 145–189, 2014.
- [5] F. Murtagh and P. Contreras, “Random projection towards the baire metric for high dimensional clustering,” in *Statistical Learning and Data Sciences*, pp. 424–431, Springer, Berlin, Germany, 2015.
- [6] T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, and M. Palaniswami, “Fuzzy c-means algorithms for very large data,” *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 6, pp. 1130–1146, 2012.
- [7] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques: Concepts and Techniques*, Elsevier, 2011.
- [8] S. Khan, G. Situ, K. Decker, and C. J. Schmidt, “GoFigure: automated gene ontology™ annotation,” *Bioinformatics*, vol. 19, no. 18, pp. 2484–2485, 2003.
- [9] S. Günnemann, H. Kremer, D. Lenhard, and T. Seidl, “Subspace clustering for indexing high dimensional data: a main memory index based on local reductions and individual multi-representations,” in *Proceedings of the 14th International Conference on Extending Database Technology (EDBT ’11)*, pp. 237–248, ACM, Uppsala, Sweden, March 2011.
- [10] J. C. Bezdek, R. Ehrlich, and W. Full, “FCM: the fuzzy c-means clustering algorithm,” *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.
- [11] R. J. Hathaway and J. C. Bezdek, “Extending fuzzy and probabilistic clustering to very large data sets,” *Computational Statistics & Data Analysis*, vol. 51, no. 1, pp. 215–234, 2006.
- [12] P. Hore, L. O. Hall, and D. B. Goldgof, “Single pass fuzzy c means,” in *Proceedings of the IEEE International Fuzzy Systems Conference (FUZZ ’07)*, pp. 1–7, London, UK, July 2007.
- [13] P. Hore, L. O. Hall, D. B. Goldgof, Y. Gu, A. A. Maudsley, and A. Darkazanli, “A scalable framework for segmenting magnetic resonance images,” *Journal of Signal Processing Systems*, vol. 54, no. 1-3, pp. 183–203, 2009.
- [14] W. B. Johnson and J. Lindenstrauss, “Extensions of lipschitz mappings into a Hilbert space,” *Contemporary Mathematics*, vol. 26, pp. 189–206, 1984.
- [15] P. Indyk and R. Motwani, “Approximate nearest neighbors: towards removing the curse of dimensionality,” in *Proceedings of the 13th Annual ACM Symposium on Theory of Computing*, pp. 604–613, ACM, 1998.
- [16] D. Achlioptas, “Database-friendly random projections: Johnson-Lindenstrauss with binary coins,” *Journal of Computer and System Sciences*, vol. 66, no. 4, pp. 671–687, 2003.
- [17] C. Boutsidis, A. Zouzias, and P. Drineas, “Random projections for k-means clustering,” in *Advances in Neural Information Processing Systems*, pp. 298–306, MIT Press, 2010.
- [18] C. C. Aggarwal and C. K. Reddy, *Data Clustering: Algorithms and Applications*, CRC Press, New York, NY, USA, 2013.
- [19] R. Avogadri and G. Valentini, “Fuzzy ensemble clustering based on random projections for DNA microarray data analysis,” *Artificial Intelligence in Medicine*, vol. 45, no. 2-3, pp. 173–183, 2009.

- [20] X. Z. Fern and C. E. Brodley, "Random projection for high dimensional data clustering: a cluster ensemble approach," in *Proceedings of the 20th International Conference on Machine Learning (ICML '03)*, vol. 3, pp. 186–193, August 2003.
- [21] M. Popescu, J. Keller, J. Bezdek, and A. Zare, "Random projections fuzzy c-means (RPFCM) for big data clustering," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE '15)*, pp. 1–6, Istanbul, Turkey, August 2015.
- [22] A. Fahad, N. Alshatri, Z. Tari et al., "A survey of clustering algorithms for big data: taxonomy and empirical analysis," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 267–279, 2014.
- [23] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*, vol. 4, Pearson Prentice Hall, Upper Saddle River, NJ, USA, 6th edition, 2007.
- [24] C. Boutsidis, A. Zouzias, M. W. Mahoney, and P. Drineas, "Randomized dimensionality reduction for k -means clustering," *IEEE Transactions on Information Theory*, vol. 61, no. 2, pp. 1045–1062, 2015.
- [25] X. Chen and D. Cai, "Large scale spectral clustering with landmark-based representation," in *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pp. 313–318, 2011.
- [26] D. Cai and X. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE Transactions on Cybernetics*, vol. 45, no. 8, pp. 1669–1680, 2015.
- [27] G. H. Golub and C. F. Van Loan, *Matrix Computations*, vol. 3, JHU Press, 2012.
- [28] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1650–1654, 2002.
- [29] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.
- [30] D. T. Anderson, J. C. Bezdek, M. Popescu, and J. M. Keller, "Comparing fuzzy, probabilistic, and possibilistic partitions," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 5, pp. 906–918, 2010.
- [31] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, 1991.
- [32] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, and S. Wu, "Understanding and enhancement of internal clustering validation measures," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 982–994, 2013.

Hindawi

Submit your manuscripts at
<http://www.hindawi.com>