International Journal of Database Theory and Application
Vol.9, No.12 (2016), pp.311-330
http://dx.doi.org/10.14257/ijdta.2016.9.12.28

FastMap Projection for High-Dimensional Data: A Cluster Ensemble Approach

Imran Khan[1], Kamen Ivanov[2], and Qingshan Jiang[1]

[1]*Shenzhen Key Laboratory of High Performance Data Mining, Shenzhen Institutes of*
Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China.
[2]*Shenzhen Key Laboratory for Low-cost Healthcare, Shenzhen Institutes of Advanced*
Technology, Chinese Academy of Sciences, Shenzhen 518055, China.
**Corresponding authors: imran.khan@siat.ac.cn, qs.jiang@siat.ac.cn*

Abstract

High-dimensional data with many features present a significant challenge to current clustering algorithms. Sparsity, noise, and correlation of features are common properties of high-dimensional data. Another essential aspect is that clusters in such data often exist in various subspaces. Ensemble clustering is emerging as a leading technique for improving robustness, stability, and accuracy of high-dimensional data clusterings. In this paper, we propose FastMap projection for generating subspace component data sets from high-dimensional data. By using component data sets, we create component clusterings and provides a new objective function that ensembles them by maximizing the average similarity between component clusterings and final clustering. Compared with the random sampling and random projection methods, the component clusterings by FastMap projection showed high average clustering accuracy without sacrificing clustering diversity in synthetic data analysis. We conducted a series of experiments on real-world data sets from microarray, text, and image domains employing three subspace component data generation methods, three consensus functions, and a proposed objective function for ensemble clustering. The experiment results consistently demonstrated that the FastMap projection method with the proposed objection function provided the best ensemble clustering results for all data sets.

Keywords*: FastMap, Ensemble clustering, High-dimensional data, Consensus function*

1. Introduction

The emergence of new application domains results in very high-dimensional big data that is a big challenge in cluster analysis [1-4]. Sparsity, noise, correlation and informativeness of features are basic properties of high-dimensional data in real applications. Another prominent aspect is that clusters in such data usually exist in various subspaces. To effectively cluster high-dimensional data, researchers have proposed different clustering methods, including subspace clustering methods [5-8]. However, most algorithms lack in good clustering performance [9]. The ensemble clustering methods are guaranteeing to solve this problem.

Ensemble clustering is an emerging clustering procedure that combines multiple clusterings produced from samples of a given data set into a single clustering with a result which is usually much better than the results of individual clusterings on the data set [10-11]. Ensemble clustering is more useful in

ISSN: 2005-4270 IJDTA

clustering high-dimensional complex data than the clustering methods that provide single clustering results. However, the clustering ensemble created from a high-dimensional data set is more stable and usually more accurate than any of the individual component clusterings. Due to this advantage, ensemble clustering becomes attractive in clustering high-dimensional data such as text, microarray, and image data [9, 12].

Given a data set, the process of ensemble clustering is performed in two stages, producing a set of individual component clusterings from the data set and combining the component clusterings into a clustering ensemble. The quality of the final clustering ensemble is determined by the methods to carry out these two steps. Different methods result in different ensemble clustering algorithms.

When generating component clusterings, the efforts are mainly concentrated on increasing the diversity of the component clusterings [13]. This is generally achieved in three ways [14]. The first one involves using one clustering algorithm with varying parameter settings [15]. The second approach is to use various clustering algorithms to cluster the same data set to produce different clusterings [16]. Finally, the third approach suggests to sample the given data set to form different component data sets and use a clustering algorithm to cluster them and produce component clusterings [10]. The ensemble clustering is created by utilizing an ensemble function to combine multiple component clusterings into one final clustering. In this step, three consensus functions including the direct method, the feature-based method [17], the graph-based approach [10], and one objective function are used. However, these consensus functions are not suitable for noisy and large data. The main objective of integration is to produce a clustering ensemble with a higher accuracy than the accuracies of the individual component clusterings.

Recently, two methods for generating low-dimensional component data have been used to resolve the problem of ensemble clustering of high-dimensional data. One is to randomly sample distinct subsets of features to generate subspace component data sets [10, 17]. The other is to project the given high-dimensional data into low-dimensional component data sets by randomly generated projection matrices [18]. The main benefit of these methods is that they can generate diverse component clusterings. A serious lapse is that low-dimensional component data sets firmly deviate from the original data set, which leads to a strong difference in the clustering structures between the component data sets and the original data. As an outcome, the quality of component clusterings is significantly reduced.

This paper is a revised and expanded version of a paper entitled Ensemble Clustering of High Dimensional Data with FastMap Projection presented at Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD, 2014, Taiwan [2]. In this paper, we present a new ensemble clustering technique to improve the results. We describe a new low-dimensional component data generation method by FastMap [1], an algorithm that is used to generate a low-dimensional transformation of high-dimensional data. Given a distance matrix of *N* objects, FastMap uses the well known Cosine Law to compute the coordinates of the *N* objects that are projected to the line of two pivot objects selected from the data set. By removing the distance component from the newly generated dimension, a new set of coordinates is computed. This process repeats until the k-dimensional representation of the *N* objects is obtained. The advantage of FastMap projection in comparison with random sampling and random projection is that it

can better preserve the clustering structure of the original data in its generated component data sets. We generate multiple component data sets and apply the well-known *k*-means algorithm to generate the component clusterings. We use a new objective function to ensemble the multiple component clusterings into a single clustering by maximizing the average similarity between component clusterings and the clustering ensemble.

We have conducted a series of experiments on both synthetic and real data to evaluate the FastMap projection in ensemble clustering from different perspectives. We compared the results of ensemble clusterings produced with random sampling, random projection, and FastMap projection. We used three consensus functions and the proposed objective function to generate clustering ensembles. The experimental results showed consistent improvement in accuracy of clustering ensembles produced from the FastMap projection with the objective function in comparison with the random sampling and random projection methods.

The rest of this paper is organized as follows. In section 2, we provide related work. In Section 3, we explain the motivation of this work based on the analysis of synthetic data sets. In Section 5, we describe the FastMap projection method for generating subspace component data sets in ensemble clustering and ensemble clustering method. Section 4 illustrates the experimental results. Finally, we conclude this work and discuss future work in Section 6.

2. Related Work

Many different methods exist for ensemble clustering. According to [19] two main classes of ensemble clustering algorithms can be discovered: methods based on finding the median partition and approaches based on object co-occurrence. While the second class is rather heuristic, the first class builds on the formulation of the generalized median for clusterings. More specifically, the median clustering is the clustering which minimizes the sum of distances (SOD) to the clusterings in the ensemble.

The first work was for ensemble clustering that used the median partition formulation was proposed by Strehl and Ghosh [10]. The authors aimed to maximize the sum of similarity values with the normalized mutual information (NMI) as the distance function. Since the optimization of this objective function is computationally intractable, three approximation heuristics were proposed. The main idea was to represent the ensemble as a graph and to use different graph-based methods to obtain the median partition from this graph. Although the problem formulation was done by using the median partition, the proposed problem solution has to be classified as a co-occurrence based method [19]. Instead of using NMIT Opchy et al. [20] use the category utility function. In this case, a fast optimization scheme is introduced by making use of the observation that the problem can be transformed into another feature space and solved by *k*-means. While the benefit of this approach is its low complexity, the knowledge of the true number of clusters is required by the method.

The Mirkin-metric is used in [21-22] as a distance measure and several simple heuristics based on genetic algorithms like Simulated Annealing or Best One-element Move are used to minimize the SOD criterion. Further methods, which aim to optimize the SOD criterion, include the kernel-based method [23] or the NMF-based method [24]. The evaluation presented in [10] reveals that the median

partition approach corresponds very well to recovering the true labels of data sets. This is further supported by an analysis of the median partition [25]. By presuming some simplifying conditions, the authors have proved that the consensus solution converges against the ground truth. In contrast to the median partition approach the co-occurrence based method uses a voting mechanism. More specifically, it attempts to count how many times two objects belong to the same cluster. This information can be collected into the co-association matrix, which is used as the similarity measure and a clustering algorithm is applied to find the consensus clustering [15]. A refined cluster-association matrix which also takes into account the relations between clusters is proposed in [14]. The columns of the matrix are then interpreted as feature vectors and clustered by an ordinary clustering algorithm.

The ensemble methods discussed above are not suitable for noisy and large data sets. A well-known paper [10] aims at combining soft partitioning of data (e.g., produced by fuzzy *k*-means) without hardening the partitions before entering them into a consensus mechanism. The authors propose soft versions of CSPA, HGPA, and MCLA. Our work on ensemble clusterings differs from all the previous approaches that include objective functions, weighted ensemble clustering, and subspace ensemble clustering. Subspace ensemble clustering has become a useful strategy to find robust clusters from sparse and high-dimensional data.

3. Motivation

The diversity of the input clusterings in ensemble clustering plays a significant role to generate a final partition that is superior to the participating ones. In this section, we demonstrate the superiority of proposed component data sets generation method to produce diverse input clusterings.

We analyzed the results on synthetic data sets to demonstrate the problems of the random sampling and random projection methods in the generation of low-dimensional data sets for component clusterings. We generated six synthetic data sets, each consisting of one hundred features and three clusters. Each cluster was composed of fifty data points in one hundred dimensions with a normal distribution. We set the means of the three clusters in the main dimensions as 1, 5, 3, respectively and the same unit variance for all clusters. The three clusters were generated independently and merged into one data set. Then, some noisy features with a uniform distribution between 0 and 1 were added to the data set to replace the same number of features with cluster distributions. As shown in Figure 1, we generated six data sets by adding different percentages of noise features, i.e. 0.05, 0.1, 0.2, 0.5, 0.6, and 0.7, respectively. The blue, yellow, and green colors in each circle (data set) represent the clusters with means 1, 5, and 3, respectively. The red color shows the percentage of noisy features. The presence of more noisy features in the data set determines the difficulty of clustering.

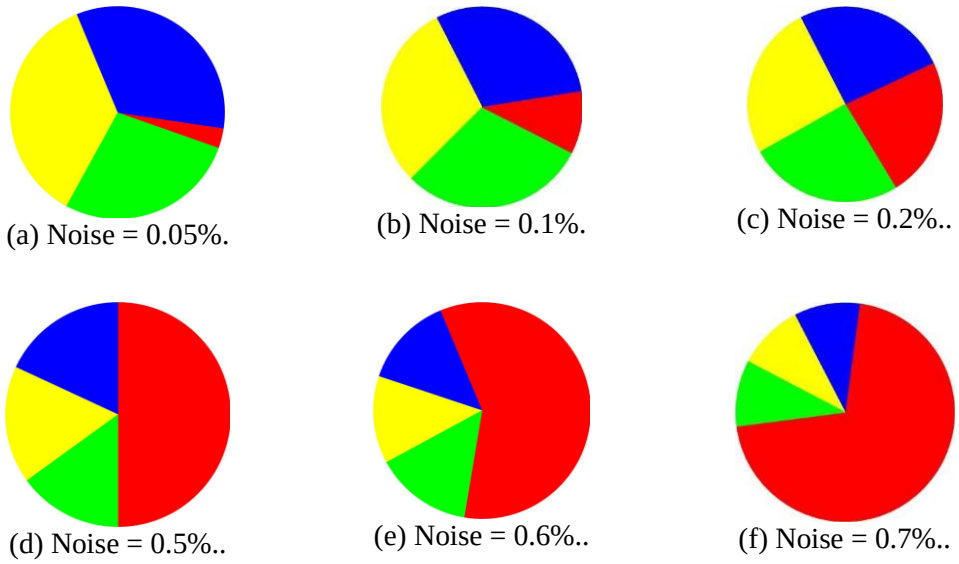


Figure 1. Six Synthetic Data Sets with Different Proportion of Noisy Features

For component clusterings, we used both random sampling and random projection methods on each synthetic data set to generate two hundred low-dimensional data sets, one hundred data sets with each method. The random sampling method randomly selected *p* features from each data set to produce a low-dimensional component data set. The random projection method projects the given data set into *p* dimensions by multiplying a random matrix $R_{d\times p}$ to the given synthetic data set where *d* is the number of dimensions in the synthetic data set. In practice, $p = q \times d$ where *q* is the sampling rate expressed in percents. The values of the random projection matrix *R* were randomly produced with a normal distribution.

We used the well-known *k*-means algorithm to cluster each of the one hundred component data sets of each sampling method into three clusters and computed the accuracy of obtained component clustering. We divided the one hundred clustering results into six accuracy groups of ([0,0.5], [0.5,0.6], [0.6,0.7], [0.7,0.8], [0.8,0.9], [0.9,1]). Figure 2 shows the frequencies of the clustering results in the different accuracy groups on the six data sets with a sampling rate of s = 2%. The yellow bars show the results of the random sampling method and the blue bars represent the results of the random projection method. The red bars in the figure show the clustering results from the component data sets generated with the FastMap projection. The details of the FastMap projection will be discussed later.

In Figure 2, we can see that many of the clusterings from random projection and random sampling fall into the low accuracy groups. The share of the clusterings falling into the higher accuracy groups is low. The probability of obtaining an accurate clustering ensemble using such clusterings is very low. The accuracy of the component clustering decreases as the number of noisy dimensions increase in the data set. The random projection method produced more consistent results than those of the random sampling method.

The diversity of component clusterings of each data set from Figure 2 is investigated by computing the normalized mutual information (NMI) between each pair of one hundred component clusterings. The computed NMI values are divided into six groups of ([0,0.5], [0.5,0.6], [0.6,0.7], [0.7,0.8], [0.8,0.9], [0.9,1]). Figure 3 shows the frequencies of NMI values in the six groups from the six data sets in Figure 2 with a sampling rate of 2%. The yellow and blue bars represent the results generated by the random sampling and random projection methods, respectively. A larger NMI value shows that two clustering results have a strong relation, and a lower NMI value represents that two clustering results are independent. A lower NMI value indicates more diverse component clusterings. We can see that when noisy dimensions increase, the falling of NMI values into the highest group significantly reduced. This shows that the noisy dimensions increase the diversity of component clusterings.

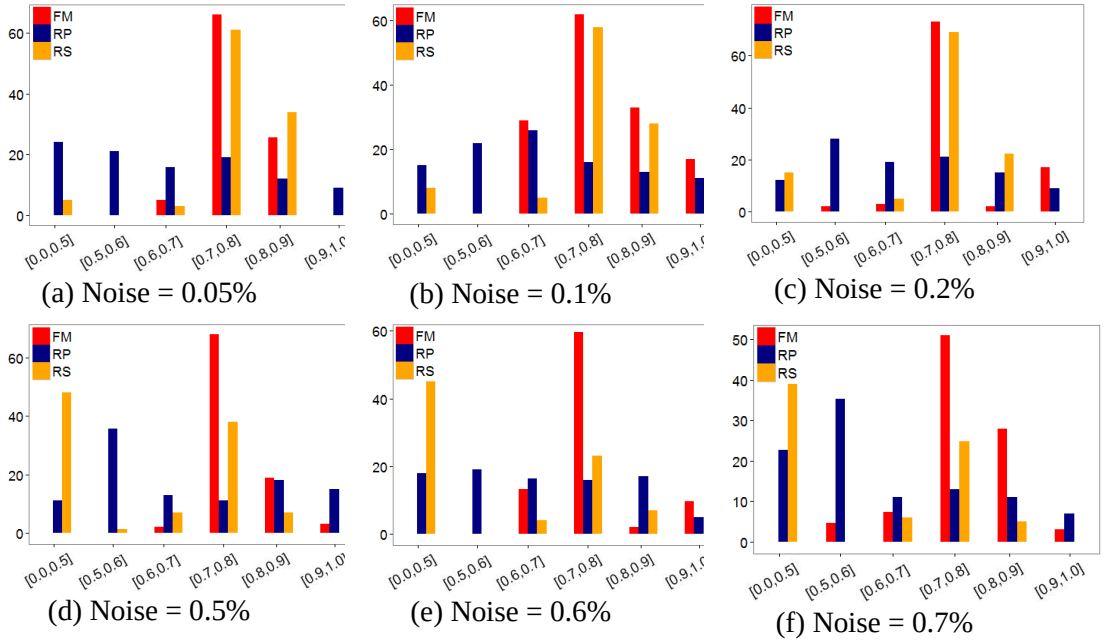


Figure 2. Clustering results in different accuracy groups (x-axis shows the accuracy intervals, and the y-axis indicates the frequency of results where the clustering accuracy falls into the corresponding interval). The yellow, blue, and red bars show the results of the random sampling (RS), random projection (RP), and FastMap (FM) projection, respectively

Figure 3 shows that both random sampling and random projection methods generated diverse component clusterings. However, as shown in Figure 2, many of these diverse component clusterings are in lower accuracy groups because of the noisy features in the component data sets. To address this problem, we propose a FastMap projection method to produce component data sets. FastMap projection preserves the clustering structure of the original data in the component data sets. As a consequence, the performance of ensemble clustering improves significantly. The red bars in Figure 2 demonstrate the results produced with the FastMap projection method. The clustering results generated by this method mostly fall into the higher accuracy groups. However, the diversity of component clusterings

remains almost unchanged, as shown in Figure 3. In the next section, we present the FastMap method in details.

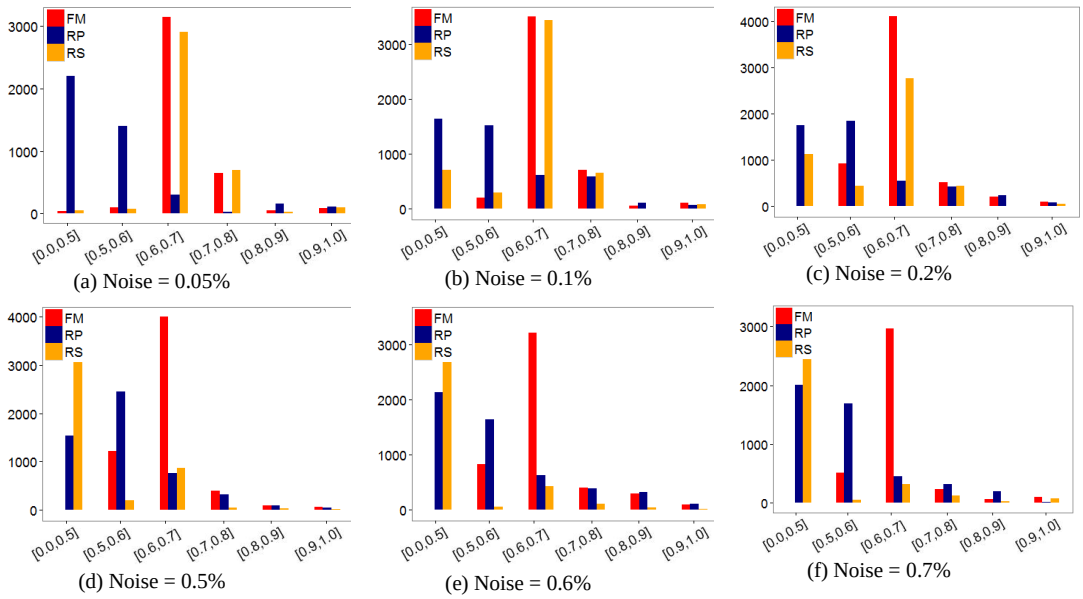


Figure 3. Distributions of the NMI Values between Pairs of One Hundred Component Clusterings Shown in Figure 2 on the Six Synthetic Data Sets

4. Proposed Scheme

Ensemble clustering of a data set *X* is a process to integrate multiple clustering results produced by one or more clustering algorithms from component data sets sampled from *X* into a single clustering of *X* with a result that is usually much better than the results of individual clusterings on *X* [20]. The subspace ensemble clustering framework consists of the following steps.

● **Step1** : Generate *K* different component data sets $\{C_1, C_2, ..., C_K\}$ from *X* using a component generation method.

● **Step2** : Cluster the *K* component data sets to produce *K* component clusterings $\{\lambda^1, \lambda^2, ..., \lambda^K\}$ independently using one or more clustering algorithms.

● **Step3** : Ensemble *K* component clusterings into a single clustering λ using an ensemble method called a consensus function.

Figure 4 shows a generic framework of ensemble clustering.

4.1 FastMap Projection for Component Data Generation

FastMap is introduced as an alternative to Multidimensional Scaling (MDS) [26] and a generalization of Principal Component Analysis (PCA) [27]. FastMap is an efficient algorithm to generate k-dimensional coordinates of *N* objects from a distance matrix of *N* objects. Given a high-dimensional data set *X* of *m* dimensions and *N* objects, a distance function is used to compute the distance matrix $S_{N\times N}$.

The only distance matrix input we have for data projection is S and it should satisfy the following properties of the triangle inequality:

- $S_{N\times N}(O_a, O_b) = 0$,
- $S_{N\times N}(O_a, O_b) = S_{N\times N}(O_b, O_a)$,
- $S_{N\times N}(O_i, O_b) \le S_{N\times N}(O_i, O_a) + S_{N\times N}(O_b, O_a)$

where O_i, O_a and O_b are the objects of X. The well-know Euclidean distance function or cosine similarity [28] between data objects is used to build a distance matrix S.

$$Similarity(O_a, O_b) = \frac{\overrightarrow{O_a} \circ \overrightarrow{O_b}}{\left\|\overrightarrow{O_a}\right\|_2 . \left\|\overrightarrow{O_b}\right\|_2} \tag{1}$$

where $\circ$ is the inner product of two vectors and $\|*\|_2$ represents the Euclidean norm of the vector. The similarity of two vectors $\overrightarrow{O_a}$ and $\overrightarrow{O_b}$ is measured by considering an angle $\cos(\theta)$. The cosine similarity is used to project all the vectors on the unit hyper-sphere and measures the cosine angle of the projections. In order to be used for FastMap, a distance function is defined that decreases with increasing of similarity.

$$\begin{aligned} D(O_a, O_b) &= 2 * \sin(\theta / 2) \\ &= \sqrt{2 * (1 - \cos(\theta))} \\ &= \sqrt{2 * (1 - Similarity(O_a, O_b))} \end{aligned} \tag{2}$$

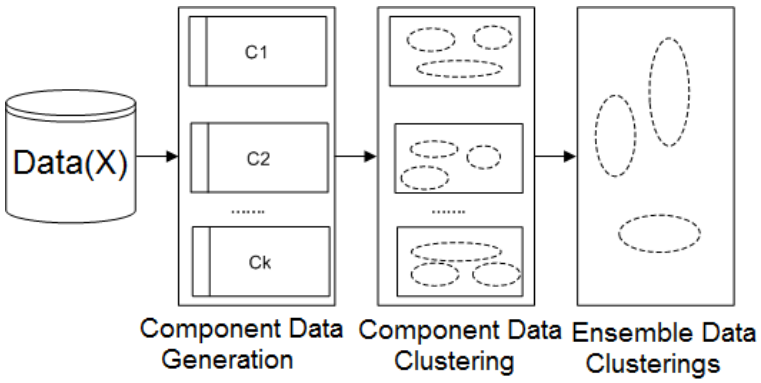


Figure 4. Generic Framework of Ensemble Clustering

In our experiments, we used equation 2 to generate the distance matrix $S_{N\times N}$. A core step of FastMap projection is to carefully select a line for data projection. To do that, two pivot objects O_a and O_b are selected, and a line is considered that passes through them in a given space. The pivot objects O_a and O_b are chosen

which are far apart from each other in a given data set. The coordinates of data objects on the selected line are computed by using cosine law. The first dimension of an object Oi is computed by using the following cosine equation.

$$D_{b,i}^2 = D_{a,i}^2 + D_{a,b}^2 - 2m_i D_{a,b} \tag{3}$$

By using Pythagoras theorem, equation 3 is used to compute the first coordinate m_i of an object O_i as

$$m_i = \frac{D_{a,i}^2 + D_{a,b}^2 - D_{b,i}^2}{2D_{a.b}} \tag{4}$$

where $D_{a,i}$ is a distance between pivot objects O_a and O_i, for $i = 1,2,....,N$. The coordinates of all N objects are computed, according to Lemma 1 in [1], a reduced distance matrix S' of N objects is computed as

$$D'(O_i, O_j) = \sqrt{D(O_i, O_j)^2 - (m_i - m_j)^2} \tag{5}$$

where D' is the reduced distance in $S'_{N\times N}$, D is the distance in $S_{N\times N}$, m_i and m_j are computed coordinates of the previous dimension for all $I = 1,2,...., N$. Given $S'_{N\times N}$, a new pair of pivot objects is chosen and equation 4 is used to compute the coordinates of the second dimension. We repeat this process k times to generate k-dimensional component data sets using X.

Using FastMap, we can use a random process to select different pairs of pivot objects to produce different projections of data as component data sets. We employ the well-known k-means algorithm on each component data set to generate component clusterings. In the next section, we propose an ensemble method to combine the generated component clusterings into one clustering solution.

4.2 Ensemble Clustering

Given a set of component data sets, we apply the k-means algorithm on each data set to produce component clusterings. Let λ^1 , λ^2, ..., λ^e be e component clusterings. They can be represented into a matrix E as

$$E = \begin{array}{c} \\ 1 \\ 2 \\ . \\ . \\ N \end{array} \begin{array}{c} \begin{array}{cccccc} \lambda^1 & \lambda^2 & . & . & \lambda^e \end{array} \\ \begin{bmatrix} l_{11} & l_{12} & . & . & l_{1e} \\ l_{21} & l_{22} & . & . & l_{2e} \\ . & . & . & . & . \\ . & . & . & . & . \\ l_{N1} & l_{N2} & . & . & l_{Ne} \end{bmatrix} \end{array} \tag{6}$$

where each row is an object and each column is the set of cluster labels of *N* objects in a clustering. Each cluster in the e clusterings has a unique label. The set of unique cluster labels is listed as

$$L = \{l_{11}, l_{21}, l_{12}, l_{32}, l_{13}, \ldots, l_{ke}\} \tag{7}$$

Taking two columns λ^x and λ^y, we now define the Normalized Mutual Information (NMI) [20] between clusterings λ^x and λ^y as

$$NMI(\lambda^x, \lambda^y) = \frac{I(\lambda^x, \lambda^y)}{\sqrt{H(\lambda^x)H(\lambda^y)}} \tag{8}$$

where $I(\lambda^x, \lambda^y)$ is the mutual information between clusterings λ^x and λ^y, and $H(\lambda^i)$ is the entropy of the clustering. The mutual information is defined as

$$I(\lambda^x, \lambda^y) = \sum_{X_c^x \in \lambda^x} \sum_{X_c^y \in \lambda^y} \frac{\left|X_c^x \cap X_c^y\right|}{N} \log_2 \left(\frac{N\left|X_c^x \cap X_c^y\right|}{\left|X_c^x\right|\left|X_c^y\right|} \right) \tag{9}$$

where X_j^i is the set of data points in component data set that are in the same cluster *j* in clustering λ^i, *N* is the total number of data points, and |.| and $\cap$ are the cardinality and intersection operators, respectively. Next, we define the entropy $H(\lambda^i)$ of a clustering, λ^i, as

$$H(\lambda^i) = -\sum_{X_c^i \in \lambda^i} \frac{\left|X_c^i\right|}{N} \log_2 \left(\frac{\left|X_c^i\right|}{N} \right) \tag{10}$$

The MNIs of all pairs of clusterings are represented in matrix R as

$$R = \begin{array}{c} \\ \lambda^1 \\ \lambda^2 \\ . \\ . \\ \lambda^e \end{array} \begin{array}{c} \begin{array}{ccccc} \lambda^1 & \lambda^2 & . & . & \lambda^e \end{array} \\ \begin{bmatrix} NMI_{11} & NMI_{12} & . & . & NMI_{1e} \\ NMI_{21} & NMI_{21} & . & . & NMI_{2e} \\ . & . & . & . & . \\ . & . & . & . & . \\ NMI_{e1} & NMI_{e2} & . & . & NMI_{ee} \end{bmatrix} \end{array} \tag{11}$$

From R, we select a clustering as the reference clustering λ^r by computing the average of each row. The largest average value of the row *r* gives a reference clustering λ^r.

Given the column vector of the reference clustering λ^r and the set of cluster labels L, we replace the cluster label of the first object in the reference clustering λ^r with the first label in L to generate a changed reference clustering $\lambda^{r\prime}$. Then, we compute the average of the normalized mutual information between $\lambda^{r\prime}$ and all other clusterings as

$$NMI_a(\Delta, \lambda^{r'}) = \frac{1}{(e-1)} \sum_{i=1}^{(e-1)} NMI(\lambda^{r'}, \lambda^i) \tag{12}$$

where Δ is the set of clusterings excluding the reference clustering. If $NMI_a(\Delta,\lambda^{r\prime}) > NMI_a(\Delta,\lambda^r)$, we replace λ^r with $\lambda^{r\prime}$ and $NMI_a(\Delta, \lambda^r)$ with $NMI_a(\Delta, \lambda^{r\prime})$. Otherwise, we keep both λ^r and $NMI_a(\Delta,\lambda^r)$ unchanged. We continue this process until all labels in L are tested. After this iterative loop, the first object in the reference clustering is assigned a cluster label that maximizes $NMI_a(\Delta,\lambda^r)$. The same iterative process is repeated until the last object is complete. Then, the process restarts from the first object of λ^r. In each loop on N objects, the number of changes of object labels is recorded. The iterative process stops when no object changes its cluster label after a loop on N objects. The reference clustering λr is the final clustering ensemble.

5. Experimental Results and Analysis

In this section, we present a series of experiments on real-world data to demonstrate the performance of ensemble clusterings with the FastMap projection method in generating component data sets. We show comparisons of random sampling, random projection, and FastMap projection in combination with three consensus functions and an objective function based ensemble clustering method.

5.1 Data Sets

Six high-dimensional data sets were used in these experiments. All data sets are diverse in the number of records, the number of features and the number of clusters. Detailed information of the data sets is shown in Table 1. The data sets BASEHOCK, GLI85, and PIX10P were chosen from the available websites dedicated to data mining at Arizona state university. The SRBCT and Internet Ad were chosen from the web site of UCI machine learning repository. The data set La1s was used as a text document classification benchmark [29]. Data with heterogeneous characteristics is important for exploring the strength and weakness of algorithms in different applications.

5.2 Experiment Settings

The performance of FastMap (FM) projection is investigated by comparing with Random Sampling (RS) and Random Projection (RP). We produced component data set using each component data generation method for the given data set, and applied the *k*-means algorithm on each component data set to generate component clusterings. The component clusterings from each method are aggregated into one clustering ensemble by using three consensus functions and one objective function. Combining three component data set generation methods with three consensus functions and one objective function, we investigated twelve ensemble clustering techniques. We compared the ensemble clustering results with

true class labels in the data sets and used three evaluation measures to evaluate the performance of the twelve ensemble clustering techniques.

For random projection, Boutsidis et al. [30] recommended to set the dimensions of projected data sets as d = k/∂^2 where $\partial \in (0, 0.34)$ and *k* is the number of true labels in the original data set. ∂ was determined through some initial tests. In the experiments, ten values for ∂ were tested and the best results were recorded.

The well-known three consensus functions are similarity-based consensus function (CSPA), hypergraphbased consensus function (HGPA), and meta cluster-based consensus function (MCLA) [10]. The ensemble method we propose here is called Objective Function based Ensemble Clustering (OFEC).

The combinations of three component data generation methods with three consensus functions and one objective function result in 12 ensemble clustering techniques denoted as RS-CSPA, RP-CSPA, FM-CSPA, RS-HGPA, RP-HGPA, FM-HGPA, RS-MCLA, RP-MCLA, FM-MCLA, RS-OFEC, RP-OFEC, and FMOFEC, respectively. We used the baseline clustering algorithm *k*-means on each original data set ten times. The average result of obtained clusterings from each data set are presented. We denoted this method as KM-Avg.

In experiments, we tested three different numbers of component clusterings to generate clustering ensembles. The results showed no significant variation. The results given below were taken from the clustering ensembles with ten component clusterings. We also tested different sampling rates and set 15% which is more suitable for all data sets in the following results. The number of clusters *k* was set to be the actual number of classes in the real-world data sets.

Table 1. Real-world Data Sets

Data Sets	#Instances	#Features	Source	#Classes
PIX10P	100	10,000	Image	10
BASEHOCK	1993	4862	Text	02
GLI85	c85	22,283	Microarray	02
La1s	887	13,195	Text	06
SRBCT	83	2308	Microarray	04
Internet Ad	1000	1558	Multivariate	02

5.3 Evaluation Methods

Clustering evaluation is a critical and often ill-posed task. In fact, many kinds of objective clustering functions were defined [29]. We used three methods to evaluate the results of ensemble clustering with the twelve ensemble clustering methods, one unsupervised method and two supervised methods. The unsupervised method is Compactness (CP) which is computed as

$$CP = \frac{1}{n}\sum_{x=1}^{k} n_x \left(\frac{\sum_{o_i o_j \in C_x} d(o_i, o_j)}{n_x (n_x - 1/2)} \right) \tag{13}$$

where $d(o_i, o_j)$ is the distance between the objects o_i and o_j in a cluster C_x, n_x is the number of objects in a cluster C_x. The smaller the value of CP, the better the

clustering result. The two supervised evaluation methods are Adjusted Rand Index (ARI) and Clustering Accuracy (CA), calculated as follows

$$CA = \frac{1}{n}\sum_{x=1}^{k} max_y n_{x,y}$$

$$RI = \frac{\sum_{x=1}^{k}\sum_{y=1}^{k}\binom{n_{x,y}}{2} - 3}{\frac{1}{2}(s_1 + s_2) - s_3} \tag{14}$$

where $n_{x,y}$ is the total number of objects in cluster x and class y, n is the total number of objects in the given data set $s_1 = \sum_{x=1}^{k}\binom{n_x}{2}$, $s_2 = \sum_{y=1}^{k}\binom{n_y}{2}$ and $S3 = 2 S_1 S_2/n(n-1)$. The larger the values of these measures, the better the clustering result.

5.4 Experimental Results

Table 2 shows the evaluations of clustering results of six real-world data sets produced with twelve ensemble clustering techniques and one baseline clustering method. The evaluation on each data set is performed into five groups, the first four groups are the evaluations of the ensemble clustering results, and the last group is the baseline evaluation. Each group in the ensemble clustering techniques lists the evaluations of one ensemble clustering function in combination with three component data generation methods. The best results of the three evaluation measures are marked in bold font. The best result of each method for the same data set and evaluation measure is underlined.

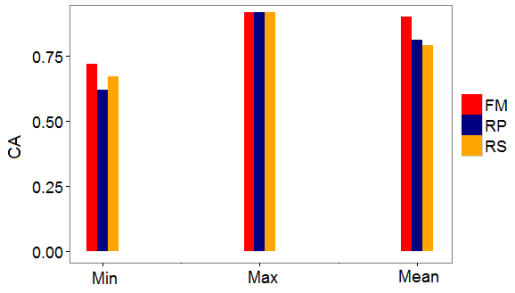
Table 2. Comparison of Clustering Results on Real-world Data Sets

Methods	**PIX10P**			**BASEHOCK**			**GLI85**		
	CP	**ARI**	**CA**	**CP**	**ARI**	**CA**	**CP**	**ARI**	**CA**
RS-CSPA	1123	0.61	0.68	22.9	0.46	0.54	561530	0.51	0.79
RP-CSPA	**1025**	0.82	0.88	21.1	0.53	0.59	542330	0.57	0.78
FM-CSPA	1021	0.89	**0.96**	**20.0**	**0.57**	**0.67**	**529280**	**0.61**	**0.84**
RS-HGPA	1152	0.77	0.85	20.8	0.46	0.50	581760	0.45	0.78
RP-HGPA	1096	0.69	0.79	**20.2**	0.49	0.54	568010	0.49	0.80
FM-HGPA	**1009**	**0.87**	**0.89**	21.5	**0.52**	**0.55**	**525600**	**0.54**	**0.82**
RS-MLCA	998	0.71	0.82	24.9	0.52	0.57	570990	0.47	0.74
RP-MLCA	988	0.76	0.84	24.4	**0.57**	0.60	547150	0.55	**0.76**
FM-MLCA	**977**	**0.78**	**0.92**	**21.1**	0.55	**0.65**	**522340**	**0.61**	0.75
RS-OFEC	942	0.79	0.85	23.1	0.53	0.59	552314	0.58	0.80
RP-OFEC	898	0.83	0.90	20.2	**0.58**	0.62	538140	0.63	0.83
FM-OFEC	**878**	**0.89**	**0.94**	**19.3**	0.57	**0.69**	**521367**	**0.64**	**0.84**
KM-Avg	964	0.74	0.84	21.8	0.41	0.46	558310	0.38	0.68
Methods	**SRBCT**			**La1s**			**Internet Ad**		
	CP	**ARI**	**CA**	**CP**	**ARI**	**CA**	**CP**	**ARI**	**CA**
RS-CSPA	7.76	0.09	0.49	36.8	0.57	0.49	171	0.54	0.65

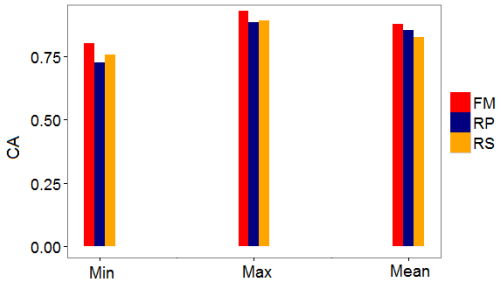
RP-CSPA	7.53	**0.16**	0.50	36.0	0.67	0.54	164	**0.59**	0.65
FM-CSPA	**7.23**	0.13	**0.51**	**34.9**	**0.72**	**0.64**	**145**	0.57	**0.75**
RS-HGPA	8.31	0.03	0.45	36.9	0.61	0.40	241	0.51	0.52
RP-HGPA	7.64	0.08	0.49	36.9	0.69	0.48	216	0.53	0.53
FM-HGPA	**7.13**	0.12	**0.51**	**36.7**	**0.72**	**0.54**	**202**	**0.56**	**0.54**
RS-MLCA	7.43	0.04	0.48	37.0	0.66	0.49	170	0.48	0.55
RP-MLCA	7.49	**0.09**	0.51	35.7	0.68	0.60	153	0.55	0.64
FM-MLCA	**7.41**	0.08	**0.54**	**34.8**	**0.73**	**0.62**	**141**	**0.64**	**0.74**
RS-OFEC	7.33	0.09	0.50	36.2	0.69	0.52	168	0.57	0.64
RP-OFEC	**7.10**	0.13	0.53	20.3	0.72	0.61	149	**0.65**	0.69
FM-OFEC	7.19	**0.15**	**0.53**	**34.2**	**0.75**	**0.67**	**139**	0.63	**0.72**
KM-Avg	7.38	0.09	0.20	34.7	0.67	0.51	179	0.47	0.49

In Table 2, we can see that in the six data sets all best results evaluated by the three supervised measures were obtained using the ensemble clustering techniques, not from the baseline clustering technique. This indicates that the ensemble clustering techniques are more suitable to these six high-dimensional data sets than the baseline clustering method. Among the three component data generation methods, the FastMap projection method consistently produced the best result. The consistent results demonstrated the superiority of the FastMap method over the random sampling and random projection methods in ensemble clustering. Comparing the different ensemble clustering functions, the Objective Function for Ensemble Clustering (OFEC) performed the best.

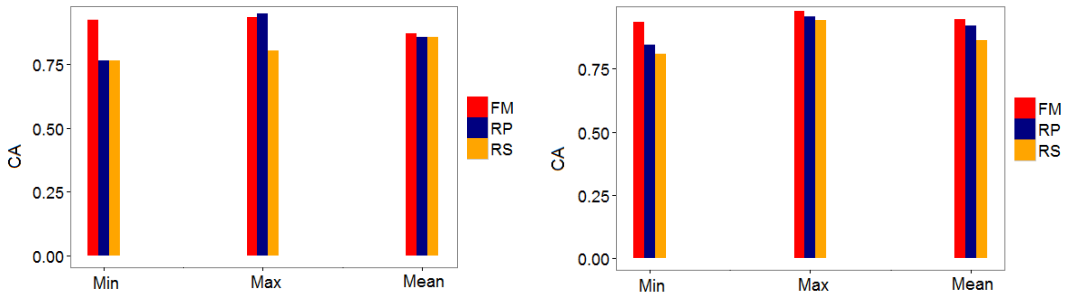
The unsupervised measure CP shows that the well-separated clusters are achieved by FM-OFEC ensemble method for all data sets except the SRBCT data set. The higher CP was achieved on GLI85 data set while the lower CP achieved on SRBCT data set. The majority of the best results of evaluation measures ARI and CA were achieved by the FM-OFEC among the ensemble methods. The comparative analysis between ensemble methods and baseline clustering method shows that the performance of ensemble methods is the best. In all scenarios, FM-OFEC outperformed all other state-of-the-art methods.



(a) CSPA.



(b) HGPA.



(c) MCLA. (d) OFEC.

Figure 5. Distributions of Min, Max and Mean of Accuracies of Thirty Clustering Ensembles from PIX10P Data by Three Component Data Generation Methods with Each Ensemble Function

The detailed analysis of the statistical performance of ensemble clustering methods can be evaluated by generating multiple clusterings ensemble from the given data set. We generated thirty clustering ensembles by each ensemble clustering method from a data set, and computed the minimum, maximum, and mean of accuracies of the thirty clusterings. Each clustering ensemble was generated from ten component clusterings which were generated with k-means. Figure 5 shows the distributions of Min, Max, and Mean of accuracies of thirty clustering ensembles from PIX10P data set by three component data generation methods in the combination of four ensemble functions. As it can be seen from the figure where we compare the Min, Max, and Mean of three component data generation methods with each ensemble function, the FastMap method produced the largest Min, Max, and Mean values in all ensemble functions. The results in OFEC are more significant. Similar trends were also observed from other five data sets.

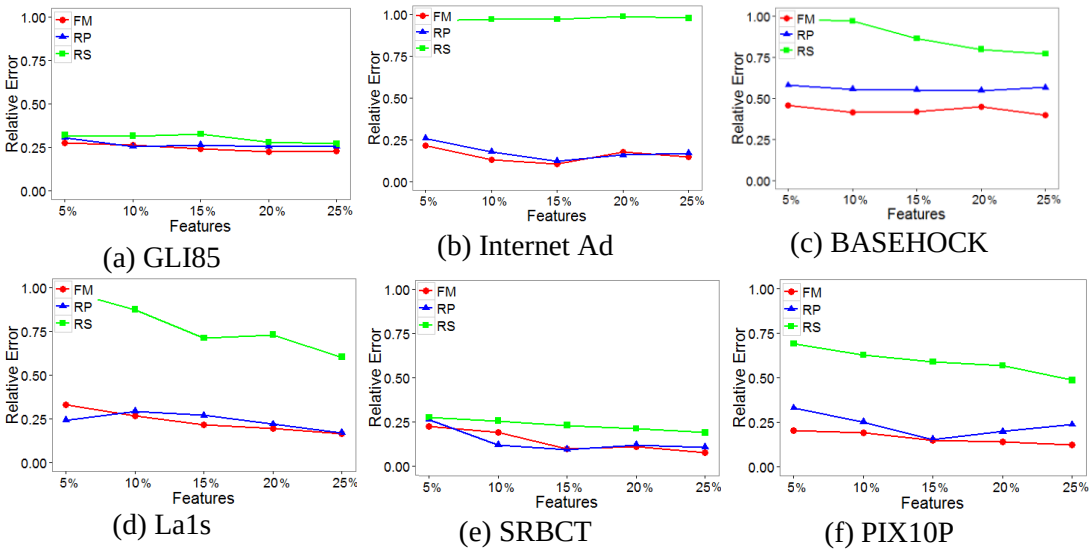
5.5 Analysis of Sampling Rates and Future Strata

The impact of sampling rate of features in component data sets and the number of feature groups or strata on the performance of clustering ensembles was also investigated in the experiments. The optimal sampling rate allows to properly represent the original data (structure), while if it is too small it could lead to overfitting errors. To evaluate the influence of the sampling rate, we used a distance-based measure called relative measure, computed for different sampling rates of the component data sets. The relative distance error is computed by selecting the one hundred instances randomly from the original data set, and also the one hundred instances from the component data set, measuring a 100 × 100 distance matrix using each set of one hundred instances. Then, the relative error can be defined as

$$Error = \sqrt{\frac{1}{10000}\sum_{i}\left(\frac{fC_i}{O_i} - 1\right)^2} \tag{15}$$

where O_i is the distance matrix of one hundred instances from the original data set, C_i is the distance matrix of one hundred instances from the component data set, and f is a scaling factor. The smaller the value, the better component data set. The scaling factor is used to minimize the cost function. It is defined as

$$f = \frac{\sum_i C_i / O_i}{\sum_i C_i^2 / O_i^2} \tag{16}$$



(a) GLI85 (b) Internet Ad (c) BASEHOCK

(d) La1s (e) SRBCT (f) PIX10P

Figure 6. Performance of Component Data Generation Methods Against Different Sampling Rates

Figure 6 shows the average relative distance error of the component data sets for various sampling rates of each data set. We generated component data sets by three different component data generation methods. In ensemble clustering, the component data sets with different dimension sizes may allow for recovering the original clustering structure of data, but our experiments were performed with the same sampling rate for majority of the data sets when the sample rate increased with 15%, the performance started to drop. This is because the diversity of component clusterings decreases as sampling rate increases to a certain level, which starts to affect the performance of clustering ensembles. We can see from the figure that the suitable sampling rate is approx 15%. We can also find that the overall performance of FastMap projection for different sampling rates is better than the performance of the random sampling and random projection. The random sampling and random projection methods cannot preserve well the clustering structure of the original data in their generated low-dimensional component data sets, which leads to increasing of the discrepancy of clustering structures in component data sets, thus affecting the performance of ensemble clustering for high-dimensional data.

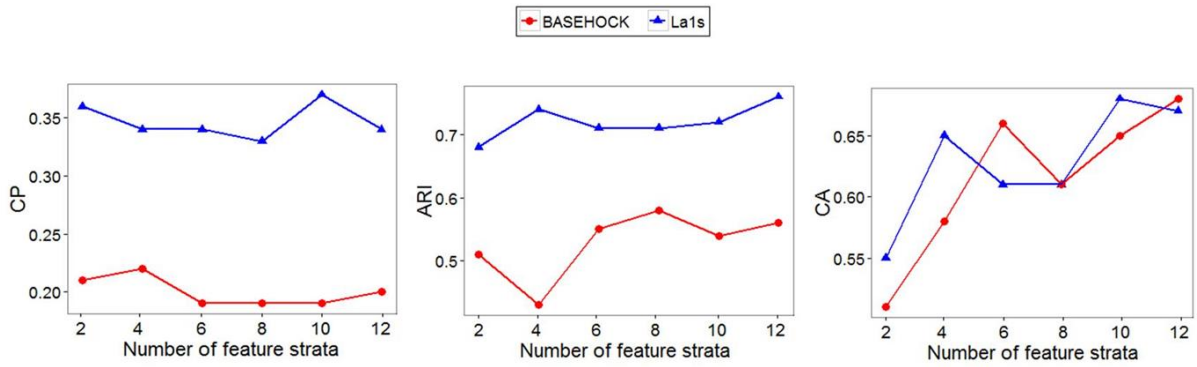


Figure 7. Performances of Clustering Ensembles of BASEHOCK and La1s Data Sets by FMOFEC Technique Against Different Numbers of Feature Strata

Figure 7 illustrates the performances of clustering ensembles of two data sets BASEHOCK and La1s by the FMOFEC technique against different numbers of feature strata. From the results of the three data sets, we can see that there is no significant difference of evaluation measures against the different number of feature strata. Based on CA and ARI evaluation measure the number of feature strata for both data sets should be more than twelve. The same results were also observed from other data sets and consensus functions.

6. Conclusion

In this paper, we have presented the FastMap projection method for generating subspace component data sets in ensemble clustering. This method can better preserve the clustering structure of the original data in its generated component data sets. As a result, the component clusterings created from these data sets have high accuracies in comparison with the results from the methods of random sampling and random projection while at the meantime, the diversity of the component clusterings is not sacrificed much. We have defined a new objective function to ensemble component clusterings by maximizing the average similarity between the component clusterings and the final clustering ensemble. Experiment results on six real-world data sets have demonstrated a consistent performance of the FastMap projection method with the proposed objective function in ensemble clustering. Our future goal is to investigate parallel and distributed algorithms for ensemble clustering with FastMap projection to resolve big data clustering problem.

Acknowledgements

This research work was supported by Shenzhen Technology Development Foundation Grants No. CXZZ20150813155917544 and No. JSGG20160229123657040.

References

[1] H. P. Kriegel, P. Kroger and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 3, no. 1, **(2009)**, p. 1.
[2] I. Khan, J. Z. Huang, N. T. Tung, and G. Williams, "Ensemble clustering of high dimensional data with fastmap projection", in Pacific-Asia Conference on Knowledge Discovery and Data Mining, **(2014)**, pp. 483-493.
[3] I. Khan, J. Huang, and K. Ivanov, "Incremental density-based ensemble clustering over evolving data streams", Neurocomputing, vol. 191, **(2016)**, pp. 34-43.
[4] I. Khan, J. Huang, and N. Tung, "Learning time-based rules for prediction of alarms from telecom alarm data using ant colony optimization", Intl. Journal of Computer and Information Technology, vol. 13, no. 1, **(2013)**.
[5] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan and D. Papadopoulos, "Locally adaptive metrics for clustering high dimensional data", Data Mining and Knowledge Discovery, vol. 14, no. 1, **(2007)**, pp. 63-97.
[6] L. Jing, M. K. Ng and J. Z. Huang, "An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data", IEEE Transactions on knowledge and data engineering, vol. 19, no. 8, **(2007)**, pp. 1026-1041.
[7] W. Limin, Z. Li, H. Xuming, J. Qiang, M. Guangyu, and L. Ying, "An improved affinity propagation clustering algorithm based on entropy weight method and principal component analysis", International Journal of Database Theory and Application, vol. 9, no. 6, **(2016)**, pp. 227-238.
[8] L. Cui, D. Pi and C. Wang, "Topic discovery algorithm based on mutual information and label clustering under dynamic social networks", International Journal of Database Theory and Application, vol. 9, no. 5, **(2016)**, pp. 169-180.
[9] C. Domeniconi and M. Al-Razgan, "Weighted cluster ensembles: Methods and analysis", ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 2, no. 4, **(2009)**, p. 17.
[10] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions", Journal of machine learning research, vol. 3, **(2002)**, pp. 583-617.
[11] P. Hore, L. O. Hall and D. B. Goldgof, "A scalable framework for cluster ensembles", Pattern Recognition, vol. 42, no. 5, **(2009)**, pp. 676-688.
[12] Z. Yu, H. S. Wong and H. Wang, "Graph-based consensus clustering for class discovery from gene expression data", Bioinformatics, vol. 23, no. 21, **(2007)**, pp. 2888-2896.
[13] L. I. Kuncheva and S. T. Hadjitodorov, "Using diversity in cluster ensembles", in Systems, man and cybernetics, 2004 IEEE international conference on, vol. 2, **(2004)**, pp. 1214-1219.
[14] N. I. On, T. Boongoen, S. Garrett and C. Price, "A link-based approach to the cluster ensemble problem", IEEE transactions on pattern analysis and machine intelligence, vol. 33, no. 12, **(2011)**, pp. 2396-2409.
[15] A. L. N. Fred and J. Anil, "Combining multiple clusterings using evidence accumulation", IEEE transactions on pattern analysis and machine intelligence, vol. 27, no. 6, **(2005)**, pp. 835-850.
[16] H. Ayad and M. Kamel, "Finding natural clusters using multi-clusterer combiner based on shared nearest neighbors," in International Workshop on Multiple Classifier Systems, **(2003)**, pp. 166-175.
[17] B. M. Bidgoli, A. P. Topchy and W. F. Punch, "A comparison of resampling methods for clustering ensembles", in IC-AI, **(2004)**, pp. 939-945.
[18] X. Z. Fern, Brodley, and E. Carla, "Random projection for high dimensional data clustering: A cluster ensemble approach", in ICML, vol. 3, **(2003)**, pp. 186-193.
[19] S. V. Pons and J. R. Shulcloper, "A survey of clustering ensemble algorithms", International Journal of Pattern Recognition and Artificial Intelligence, vol. 25, no. 3, **(2011)**, pp. 337-372.
[20] A. Topchy, A. K. Jain and W. Punch, "Clustering ensembles: Models of consensus and weak partitions," IEEE Transactions on pattern analysis and machine intelligence, vol. 27, no. 12, **(2005)**, pp. 1866-1881.
[21] V. Filkov and S. Skiena, "Integrating microarray data by consensus clustering", International Journal on Artificial Intelligence Tools, vol. 13, no. 4, **(2004)**, pp. 863-880.
[22] M. Bertolacci and A. Wirth, "Are approximation algorithms for consensus clustering worthwhile", in SDM, **(2007)**, pp. 437-442.
[23] S. V. Pons, J. C. Morris and J. R. Shulcloper, "Weighted partition consensus via kernels", Pattern Recognition, vol. 43, no. 8, **(2010)**, pp. 2712-2724.
[24] T. Li, C. Ding and M. I. Jordan, "Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization", in Seventh IEEE International Conference on Data Mining, **(2007)**, pp. 577-582.
[25] A. P. Topchy, M. Law, A. K. Jain and A. L. Fred, "Analysis of consensus partition in cluster ensemble," in Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on, **(2004)**, pp. 225-232.
[26] W. S.Torgerson, "Multidimensional scaling: I. theory and method", Psychometrika, **(1952)**, pp. 401-419.
[27] Hotelling and Harold, "Analysis of a complex of statistical variables into principal components", Journal of educational psychology, vol. 24, no. 6, **(1933)**, pp. 417.

[28] D. Sankoff and J. B. Kruskal, "Time warps, string edits, and macromolecules: The theory and practice of sequence comparison", Canadian Journal of Statistics, **(2008)**, pp. 167-168.
[29] A. K. Jain and R. C. Dubes, "Algorithms for clustering data", Prentice-Hall, Inc., **(1988)**.
[30] C. Boutsidis, A. Zouzias and P. Drineas, "Random projections for k-means clustering", in Advances in Neural Information Processing Systems, **(2010)**, pp. 298-306.

Authors

Imran Khan is currently a Ph.D. candidate at the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He obtained MS degree from the National University of Computer & Emerging Sciences, Islamabad (Pakistan) in 2011. His research interests include data mining, analysis of complex data, and data warehouse and the business intelligent system.

Kamen Ivanov obtained B.Sc. degree from the Technical University of Sofia, Bulgaria in 2003. He is currently working towards a Ph.D. degree in Biomedical Electronics Engineering at the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences,

Shenzhen, China. His research interests include medical instrumentation and biomedical signal processing.

Qingshan Jiang is currently a professor at the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He received a Ph.D. in mathematics from Chiba Institute of Technology, Japan, in 1996; and a Ph.D. in computer science from University of Sherbrooke, Canada, in 2002. In 1999, he worked as a post-doc fellow at The Fields Institute for Research in Mathematical Sciences, University of Toronto, Canada. His research interests include Data mining, information security, pattern recognition, Massive data analysis, database technology.

