Knowledge-Based Systems 70 (2014) 293–300

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Short Communication

DSKmeans: A new kmeans-type approach to discriminative subspace clustering

Xiaohui Huang [a], Yunming Ye [a,*], Huifeng Guo [a], Yi Cai [b], Haijun Zhang [a], Yan Li [c]

[a] Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China
[b] School of Software Engineering, South China University of Technology, Guangzhou, China
[c] Shenzhen Polytechnic, Liuxian Road, Shenzhen 518055, China

ARTICLE INFO

Article history:
Received 23 February 2014
Received in revised form 15 June 2014
Accepted 15 July 2014
Available online 27 July 2014

Keywords:
Kmeans clustering
Feature selection
3-Order tensor
Data mining
Subspace clustering

ABSTRACT

Most of kmeans-type clustering algorithms rely on only intra-cluster compactness, i.e. the dispersions of a cluster. Inter-cluster separation which is widely used in classification algorithms, however, is rarely considered in a clustering process. In this paper, we present a new discriminative subspace kmeans-type clustering algorithm (DSKmeans), which integrates the intra-cluster compactness and the inter-cluster separation simultaneously. Different to traditional weighting kmeans-type algorithms, a 3-order tensor is constructed to evaluate the importance of different features in order to integrate the aforementioned two types of information. First, a new objective function for clustering is designed. To optimize the objective function, the corresponding updating rules for the algorithm are then derived analytically. The properties and performance of DSKmeans are investigated on several numerical and categorical data sets. Experimental results corroborate that our proposed algorithm outperforms the state-of-the-art kmeans-type clustering algorithms with respects to four metrics: Accuracy, RandIndex, Fscore and Normal Mutual Information(NMI).

1. Introduction

Clustering techniques have been used extensively in many fields in nature [1], such as bioinformatics [2], text organizations [3], and community detection [4], to name just a few. Clustering is an unsupervised classification technique that aims at partitioning a data set into clusters such that the objects within a cluster are similar and the objects in different clusters are dissimilar according to certain pre-defined criteria [5].

The clustering algorithms [6] can be summarized as partitioning methods, hierarchical methods, density-based methods, grid-based methods and model-based methods, etc. The kmeans-type clustering algorithm is a widely used partitioning methods in many real-life applications. Many researchers extended the kmeans algorithms by different types of weighting ways. From the weighting ways, existing kmeans-type algorithms can be classified into three categories: (1) No weighting kmeans-type algorithms [7–10], which treat all features equally in the process of minimizing the dispersions of clusters. Different features, however, have different discriminative capabilities in real-world applications. Therefore, different types of feature selection and weighting methods have been proposed in many clustering processes. (2) Vector weighting kmeans-type algorithms, which have been reported in [5,11–15]. (3) Matrix weighting kmeans-type algorithms, the examples of which are proposed in [16–21,3,22,23]. Most of these weighting kmeans-type clustering algorithms only consider that the objects in the same cluster are similar, i.e. minimizing the dispersions of all the clusters, in a way that the features are weighted by using different methods.

However, a feature in a cluster may have different discriminative capabilities when we compare this cluster with other clusters. For example, there are three clusters (C1, C2 and C3) in Fig. 1 (the distributions of features are listed in the table). W_{12} and W_{13} are weighting vectors when we compare cluster 1 (C1) with cluster 2 (C2) and cluster 3 (C3), respectively. We can observe that the features "Olympic, sport, chaos, riots" have more discriminative capabilities when comparing C1 with C2. In contrast, comparing C1–C3, the features "London, England, Beijing, China" have more discriminative capabilities. The same features in C1"London, England" have different discriminative capabilities when comparing to different clusters, i.e. "London, England" have less discriminative capabilities in distinguishing C1 and C2, while they have more discriminative capabilities in identifying C1 and C3.

Motivated by the example in Fig. 1, we propose a new kmeans-type algorithm by integrating the intra-cluster compactness and the inter-cluster separation with a 3-order tensor weighting

* Corresponding author.

http://dx.doi.org/10.1016/j.knosys.2014.07.009
0950-7051/

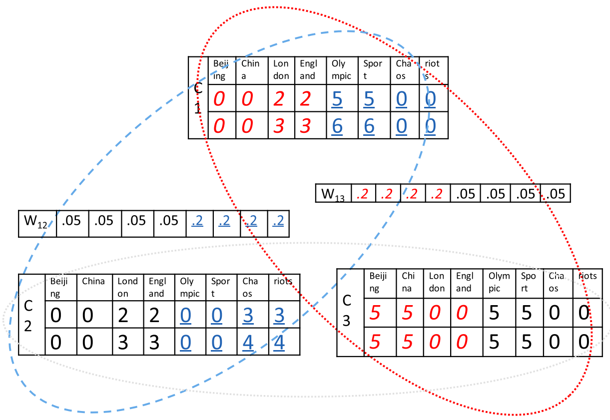

Fig. 1. An example of three clusters.

method. A row vector of each slice in the 3-order tensor represents the weights of a cluster comparing to another cluster. For example, the vector W_{12} in Fig. 1 denotes the weights of the features in C1 when comparing to C2. We then propose a new objective function. The corresponding iterative rules of the algorithm are derived in theory by optimizing the objective function. The experimental results show that the new algorithm performs better than other kmeans-type algorithms. The main contributions of this paper are twofold:

1. We propose a 3-order tensor weighting method to discriminate the weights of features when comparing every pair of clusters.
2. We propose kmeans-type clustering algorithm by using the 3-order tensor weighting method. An objective function of the algorithm is designed and then we give the updating rules of the algorithm.

The remaining sections of this paper are organized as follows: a brief overview of related works on various kmeans-type algorithms is presented in Section 2. Section 3 introduces our kmeans-type approach to discriminative subspace clustering. Experiments on both numerical and categorical data sets are presented in Section 4. We discuss the properties of our algorithm in Section 5 and conclude the paper in Section 6.

2. Related work

Kmeans-type algorithms have been studied extensively in many years. In this section, we give a brief survey of kmeans-type clustering from two aspects: the algorithms considering only the intra-cluster compactness and the algorithms integrating both the intra-cluster compactness and the inter-cluster separation.

2.1. Kmeans-type algorithms using only intra-cluster compactness

Most kmeans-type algorithms attempt to minimize the intra-cluster compactness to find a partition for a data set. Basic kmeans minimizes the sum of the squared distances between the empirical means of the clusters and the objects in the clusters and it equally treats all the features in the objects in the process of minimization. Many different ways have been used to extend basic kmeans. In k-medoids [24] and k-median [25], the means of the clusters in basic kmeans are replaced by the medoid (the most centrally located object) and the median, respectively. In order to solve the textual data clustering problem, many studies have used cosine metric instead of Euclidean distance, called spherical kmeans [26].

Shamir and Tishby [27] studied the behavior of clustering stability using kmeans clustering framework based on an explicit characterization of its asymptotic behavior. This paper concluded that kmeans-type algorithms do not "break down" in the large samples, in the sense that even when the sample size goes to infinity and the kmeans-type algorithms becomes stable for any choice of *K*. Since the clustering results of the kmeans-type algorithms are sensitive to the choice of initial centroids, many methods [28,29] are proposed to overcome this problem. Arthur and Vassilvitskii proposed kmeans++ [28] which chooses a initial centroid according to the distances with existing centroids. Another limitation of kmeans-type algorithms is to require manually tuning the parameter *K*(the number of clusters). For solving this problem, Pelleg et al. proposed *X*-means [30] which can automatically find the number of clusters by optimizing a criterion such as Bayesian Information Criterion.

The major problem of the kmeans-type algorithms mentioned above is to treat all features equally in the clustering process. In fact, the useful clusters in a data set usually occur in a subset of all the features [5]. To find this type of clustering structure, some researchers attempt to weight features with different methods [11,15,5,12]. These kmeans-type algorithms assigns a weight to a feature in the entire data set.

Subspace clustering algorithms are another types of weighting algorithms which seek to group objects into clusters in different subsets of features for different clusters. Subspace clustering algorithms assign a weight to a feature in each cluster. It pursues two goals simultaneously: finding a subsets of features for each cluster and partitioning the data set into different clusters from different subsets of features. In recent years, subspace clustering and feature weighting have been studied extensively [17,20,21,3,22,23]. Han et al. proposed attributes-weighting clustering algorithms (AWA) [21], which assign the bigger weights to the features which have smaller dispersions and the smaller weights to the features have larger dispersions for each cluster. Based on AWA [21], Jing et al. proposed an entropy weighting kmeans (EWkmean) [3], which minimizes the intra-cluster compactness and maximizes the negative weight entropy to stimulate more features contributing to the identification of a cluster. Based on EWkmeans [3], Ahmad and Dey [31] developed a kmeans-type clustering algorithm for subspace clustering of mixed numerical and categorical data sets. In a later study, Chen et al. [22] proposed a feature group weighting method for subspace clustering of high-dimensional data. To utilize a priori knowledge in the process of kmeans clustering, Pedrycz et al. [32] proposed a proximity-based fuzzy clustering which is able to fuse some constraints that specify an extent to which some pairs of objects are regarded similar or different.

2.2. Kmeans-type algorithms integrating both intra-cluster compactness and inter-cluster separation

In order to improve the clustering performance, some researchers introduced the information of inter-cluster separation to kmeans-type algorithms [13,14,33–35,23]. From the schemes of using the inter-cluster separation, these kmeans-type algorithms can be classified into two classes. (1) Calculating the distances between all pairs of objects which belong to different clusters as inter-cluster separation [13,14,33,34]. The hierarchical clustering method is used by Soete [13,14] to solve the feature selection problem. The hierarchical clustering method, however, requires high computational cost and cannot deal with large scale data set [5]. For overcoming the high computational cost, Makarenkov and Legendre [33] extended [13] to weight feature for kmeans clustering. Friedman and Meulman [34] developed the clustering objects on subsets of features algorithm for subspace clustering. (2) Calculating the distance between the centroid of each cluster and the global centroid as inter-cluster separation [35,23]. Wu et al. [35] introduced the inter-cluster separation to the fuzzy c-means model by calculating the distances between the centroids of the clusters and the global centroid. Inspired the method of [35], Deng et al. presented an enhanced soft subspace clustering(ESSC) [23] algorithm which is able to balance intra-cluster compactness and inter-cluster separation. Negative values, however, may be produced in the membership matrix if the balancing parameter is large. Moreover, ESSC has three manual input parameters which is difficult to find an appropriate group of parameters in real application.

2.3. Characteristic of our kmeans-type algorithm

Our proposed approach is related to the EWkmeans [3] which employs entropy to control the distribution of the weights. Different to EWkmeans that considers only the intra-cluster compactness, our approach introduces a 3-order tensor weight for synthesizing two clustering cues: the intra-cluster compactness and the inter-cluster separation.

We have also noticed that ESSC [23] has introduced the inter-cluster separation into a fuzzy weighting kmeans-type model. ESSC employs the matrix weighting that make each centroids of the clusters far away from the global centroid. However, ESSC cannot suggest the characteristic that a feature in a cluster has different discriminative capabilities while we compare this cluster to other clusters. In order to suggest this characteristic, we maximize the distance of each pair of clusters, instead of the traditional ways that maximize the distance with the global centroids. Thus, this paper proposes a 3-order tensor weighting kmeans-type algorithm, i.e. each cluster has $K-1$(K is the number of the clusters) weighting vectors. Each vector denotes a group weights of a cluster while we compare this cluster to other clusters.

3. Discriminative subspace Kmeans (DSKmeans) clustering model

In this section, we present a 3-order weighting kmeans-type approach to discriminative subspace clustering which is able to utilize intra-cluster compactness (i.e. the information of dispersions) and inter-cluster separation (i.e. the distances between different clusters) simultaneously. First, we develop a new objective function for the model. And then, the corresponding iterative rules are derived in theory. At last, we give the process of DSKmeans algorithm.

3.1. Optimization model

Let $X = \{X_1, X_2, \ldots, X_n\}$ be a set of n objects. Object $X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,m}\}$ is characterized by a set of m features (dimensions). The membership matrix U is a $n \times K$ binary matrix, where $\mathrm{u}_{\mathrm{i,p}} = 1$ indicates that object i is allocated to cluster p, otherwise, it is not allocated to cluster p. $Z = \{Z_1, Z_2, \ldots, Z_K\}$ is a set of K vectors representing the centroids of K clusters. The weight W is a 3-order tensor, each value $w_{p,q,j}$ in which denotes the importance of the feature j in cluster p while we compare cluster p to cluster q, where $p \neq q$. The objective function of DSKmeans can be formulated as

$$P(U,W,Z) = \sum_{p=1}^{K}\sum_{\substack{q=1\\q\neq p}}^{K}\sum_{j=1}^{m} w_{p,q,j} D_{p,q,j} + \gamma \sum_{p=1}^{K}\sum_{\substack{q=1\\q\neq p}}^{K}\sum_{j=1}^{m} w_{p,q,j}\log(w_{p,q,j}), \tag{1}$$

$$D_{p,q,j} = \sum_{i=1}^{n} u_{i,p}[(x_{i,j} - z_{p,j})^2 - \eta(z_{p,j} - z_{q,j})^2], \tag{2}$$

subject to

$$\begin{cases} \sum_{p=1}^{K} u_{i,p} = 1, u_{i,p} \in \{0,1\}, \\ \sum_{j=1}^{m} w_{p,q,j} = 1, 0 \leqslant w_{p,q,j} \leqslant 1, \end{cases} \tag{3}$$

where γ is a parameter that control the distribution of the weight and parameter η is used as balancing the effect of intra-cluster compactness and inter-cluster separation. In the objective function as shown in Eq. (1), the first term includes two parts: one is the sum of dispersions of all the clusters which involves in minimizing the intra-cluster compactness; the other is the sum of the distances between centroids of different clusters which involves in maximizing the inter-cluster separation. The second term is the sum of the negative weight entropy which is able to adjust the strength of the incentive for clustering on features. The effect of inter-cluster separation reduces with the decrease of η value. If $\eta = 0$, the effect of inter-cluster separation vanishes and the DSKmeans degenerates to EWkmeans. The different weighting vectors of a cluster when we compare this cluster to the other clusters are equal when $\eta = 0$.

3.2. DSKmeans clustering algorithm

In this subsection, we minimize the objective function P as shown in Eq. (1) with the constraints of Eq. (3) to obtain the iterative rules of the algorithm. The general method to optimization of P is to use the partial optimization for U, W and Z. In this method, we fix two constants of U, W, Z and minimize the objective function P with respect to the other constant. The following is three theorems to solve U, W and Z, respectively.

Theorem 1. *Let weights $\widehat{W}$ and centroids $\widehat{Z}$ be fixed, $P(U, \widehat{W}, \widehat{Z})$ is minimized iff*

$$\hat{u}_{i,p} = \begin{cases} 1, d_{i,p,} \leqslant d_{i,p'}, p \neq p', 1 \leqslant p' \leqslant K, \\ 0, otherwise, \end{cases} \tag{4}$$

where

$$d_{i,p} = \sum_{\substack{q=1\\q\neq p}}^{K}\sum_{j=1}^{m} \hat{w}_{p,q,j}[(x_{i,j} - \hat{z}_{p,j})^2 - \eta(\hat{z}_{p,j} - \hat{z}_{q,j})^2]. \tag{5}$$

The proof process of Theorem 1 can be found in [7,8].

Theorem 2. *Let membership matrix $\widehat{U}$ and centroids $\widehat{Z}$ be fixed, $P(\widehat{U}, W, \widehat{Z})$ is minimized iff*

$$\hat{w}_{p,q,j} = \exp\left(-\frac{D_{p,q,j}}{\gamma}\right) \Big/ \sum_{j'=1}^{m} \exp\left(-\frac{D_{p,q,j'}}{\gamma}\right), \tag{6}$$

where $D_{p,q,j}$ is given in (2).

Proof. We use the Lagrangian multiplier technique to obtain the following unconstrained minimization problem:

$$\Phi(W,\alpha) = \sum_{p=1}^{K}\sum_{\substack{q=1\\q\neq p}}^{K}\sum_{j=1}^{m} w_{p,q,j} D_{p,q,j} + \gamma \sum_{p=1}^{K}\sum_{\substack{q=1\\q\neq p}}^{K}\sum_{j=1}^{m} w_{p,q,j}\log(w_{p,q,j}) - \sum_{p=1}^{K}\sum_{\substack{q=1\\q\neq p}}^{K} \alpha_{p,q}\left(\sum_{j=1}^{m} w_{p,q,j} - 1\right), \tag{7}$$

where $\{\alpha_{p,q}\}$ is a matrix containing the lagrange multipliers corresponding to the constraints. If $\widehat{W}, \hat{\alpha}$ are the values of minimizing

$\Phi(W, \alpha)$, its gradient in the weight of features $w_{p,q,j}$ must vanish. Thus,

$$\frac{\partial \Phi(\widehat{W}, \hat{\alpha})}{\partial \hat{w}_{p,q,j}} = D_{p,q,j} + \gamma(1 + \log \hat{w}_{p,q,j}) - \hat{\alpha}_{p,q} = 0, \tag{8}$$

$$\frac{\partial \Phi(\widehat{W}, \alpha)}{\partial \hat{\alpha}_{p,q}} = \sum_{j=1}^{m} \hat{w}_{p,q,j} - 1 = 0. \tag{9}$$

From (8), we obtain

$$\hat{w}_{p,q,j} = \exp\left(\frac{-D_{p,q,j} - \gamma + \hat{\alpha}_{p,q}}{\gamma}\right) = \exp\left(\frac{\hat{\alpha}_{p,q} - \gamma}{\gamma}\right) \exp\left(\frac{-D_{p,q,j}}{\gamma}\right). \tag{10}$$

Substituting(10) into (9), we have

$$\exp\left(\frac{\alpha_{p,q} - \gamma}{\gamma}\right) \sum_{j=1}^{m} \exp\left(\frac{-D_{p,q,j}}{\gamma}\right) = 1. \tag{11}$$

It follows that

$$\exp\left(\frac{\alpha_{p,q} - \gamma}{\gamma}\right) = \frac{1}{\sum_{j=1}^{m} \exp\left(\frac{-D_{p,q,j}}{\gamma}\right)}. \tag{12}$$

Substituting (12) back to (10), we obtain the expression Eq. (6) of $w_{p,q,j}$. □

Theorem 3. *Let membership matrix $\widehat{U}$ and weights $\widehat{W}$ be fixed, $P(\widehat{U}, \widehat{W}, Z)$ is minimized iff centroids Z satisfies the following equations*

$$\left[(1-\eta)\sum_{\substack{q=1\\q\neq p}}^{K} w_{p,q,j} \sum_{i=1}^{n} u_{i,p} - \eta \sum_{\substack{q=1\\q\neq p}}^{K} w_{q,p,j} \sum_{i=1}^{n} u_{i,q}\right] z_{p,j} + \eta \sum_{\substack{q=1\\q\neq p}}^{K} \left(w_{p,q,j} \sum_{i=1}^{n} u_{i,p} + w_{q,p,j} \sum_{i=1}^{n} u_{i,q}\right) z_{q,j} = \sum_{\substack{q=1\\q\neq p}}^{K} w_{p,q,j} \sum_{i=1}^{n} u_{i,p} x_{i,j}. \tag{13}$$

Proof. Given $\widehat{U}$ and $\widehat{W}$, we minimize the objective function as shown in Eq. (1) with respect to $z_{p,j}$, the centroid of feature j in cluster p. We can compute the optimal value of $z_{p,j}$ by setting the gradient of $z_{p,j}$ to zero, i.e. $\frac{\partial P(\widehat{U}, \widehat{W}, Z)}{\partial z_{p,j}} = 0$. Then, we obtain

$$\frac{\partial P(\widehat{U}, \widehat{W}, Z)}{\partial z_{p,j}} = \left[(1-\eta)\sum_{\substack{q=1\\q\neq p}}^{K} w_{p,q,j} \sum_{i=1}^{n} u_{i,p} - \eta \sum_{\substack{q=1\\q\neq p}}^{K} w_{q,p,j} \sum_{i=1}^{n} u_{i,q}\right] z_{p,j} + \eta \sum_{\substack{q=1\\q\neq p}}^{K} \left(w_{p,q,j} \sum_{i=1}^{n} u_{i,p} + w_{q,p,j} \sum_{i=1}^{n} u_{i,q}\right) z_{q,j} - \sum_{\substack{q=1\\q\neq p}}^{K} w_{p,q,j} \sum_{i=1}^{n} u_{i,p} x_{i,j} = 0. \tag{14}$$

Rearranging the structure of Eq. (14), we can obtain Eq. (13). We have to solve this equations to gain the values of centroids Z.

However, solving this equations is time-consuming and this iterative rule cannot apply for categorical data set. For simplification and applying for the categorical data set, we choose the alternative way to compute the centroids, which is

$$z_{p,j} = \frac{\sum_{i=1}^{n} \hat{u}_{i,p} x_{i,j}}{\sum_{i=1}^{n} u_{i,p}}. \tag{15}$$

If the feature is categorical, $z_{p,j}$ is the mode of the feature value in cluster p [36].

In general, we set γ and η to positive real values. The DSKmeans algorithm is an extension of the matrix weighting of EWkmeans [3] to the 3-order weighting by considering the intra-cluster compactness and inter-cluster separation. The procedure of DSKmeans can be described as Algorithm 1. □

Table 1
Six numerical data sets and two categorical data sets.

DataSet	No. of features	No. of clusters	No. of objects
Iris	4	3	150
Glass	9	6	214
Ecoli	5	8	336
Robot2	2	4	5654
Robot4	4	4	5654
GeneCNS34	7129	2	34
Chess	37	2	3196
Molecular	61	2	1535

Algorithm 1. Discriminative subspace kmeans algorithm

Input: $X = \{X_1, X_2, \ldots, X_n\}, K$.
Output: U, Z, W.
Initialize: Randomly choose an initial $Z^0 = Z_1, Z_2, \ldots, Z_K$ and weights $\{w_{p,q,j}\}$.
repeat
  Fixed $\widehat{Z}, \widehat{W}$, solve the membership matrix U with (4);
  Fixed $\widehat{U}, \widehat{W}$ solve the centroids Z with (15);
  Fixed $\widehat{U}, \widehat{Z}$ solve the weight W with (6);
until Convergence.

4. Experiment

4.1. Experimental setup

In experiments, the performance of proposed approach is extensively evaluated on eight real-life data sets: six numerical data sets and two categorical data sets, reported in Machine Learning Repository.[1] The properties of these data sets are described in Table 1. The benchmark clustering algorithms – basic kmeans(Bkmeans), Bisecting kmeans(BSkmeans) [9], Wkmeans [5], AWA [21], EWkmeans [3] as well as ESSC [23] are chosen for the performance comparison with the proposed algorithm. Wherein, basic kmeans(Bkmeans), Bisecting kmeans(BSkmeans) [9] are no weighting kmeans-type clustering algorithms, Wkmeans [5] is vector kmeans-type clustering algorithm and AWA [21], EWkmeans [3] as well as ESSC [23] are matrix weighting kmeans-type clustering algorithms. Similar to our DSKmeans, ESSC [23] also considers both the intra-cluster compactness and the inter-clusters. For the Wkmeans, AWA, EWkmean and ESSC, we choose the optimal parameter values according to [5,21,3,23].

As we know that the result of the kmeans-type clustering is a local optimal solution. The clustering result relies on the initial centroids of clusters. In weighting kmeans-type clustering algorithms, the initial weights also influence the clustering result. Hence, to compare the performance between DSKmeans and the existing algorithms, we run all the compared algorithms 100 times by initializing all the algorithms with the same centroids at each time. Then, we calculate the average Accuracy, Rand Index, Fscore and Normal Mutual Information(NMI) of these results.

[1] http://archive.ics.uci.edu/ml/.

4.2. Performance metric

In this paper, to evaluate the performance of our proposed algorithm, we have used four performance metrics including Accuracy (Acc), RandIndex (RI), Fscore and Normal Mutual Information (NMI). In this subsection, we briefly introduce the computational process of these four metrics. Let C_q be the set of the class of data set(the annotated class) and C'_p be the set of the cluster generated by the clustering algorithm. We calculate the clustering accuracy as

$$Acc = \frac{\sum_{i=1}^{K} a_i}{n} \quad (16)$$

where a_p is the number of objects in C_p that are clustered to C'_p and n is the number of objects in the data set. Acc is the percentage of the objects that are correctly recovered in a clustering result.

Fscore combines the information of precision and recall which is extensively applied in evaluating the clustering result [3]. The precision and recall are calculated as

$$Precision\left(C'_p, C_q\right) = \frac{n_{p,q}}{|C'_p|}, \quad (17)$$

$$Recall\left(C'_p, C_q\right) = \frac{n_{p,q}}{|C_q|}, \quad (18)$$

where $n_{p,q}$ is the number of the objects of cluster C'_p in class C_q. The Fscore of the cluster C'_p and class C_q can be computed as

$$F\left(C'_p, C_q\right) = \frac{2 * P\left(C'_p, C_q\right) * R\left(C'_p, C_q\right)}{P\left(C'_p, C_q\right) + R\left(C'_p, C_q\right)}. \quad (19)$$

NMI is an increasingly popular measure of clustering quality [3], which can be formulated as

$$NMI = \frac{\sum_{p=1}^{K}\sum_{q=1}^{K} n_{p,q} \log\left(\frac{n \times n_{p,q}}{n_p \times n_q}\right)}{\sqrt{\left(\sum_{p=1}^{K} n_p \log\frac{n_p}{n}\right)\left(\sum_{q=1}^{K} n_q \log\frac{n_q}{n}\right)}}, \quad (20)$$

where n is the total number of objects, n_p, n_q and $n_{p,q}$ are the numbers of objects in clusters C'_p, class C_q and both the clusters C'_p and class C_q, respectively. NMI is a measure between 0 and 1. NMI equals 1 when two partitions are equivalent.

The Rand Index(RI) [5,23] is another metric using to evaluate the performance of the clustering algorithm, which is defined as follow

$$RI = \frac{a + b}{N(N-1)/2}, \quad (21)$$

where a is the number of pairs of data objects having different class labels and belonging to different clusters; b is the number of pairs of data objects having the same cluster labels and belonging to the same clusters; N is the size of the entire data set.

4.3. Numerical data set

4.3.1. Parametric study

Since EWkmeans and DSKmeans have a common parameter γ which is used to control the distribution of weight W, in experiments, we first search the best γ according to EWkmeans [3]. Then, we search best η based on the best value of γ gotten with algorithm EWkmeans. Thus, we can gain the local optimal combination of parameters γ and η. Fig. 2 shows the changing trends of the average results produced by DSKmeans after running 100 times for six numerical data sets in different values of η. We can observe that the results of DSKmeans increase with the increase of the value of η, and then, to certain value of η, the performance decreases with the increase of the value of η. When $\eta = 0$, the result of DSKmeans is equal to that of EWkmeans.

4.3.2. Results and analysis

In these experiments, we chose the best parameter η of balancing intra-cluster compactness and inter-cluster separation. We have also conducted an empirical study to demonstrate the effect of different settings of parameter η on the results (see Section 4.3.1). And the compared algorithms: Wkmeans [5] and AWA [21] have also a parameter β which is used to tune the weights of features. In experiments, we also search the best β for evaluating the performance of the algorithms. Since ESSC [23] is a fuzzy clustering algorithm, it has a fuzzy index parameter α except parameters γ and η. In experiments, we use the empirical value for α according the study of the algorithm ESSC [23]. The average clustering results after running 100 times of seven algorithms in six numerical data sets are shown in Tables 2–7. The values in brackets are the standard deviations of results produced by the running the algorithms 100 times. The bolds in the tables represent that the corresponding algorithm obtains the best result on the performance metric. From these results, we can observe that DSKmeans produces better results than the other six algorithms on all the data sets in overall. For data sets, Ecoli and Robot4, DSKmeans outperforms all other algorithms in all four metrics. In comparison to other algorithms, DSKmeans is able to deliver about 10% Fscore improvement in Ecoli and about 3% Acc improvement in Robot4. For the other four data sets, DSKmeans performs better

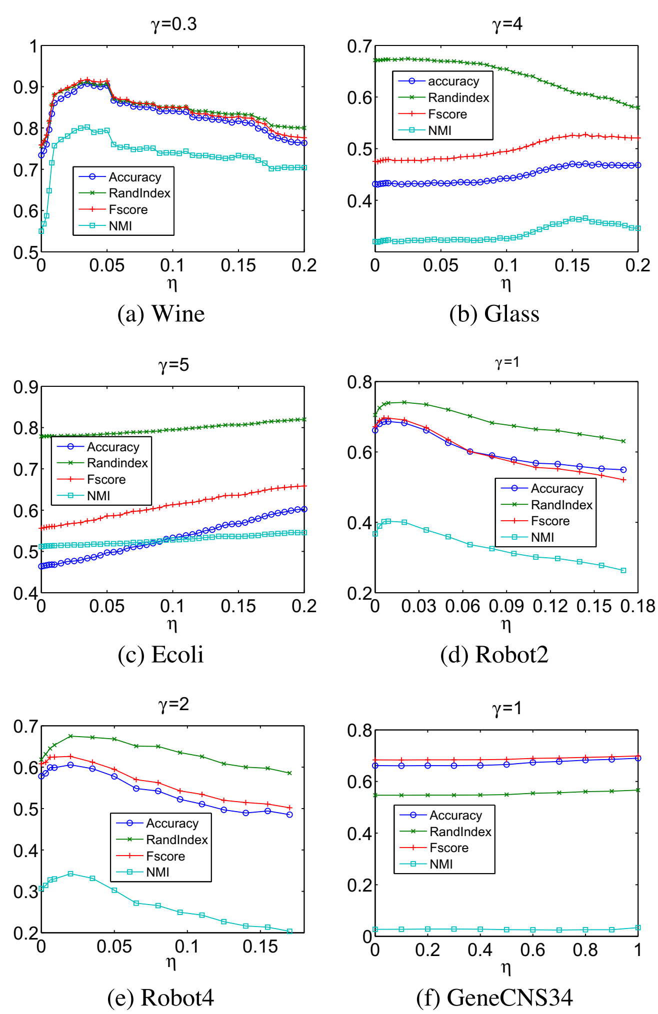

Fig. 2. The changing trends of acc, RI, fscore and NMI on numerical data sets.

Table 2
The results on Iris.

	Bkmeans	BSkmeans	Wkmeans	AWA	ESSC	EWkmeans	DSKmeans
Acc	0.7707(±0.1451)	0.8841(±0.0033)	0.8057(±0.1828)	0.8905(±0.1375)	0.8387(±0.1104)	0.7341(±0.1581)	**0.9073**(±0.1319)
RI	0.8145(±0.0715)	0.8715(±0.0028)	0.8555(±0.1066)	0.9050(±0.0762)	0.8497(±0.0554)	0.7559(±0.1291)	**0.9108**(±0.0989)
Fscore	0.8155(±0.0859)	0.8833(±0.0026)	0.8562(±0.1193)	0.9115(±0.0866)	0.8570(±0.0652)	0.7586(±0.1362)	**0.9171**(±0.1102)
NMI	0.6724(±0.0626)	0.7315(±0.0134)	0.7597(±0.1208)	**0.8170**(±0.0770)	0.7185(±0.0539)	0.5500(±0.2230)	0.8022(±0.1730)

The parameters: Wkmeans ($\beta = 7$); AWA ($\beta = 7$); ESSC ($\gamma = 0.3, \alpha = 1.2, \eta = 0.035$); EWkmeans ($\gamma = 0.3$); DSKmeans ($\gamma = 0.3, \eta = 0.035$).

Table 3
the results on Glass.

	Bkmeans	BSkmeans	Wkmeans	AWA	ESSC	EWkmeans	DSKmeans
Acc	0.4381(±0.0219)	0.4493(±0.0277)	0.4249(±0.0493)	0.4255(±0.0307)	0.4332(±0.0176)	0.4316(±0.0395)	**0.4683**(±0.0531)
RI	0.6786(±0.0122)	0.6635(±0.0172)	0.6743(±0.0319)	0.5403(±0.0539)	**0.6788**(±0.0095)	0.6714(±0.0347)	0.5959(±0.0380)
Fscore	0.4837(±0.0244)	0.4792(±0.0234)	0.4745(±0.0497)	0.4591(±0.0591)	0.4814(±0.0196)	0.4754(±0.0430)	**0.5237**(±0.0582)
NMI	0.3277(±0.0328)	0.3290(±0.0259)	0.3013(±0.0620)	0.2471(±0.0642)	0.3223(±0.0284)	0.3200(±0.0546)	**0.3548**(±0.0796)

The parameters: Wkmeans ($\beta = 7$); AWA ($\beta = 5$); ESSC ($\gamma = 4, \alpha = 1.2, \eta = 0.18$); EWkmeans ($\gamma = 4$); DSKmeans ($\gamma = 4, \eta = 0.18$).

Table 4
the results on Ecoli.

	Bkmeans	BSkmeans	Wkmeans	AWA	ESSC	EWkmeans	DSKmeans
Acc	0.4661(±0.0442)	0.4773(±0.0265)	0.4606(±0.0360)	0.4618(±0.0430)	0.4632(±0.0332)	0.4639(±0.0459)	**0.6023**(±0.0604)
RI	0.7800(±0.0186)	0.7786(±0.0083)	0.7785(±0.0165)	0.7781(±0.0179)	0.7811(±0.0109)	0.7785(±0.0200)	**0.8197**(±0.0311)
Fscore	0.5579(±0.0420)	0.5662(±0.0264)	0.5537(±0.0356)	0.5557(±0.0371)	0.5589(±0.0340)	0.5562(±0.0421)	**0.6585**(±0.0451)
NMI	0.5129(±0.0184)	0.4992(±0.0130)	0.5100(±0.0223)	0.4923(±0.0219)	0.5116(±0.0131)	0.5116(±0.0181)	**0.5456**(±0.0176)

The parameters: Wkmeans ($\beta = 7$); AWA ($\beta = 7$); ESSC ($\gamma = 3, \alpha = 1.2, \eta = 0.2$); EWkmeans ($\gamma = 5$); DSKmeans ($\gamma = 5, \eta = 0.2$).

Table 5
The results on Robot2.

	Bkmeans	BSkmeans	Wkmeans	AWA	ESSC	EWkmeans	DSKmeans
Acc	0.6457(±0.0000)	0.6330(±0.0143)	0.6300(±0.0151)	0.6335(±0.0570)	0.6035(±0.0892)	0.6615(±0.1051)	**0.6891**(±0.0871)
RI	0.7100(±0.0000)	0.7082(±0.0139)	0.7113(±0.0069)	0.7252(±0.0347)	0.6806(±0.0647)	0.7051(±0.0604)	**0.7411**(±0.0512)
Fscore	0.6630(±0.0000)	0.6380(±0.0162)	0.6348(±0.0234)	0.6302(±0.0759)	0.6316(±0.0750)	0.6718(±0.0873)	**0.6982**(±0.0702)
NMI	**0.4383**(±0.0000)	0.3894(±0.0192)	0.3890(±0.0408)	0.4048(±0.0685)	0.3480(±0.0627)	0.3673(±0.0690)	0.4045(±0.0559)

The parameters: Wkmeans ($\beta = 7$); AWA ($\beta = 7$); ESSC ($\gamma = 1, \alpha = 1.2, \eta = 0.01$); EWkmeans ($\gamma = 1$); DSKmeans ($\gamma = 1, \eta = 0.01$).

Table 6
The results on Robot4.

	Bkmeans	BSkmeans	Wkmeans	AWA	ESSC	EWkmeans	DSKmeans
Acc	0.4076(±0.0097)	0.4147(±0.0187)	0.4570(±0.0667)	0.5542(±0.0969)	0.5702(±0.1296)	0.5780(±0.1027)	**0.6056**(±0.0972)
RI	0.5821(±0.0064)	0.5878(±0.0158)	0.6175(±0.0456)	0.6694(±0.0588)	0.6335(±0.0930)	0.6183(±0.0831)	**0.6750**(±0.0680)
Fscore	0.4348(±0.0091)	0.4395(±0.0253)	0.4951(±0.0637)	0.5819(±0.0997)	0.6006(±0.1126)	0.6081(±0.0850)	**0.6260**(±0.0879)
NMI	0.1628(±0.0074)	0.1683(±0.0259)	0.2297(±0.0715)	0.3156(±0.1098)	0.3214(±0.1210)	0.3068(±0.0792)	**0.3426**(±0.0988)

The parameters: Wkmeans ($\beta = 7$); AWA ($\beta = 7$); ESSC ($\gamma = 2, \alpha = 1.2, \eta = 0.02$); EWkmeans ($\gamma = 1$); DSKmeans ($\gamma = 2, \eta = 0.02$).

Table 7
The results on GeneCNS34.

	Bkmeans	BSkmeans	Wkmeans	AWA	ESSC	EWkmeans	DSKmeans
Acc	0.6506(±0.0686)	0.6506(±0.0686)	0.6547(±0.0655)	0.6412(±0.0722)	0.5909(±0.0630)	0.6521(±0.0620)	**0.6900**(±0.0581)
RI	0.5412(±0.0413)	0.5412(±0.0413)	0.5429(±0.0404)	0.5365(±0.0406)	0.5100(±0.0293)	0.5403(±0.0379)	**0.5661**(±0.0363)
Fscore	0.6769(±0.0516)	0.6769(±0.0516)	0.6805(±0.0461)	0.6720(±0.0506)	0.6241(±0.0495)	0.6796(±0.0430)	**0.7005**(±0.0399)
NMI	0.0417(±0.0405)	0.0417(±0.0405)	0.0387(±0.0364)	0.0326(±0.0331)	**0.0451**(±0.0410)	0.0339(±0.0331)	0.0379(±0.0397)

The parameters: Wkmeans ($\beta = 7$); AWA ($\beta = 7$); ESSC ($\gamma = 1, \alpha = 1.2, \eta = 1$); EWkmeans ($\gamma = 1$); DSKmeans ($\gamma = 1, \eta = 1$).

than other algorithms in three metrics. In comparison to the existing algorithms, DSKmeans produces more than 4% Fscore improvement, 2% Acc improvement and 3% Acc improvement on data sets, Glass, Robot2 and GeneCNS34, respectively. We can also observe that no weighting algorithms: Basic kmeans and Bisecting kmeans have smaller standard deviations of results than that of weighting approaches: Wkmeans, AWA, EWkmeans, ESSC and DSKmeans in most of data sets.

4.3.3. Feature weighting

In our proposed algorithm, a feature in a cluster has $K - 1$ values of weights when we compare this cluster to the other $K - 1$

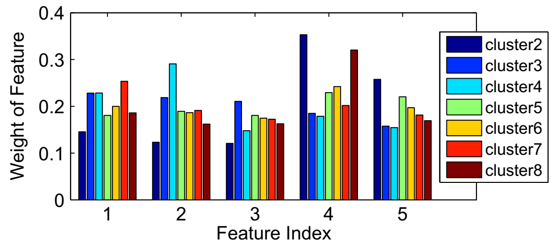

Fig. 3. The weights of the features in cluster 1 on data set Ecoli when cluster 1 compares to other seven clusters.

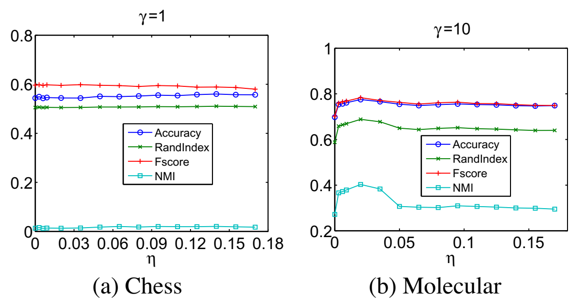

Fig. 4. The changing trends of acc, RI, fscore and NMI on categorical data sets.

clusters. These values represent the different discriminative capabilities of a feature when we compare this cluster to other $K - 1$ clusters. Fig. 3 is an example of feature weights of cluster 1 on data set Ecoli. There are five features and eight clusters on data set Ecoli. Fig. 3 shows the weights of features in cluster 1 while we compare cluster 1 to the other seven clusters. Each block represents the weights of a feature. We can observe that a feature in cluster 1 has different values of weights while we compare cluster 1 to different clusters. For example, feature 1 in cluster 1 has less discriminative capability when cluster 1 is compared to cluster 2, however, it has more discriminative capability when cluster 1 is compared to cluster 7. This result suggests that the inter-cluster separation is able to effect the distribution of weights and improve the clustering performance.

4.4. Categorical data set

4.4.1. Parametric study

In order to study the influence of the parameters to performance of algorithms, we also search the best γ for according to EWkmeans [3]. And then, we fix the parameter γ and run the Algorithm 100 times with different parameter η. Fig. 4 shows the changing trends of the average results produced by DSKmeans on two categorical data sets with different values of η. Similar to the numerical data sets, the results increase with the increment of the value of η at the beginning, and then, to the certain value of η, the results begin to reduce with the increase of the value of η.

4.4.2. Results and analysis

To further investigate the performance of the DSKmeans, we have evaluated DSKmeans in two categorical data sets. Since ESSC [23] does not give computational formulation to categorical data sets, we do not compare DSKmeans to ESSC in this two data sets. Similar to the numerical data sets, we also search the best parameters for all the algorithms. Tables 8 and 9 show the average clustering results of six algorithms in two categorical data sets. The results are also produced by running the Algorithms 100 times with different initial centroids. The values in brackets are the standard deviations. Likewise, the bolds in the tables represent that the corresponding algorithm obtains the best result on the performance metric. DSKmeans performs better than the other five algorithms in overall. Especially, the data set Molecular, DSKmeans obtains 7% Acc improvement, 10% RI improvement, 8% Fscore improvement and 13% NMI improvement comparing to the second best algorithms, EWkmeans. Since two data sets are only two clusters, the results of basic kmeans and bisecting kmeans are equal.

5. Discussion

From the results in Section 4, DSKmeans outperforms the compared kmeans-type algorithms: basic kmeans, bisecting kmeans, Wkmeans, AWA, EWkmeans and ESSC in terms of four evaluation measures: Acc, RI, Fscore and NMI in overall. These results suggest that the clustering performance can be improved by effectively utilizing the inter-cluster separation.

Comparing to traditional kmeans-type algorithms that utilize only the information of the intra-cluster, DSKmeans is able to integrate the intra-cluster and inter-cluster information simultaneously. We introduce a parameter η to balance the effect of two parts. Other algorithms are also able to employ intra-cluster compactness and inter-cluster separation, like ESSC [23]. Different to ESSC, DSKmeans uses a 3-order tensor to weight the features, which is more effective to express the characteristic that a feature in a cluster has different discriminative capabilities when we compare this cluster to the other clusters in real-world applications.

Table 8
The results on Chess.

	Bkmeans	BSkmeans	Wkmeans	AWA	EWkmeans	DSKmeans
Acc	0.5499($\pm$0.0344)	0.5499($\pm$0.0344)	0.5501($\pm$0.0347)	0.5270($\pm$0.0149)	0.5438($\pm$0.0266)	**0.5571**($\pm$0.0398)
RI	0.5072($\pm$0.0091)	0.5072($\pm$0.0091)	0.5073($\pm$0.0092)	0.5017($\pm$0.0024)	0.5051($\pm$0.0056)	**0.5095**($\pm$0.0133)
Fscore	0.5808($\pm$0.0288)	0.5808($\pm$0.0288)	0.5809($\pm$0.0289)	**0.6467**($\pm$0.0342)	0.5965($\pm$0.0436)	0.5812($\pm$0.0330)
NMI	0.0121($\pm$0.0151)	0.0121($\pm$0.0151)	0.0122($\pm$0.0151)	0.0084($\pm$0.0100)	0.0130($\pm$0.0149)	**0.0186**($\pm$0.0304)

The parameters: Wkmeans ($\beta = 7$); AWA ($\beta = 7$); EWkmeans ($\gamma = 1$); DSKmeans ($\gamma = 1, \eta = 0.2$).

Table 9
The results on Molecular.

	Bkmeans	BSkmeans	Wkmeans	AWA	EWkmeans	DSKmeans
Acc	0.5440($\pm$0.0292)	0.5440($\pm$0.0292)	0.5431($\pm$0.0289)	0.6148($\pm$0.0539)	0.7102($\pm$0.0653)	**0.7867**($\pm$0.1292)
RI	0.5052($\pm$0.0070)	0.5052($\pm$0.0070)	0.5050($\pm$0.0072)	0.5318($\pm$0.0358)	0.5966($\pm$0.0441)	**0.6973**($\pm$0.1436)
Fscore	0.5466($\pm$0.0266)	0.5466($\pm$0.0266)	0.5453($\pm$0.0269)	0.6562($\pm$0.0280)	0.7090($\pm$0.0356)	**0.7899**($\pm$0.1181)
NMI	0.0084($\pm$0.0104)	0.0084($\pm$0.0104)	0.0079($\pm$0.0106)	0.1631($\pm$0.0646)	0.2876($\pm$0.0838)	**0.4187**($\pm$0.1961)

The parameters: Wkmeans ($\beta = 7$); AWA ($\beta = 7$); EWkmeans ($\gamma = 10$); DSKmeans ($\gamma = 10, \eta = 0.2$).

6. Conclusion

In this paper, we have presented a new kmeans-type algorithm by integrating intra-cluster compactness and inter-cluster separation with a 3-order tensor weighting. This work involves the following aspects: (1) A new objective function is proposed; (2) The corresponding updating rules are derived by optimizing the objective function; and (3) Extensive experiments are conducted to evaluate the performance of the new algorithm based on four evaluation metrics: Acc, RI, Fscore and NMI. The results demonstrate that the extending algorithm is more effective than the state-of-the-art algorithms.

Acknowledgements

The authors are very grateful to the editors and anonymous referees for their helpful comments. This research was supported in part by NSFC of China under Grant Nos. 61272538 and 61300209, Shenzhen Strategic Emerging Industries Program under Grants No. JCYJ20130329142551746. Y. Li's research supported in part by NSFC of China under Grant No. 61303103, and Shenzhen Science and Technology Program under Grant No. JCY20130331150354073. H.J. Zhang's work was supported in part by the NSFC of China under Grant No. 61300209, the Shenzhen Foundation Research Fund under Grant No. JCY20120613115205826 and the Shenzhen Technology Innovation Program under Grant No. CXZZ20130319100919673.

References

[1] M. Anderberg, Cluster Analysis for Applications, Tech. rep., Office of the Assistant for Study Support Kirtland AFB N MEX, 1973.
[2] G. Moreno-Hagelsieb, Z. Wang, S. Walsh, A. ElSherbiny, Phylogenomic clustering for selecting non-redundant genomes for comparative genomics, Bioinformatics 29 (7) (2013) 947–949.
[3] L. Jing, M. Ng, J. Huang, An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data, IEEE Trans. Knowl. Data Eng. 19 (8) (2007) 1026–1041.
[4] L. Tang, H. Liu, J. Zhang, Identifying evolving groups in dynamic multi-mode networks, IEEE Trans. Knowl. Data Eng.
[5] J. Huang, M. Ng, H. Rong, Z. Li, Automated variable weighting in k-means type clustering, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 657–668.
[6] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, Morgan Kaufman, 2011.
[7] J. Bezdek, A convergence theorem for the fuzzy isodata clustering algorithms, IEEE Trans. Pattern Anal. Mach. Intell. (1) (1980) 1–8.
[8] S. Selim, M. Ismail, K-means-type algorithms: a generalized convergence theorem and characterization of local optimality, IEEE Trans. Pattern Anal. Mach. Intell. (1) (1984) 81–87.
[9] M. Steinbach, G. Karypis, V. Kumar, et al., A comparison of document clustering techniques, in: KDD workshop on text mining, vol. 400, 2000, pp. 525–526.
[10] P. Bradley, U. Fayyad, C. Reina, Scaling clustering algorithms to large databases, in Proceedings of the 4th International Conference on Knowledge Discovery & Data Mining, 1998, pp. 9–15.
[11] W. DeSarbo, J. Carroll, L. Clark, P. Green, Synthesized clustering: a method for amalgamating alternative clustering bases with differential weighting of variables, Psychometrika 49 (1) (1984) 57–78.
[12] X. Chen, X. Xu, J. Huang, Y. Ye, Tw-k-means: automated two-level variable weighting clustering algorithm for multi-view data, IEEE Trans. Knowl. Data Eng. 24 (4) (2013) 932–944.
[13] G. Soete, Optimal variable weighting for ultrametric and additive tree clustering, Qual. Quant. 20 (2) (1986) 169–180.
[14] G. De Soete, Ovwtre: a program for optimal variable weighting for ultrametric and additive tree fitting, J. Classif. 5 (1) (1988) 101–104.
[15] D. Modha, W. Spangler, Feature weighting in k-means clustering, Mach. Learn. 52 (3) (2003) 217–237.
[16] C. Aggarwal, J. Wolf, P. Yu, C. Procopiuc, J. Park, Fast algorithms for projected clustering, ACM SIGMOD Record 28 (2) (1999) 61–72.
[17] L. Parsons, E. Haque, H. Liu, Subspace clustering for high dimensional data: a review, ACM SIGKDD Explorations Newsletter 6 (1) (2004) 90–105.
[18] C. Domeniconi, D. Papadopoulos, D. Gunopulos, S. Ma, Subspace clustering of high dimensional data, in: Proceedings of the SIAM International Conference on Data Mining, 2004, pp. 517–521.
[19] M. Al-Razgan, C. Domeniconi, Weighted clustering ensembles, in: Proceedings of SIAM International Conference on Data Mining, 2006, pp. 258–269.
[20] H. Frigui, O. Nasraoui, Simultaneous clustering and dynamic keyword weighting for text documents, Survey of Text Mining (2004) 45–70.
[21] E. Chan, W. Ching, M. Ng, J. Huang, An optimization algorithm for clustering using weighted dissimilarity measures, Pattern Recogn. 37 (5) (2004) 943–952.
[22] X. Chen, Y. Ye, X. Xu, J. Zhexue Huang, A feature group weighting method for subspace clustering of high-dimensional data, Pattern Recogn. (45) (2012) 434–446.
[23] Z. Deng, K. Choi, F. Chung, S. Wang, Enhanced soft subspace clustering integrating within-cluster and between-cluster information, Pattern Recogn. 43 (3) (2010) 767–781.
[24] L. Kaufman, P. Rousseeuw, et al., Finding Groups in Data: An Introduction to Cluster Analysis, vol. 39, Wiley Online Library, 1990.
[25] K. Chen, On k-median clustering in high dimensions, in: Proceedings of the seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm, pp. 1177–1185.
[26] I. Dhillon, D. Modha, Concept decompositions for large sparse text data using clustering, Mach. Learn. 42 (1) (2001) 143–175.
[27] O. Shamir, N. Tishby, Stability and model selection in k-means clustering, Mach. Learn. 80 (2) (2010) 213–243.
[28] D. Arthur, S. Vassilvitskii, k-Means++: the advantages of careful seeding, in: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.
[29] S. Shahnewaz, M. Rahman, H. Mahmud, A self acting initial seed selection algorithm for k-means clustering based on convex-hull, Inf. Eng. Inf. Sci. (2011) 641–650.
[30] D. Pelleg, A. Moore, X-means: extending k-means with efficient estimation of the number of clusters, in: Proceedings of the Seventeenth International Conference on Machine Learning, San Francisco, 2000, pp. 727–734.
[31] A. Ahmad, L. Dey, A k-means type clustering algorithm for subspace clustering of mixed numeric and categorical datasets, Pattern Recogn. Lett. 32 (7) (2011) 1062–1069.
[32] W. Pedrycz, V. Loia, S. Senatore, P-fcm: a proximity-based fuzzy clustering, Fuzzy Sets Syst. 148 (1) (2004) 21–41.
[33] V. Makarenkov, P. Legendre, Optimal variable weighting for ultrametric and additive trees and k-means partitioning: methods and software, J. Classif. 18 (2) (2001) 245–271.
[34] J. Friedman, J. Meulman, Clustering objects on subsets of attributes (with discussion), J. R. Stat. Soc.: Ser. B (Statist. Meth.) 66 (4) (2004) 815–849.
[35] K. Wu, J. Yu, M. Yang, A novel fuzzy clustering algorithm based on a fuzzy scatter matrix with optimality tests, Pattern Recognit. Lett. 26 (5) (2005) 639–652.
[36] Z. Huang, Extensions to the k-means algorithm for clustering large data sets with categorical values, Data Mining Knowl. Discovery 2 (3) (1998) 283–304.