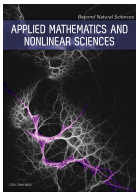
sciendo Applied Mathematics and Nonlinear Sciences 5(1) (2020) 1–10



Applied Mathematics and Nonlinear Sciences

https://www.sciendo.com

Improvement of the Fast Clustering Algorithm Improved by K-Means in the Big Data

Ting Xie[1]†, Ruihua Liu[2], Zhengyuan Wei[1]

[1]College of Science, Chongqing University of Technology, Chongqing 400054, China

[2]College of Artificial Intelligence, Chongqing University of Technology, Chongqing 400054, China

Submission Info

Communicated by Juan Luis García Guirao
Received September 23rd 2019
Accepted December 26th 2019
Available online January 20th 2020

Abstract

Clustering as a fundamental unsupervised learning is considered an important method of data analysis, and K-means is demonstrably the most popular clustering algorithm. In this paper, we consider clustering on feature space to solve the low efficiency caused in the Big Data clustering by K-means. Different from the traditional methods, the algorithm guaranteed the consistency of the clustering accuracy before and after descending dimension, accelerated K-means when the clustering centeres and distance functions satisfy certain conditions, completely matched in the preprocessing step and clustering step, and improved the efficiency and accuracy. Experimental results have demonstrated the effectiveness of the proposed algorithm.

Keywords: Big Data; Clustering; K-means; Feature space.
AMS 2010 codes: 62K86.

1 Introduction

The data Data is are quantizsed symbol of the information. Data clustering is a process to find the effective information and hidden structure feature based on data collection and reasonable division by a similarity measure, which is an important data mining technique for unsupervised learning and have an is important and widely used in pattern recognition [1–3], machine learning [4,5], image processing [6,7] and other fields. In the era of Big Data, a great deal of valuable data information is produced at all times with the rapid development of economy, science and technology. Different from traditional data, Big Data usually is sparse and has multi noise, , high dimension, sparse, heterogeneous feature fusion and so on [8–10]. How to construct efficient clustering models and algorithms for Big Data is a very important and challenging research topic, and has important scientific value and economic benefits.

†Corresponding author.
Email address: xieting@cqut.edu.cn

sciendo
ISSN 2444-8656 doi:10.2478/AMNS.2020.1.00001

Data clustering, as an important data mining technology, aims to divide data objects into several different clusters according to similarity measure, so that data objects within clusters have the greatest similarity and data objects between different clusters have the smallest similarity. A variety of data clustering algorithms have been researched in past years, which include nonnegative matrix factorizsation [11–14], mean shift [15–17], spectral clustering [18–20], sparse subspace clustering [21–23], and K-means [24–27], etc. Undoubtedly, K-means is the most commonly used and important clustering algorithm. The purpose of K-means clustering purpose is to minimizse the sum of squared Euclidean distance between each data points and its closest centere point [25]:

$$\min_{c_{j,j}} \sum_{j=1}^{k} \sum_{i \in \mathbb{C}_j} dist(m_i - c_j) \tag{1}$$

where $dist$ is the distance function, k is the clustering number, c_j is the clustering centere, $\mathbb{C}_j$ is the index set of the j-th cluster and $\cup \mathbb{C}_{j=1j}^{k} = \{1, 2, \cdots, n\}$. In fact, this process is similar to Voronoi diagram of function $dist$ in data space, so locating the global optimum solution of this problem is NP-hard even for the simplest case. Therefore, only the local minimum of K-means model is applied in practice. The most commonly used algorithm is Lloyd's algorithm [25]. Lloyd's algorithm is a heuristic iterative refinement method, which can quickly converge to the local optimum solution. For the pre-selected cluster centeres, the algorithm mainly carries out the following iterative processes: allocation steps and update steps. The purpose of the allocation step is to allocate the data points to each data centere by calculating the distance from the checkpoint to each clustering centere,; the update steps is to involves recalculateion of the new clustering centeres through the current allocation. From the Lloyd's algorithm, we can find that different clustering models and corresponding algorithms can be obtained by defining different similarity measure $dist$, selecting different initial clustering centeres and defining different recalculating clustering centeres. It is precisely because K-means model and Lloyd's algorithm have such flexible forms that they can choose suitable forms according to different problems, so they have been widely used. However, regardless of the model and algorithm, the disaster of data scale and dimension is unavoidable. We know that the most time-consuming part of the Lloyd's algorithm is the allocation step. The calculating amount of the distance between the point and centere (e.g. Euclidean distance) is $\mathrm{O}(kdn)$, and the total amount of calculation of Lloyd's algorithm is $\mathrm{O}(tkdn)$, where t is the iteration step. Therefore, when d and n is are large, the efficientcy of Lloyd's algorithm will be greatly reduced [28–30].

In the process of clustering, the basic K-means model needs to determine three parameters the following: the selection of initial iteration point c_0 and the iterative process;, the determination or estimation of clustering number k; and the definition of similarity measure $dist$ between sample points. Many improved K-means models and algorithms can be obtained by choosing different processing methods offor the above-mentioned three parameters. Several improved initializsation schemes were proposed to deal with the initializsation issue of Lloyd's algorithm. For example, one of the most important improved models is the K-means++++ [28]. This is assuming that $\overline{k}(0 < \overline{k} < k)$ initial clustering centeres have been selected,; the farther away from the current $\overline{k}$ clustering centeres, the higher the probability of selecting the $\overline{k}+1$ clustering centere. For more detail on initializsation methods for K-means, we refer to Celebi et al. the Reference [30]. According to the different selection rules of clustering number k, the improved K-means model generally includes: K-J inflection point model [31] and ISODATA [32] (Iterative Self-Organizing Data Analysis Techniques Algorithm (ISODATA) [32]. The K-J model calculates a series of cluster number k and cluster objective function value J, draws the relationship between them and uses the inflection point of the graph to determine the value of k. The ISODATA repeatedly modifies the number of clustering centeres to get a more reasonable number of categories k in the process of iteration. When given parameters, this modification is achieved by merging and splitting, we can (refer to the Reference [33]). In order to deal with data clustering problems with different data distributions, it is necessary to select different similarity measures between sample points, so that different types of K-means improved models can be obtained. Kernel K-means model is similar to the idea of kernel function in support vector machine, mapping all samples to another vector space and clustering [31]. Fuzzy C-means model introduces the fuzzy factors, calculates the membership degree from each point to each cluster and uses this membership degree to

determine the level to which each point belongs to a cluster [34]. Spherical K-means model is to solve the problem clustering data points on a sphere [35]. When using this model, we need to normalizse each data point, and then use cosine dissimilarity instead of Euclidean distance to measure the similarity between data points. In addition, there are some improved models that used Mahalanobis distance, Itakura–Saito divergence and Bergman divergence instead of Euclidean distance as a similarity measure. For sample points $x \in R^{m\times n}$, and $y \in R^{m\times n}$, S is covariance, and Φ is convex function, there are we have [36–38]:

Mahalanobis distance:

$$d(x,y) = \sqrt{(x-y)^T S^{-1}(x-y)} \tag{2}$$

Itakura–Saito divergence:

$$d(x,y) = \frac{x}{y} - \ln(\frac{x}{y}) - 1 \tag{3}$$

Bergman divergence:

$$d(x,y) = \Phi(x) - \Phi(y) - \langle \nabla\Phi(y), (x-y)\rangle \tag{4}$$

In Big Data clustering problems, the advantages of K-means and related improved algorithms are mainly focused on include the following: concise and intuitive, high computing efficiency and scalability. Comparatively, it also has obvious shortcomings: the distribution of clustering data is too strict;, different initial points lead to distinct clustering results and easily fall into local optimal solution; and computational complexity is linearly correlated with data dimension. Therefore, when dealing with the low-dimensional data clustering problems, K-means and related improved algorithms usually get more accurate results and the running time is acceptable. However, for high-dimensional Big Data, due to the impact of dimension disaster, data distribution, data size, data noise and so on, the K-means and related improved algorithms often cannot get the desired results and the computational efficiency is low. In order to deal with this problem, a common method is to reduce the dimensionality of data, that is, to seek low-dimensional features of high-dimensional data, and then to cluster within low-dimensional features. However, there are some difficulties problems in high-dimensional data clustering algorithm based on dimensionality reduction. Firstly, is low-dimensional data the dominating feature needed in clustering of practical problems? Secondly, whether the mapping of distances between low-dimensional data points is conductive clustering?. Generally speaking, data dimensionality reduction and low-dimensional clustering are two important partsteps in solving large data clustering process: data dimensionality reduction clears obstacles for low-dimensional clustering (removing data noise, reducing data dimension, etc.), and low-dimensional clustering achieves the ultimate goal of clustering. In order to achieve excellent processing results, the two processes of data dimensionality reduction and low-dimensional clustering should complement and match each other in Big Data clustering.

In this paper, unlike the existing improved K-means models, aiming at the low efficiency of traditional algorithms caused by Big Data, we purpose clustering in feature space to improve the efficiency of the algorithm while ensuring the accuracy. We point out that as long as the clustering centere and distance function satisfy certain conditions, most K-means algorithms can be accelerated with our ideas. In addition, we proved that the processing steps (data dimension reduction) and clustering steps (low-dimensional clustering) of the proposed method are completely matched in the problem of Dig Data clustering.

2 *K*-means Fast Algorithms for Big Data Clustering

In the classical K-means model, the mean of data points is usually chosen as the clustering centere and the initial value is chosen randomly. In the problems of Big Data clustering, through continuous theoretical research and practical application, researchers found that European distance is extremely unfavourable to measure the

similarity between high-dimensional data, especially sparse high-dimensional data. In addition, the selection method of random cluster centeres is also very sensitive to noise. The improved models and algorithms are mainly aimed at the above-mentioned two points. Next, we will introduce two famous improved models in practical application forms in Big Data clustering problems: spherical K-means model and K-medoids model.

For Big Data clustering, the original data are heterogeneous, noisy, high- dimensional and sparse. The directional differences between the original data are far more important than the metric measurement differences, because the length of the original data is likely to be different, even though the differences between the same measurements should be significant. Based on this, in 2011, Dhillon et al. proposed the use of cosine dissimilarity to measure the distance between data [35], that is spherical K-means model. The definition of cosine difference of two points $x \in R^{d\times n}$ and $y \in R^{d\times n}$ is as follows:

$$d(x,y) = 1 - \cos(x,y) = 1 - \frac{\langle x,y\rangle}{\|x\|\,\|y\|} \tag{5}$$

It can be obtained that the objective function of spherical K-means model has the following form:

$$\sum_{j=1}^{k}\sum_{i=1}^{n} u_{ij}\left(1 - \frac{\langle m_i, c_j\rangle}{\|m_i\|\,\|c_j\|}\right) = \sum_{j=1}^{k}\left[\sum_{i=1}^{n} u_{ij} - \left\langle \sum_{i=1}^{n} u_{ij}\frac{m_i}{\|m_i\|}, \frac{c_j}{\|c_j\|}\right\rangle\right] \tag{6}$$

Using Cauchy–Schwartz inequality, spherical K-means model can get the optimal value if and only if:

$$c_j \leftarrow \sum_{i=1}^{n} u_{ij}\frac{m_i}{\|m_i\|} \;\; \forall j \tag{7}$$

If the data isare normalizsed, i.e. $\|m_i\| = 1$, the clustering centere of spherical K-means model can be expressed as:

$$c_j \leftarrow \sum_{i=1}^{n} u_{ij}\frac{m_i}{\|m_i\|} = \sum_{i\in_j} m_i \;\; \forall j \tag{8}$$

that is, the clustering centere c_j is proportional to the sum of all the data in the j class., where$m_i \in M, M \in R^{d\times n}$ is the original data, k is the clustering number, c_j is the clustering centere, $_j$ is the index set of the j-th cluster, and $u_{ij} = \{0,1\}$ is the indicator function.

For spherical K-means model, Lloyd iteration algorithm cannot increase the objective function, but and cannot guarantee certain convergence. The algorithm process is shown in below Figure 1.

Algorithm 1: The algorithm of spherical K-means model

Input: $M \in R^{d\times n}, k \in Z_+$

Given the initial centeres of $k$cluster:
$C^{(0)} = \left\{c_1^{(0)}, c_2^{(0)}, \cdots, c_k^{(0)}\right\}, c_i^T c_i = 1$

while the number of iterations is less than N **do**

update the following iteration format

$$u_{ij}^{(l+1)} = \begin{cases} 1 \; i = \arg\max\limits_{K}\{m_j^T c_i^{(l)}\} \\ 0 \; else \end{cases}$$
$$c_j = \sum_{i=1}^{n} u_{ij}^{(l+1)} m_i$$

end while

Output: Clustering results.

Figure 1. The algorithm of the spherical K-means model.

In Big Data clustering, there are always a lot of outliers due to the influence of practical problem. Generally speaking, the accuracy of K-means will be greatly reduced when the data hasve outliers. The main reason is that the clustering centere will be seriously affected by outliers. In order to solve this problem, Kaufman and Rousseeuw proposed using medoids, a point in the centere of data (location in the centere), as the clustering centere. Therefore, this method is also called K-medoids model. The algorithm process is shown in below Figure 2.

Algorithm 2: The algorithm of K-medoids model

Input: $M \in R^{d\times n}, k \in Z_+$

Random selection of k values from Mas the initial central point

while no change in centere position **do**

each point is divided into clusters with the nearest central location points;

update the central location points of each cluster;

calculate the value of the objective function;

parameters selection.

end while

Output: Clustering results.

Figure 2. The algorithm of the K-medoids model.

Medoid minimizses the average difference between all data points in the same cluster, so it has high robustness to noise. But However, the calculation of medoid needs to compare the distance of all data points before deciding which one is medoid, which is much more complex than mean, resulting in the low computational efficiency of K-medoids model for high-dimensional data. It has good robustness but low efficiency in the application of Big Data clustering. Partitioning Around Medoids PAM (PAMPartitioning Around Medoids) is the most effective algorithm for solving K-medoids model. Given a randomly selected initial value, PAM replaces each cluster centere with a data point that reduces the objective function. This process continues to iterate until each medoids cannot be replaced. In each iteration, the computational complexity of PAM is $O(d(n-k)^2k)$. In order to reduce the computational complexity, Park and Jun proposed a fast K-medoids clustering algorithm. This method calculates the distance matrix iteratively and uses it to calculate the mew medoids. The computational complexity of this algorithm is consistent with K-means, which makes this algorithm widely concerned applicable in Big Data clustering. However, when the data scale increases sharply, the calculation of the preprocessing matrix is too large.

3 The Improved Algorithms of K-means on Feature Space

To reasonably utilise In order to make the concise and efficient K-means algorithm reasonably utilized in Big Data clustering, we will consider the improved K-means algorithm in dimension reduction space (feature space). Assuming that the rank of the data matrix $M \in R^{d\times n}$ is $r = Rank(M) \leq \min(d,n)$, and M is decomposed into $M = U\Sigma V^T$ by singular value decomposition, then:

$$U^T M = \Sigma V^T = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} [\underbrace{V_1}_{r} \underbrace{V_2}_{n-r}]^T = \begin{bmatrix} \Sigma_r V_1^T \\ 0 \end{bmatrix} \triangleq \begin{bmatrix} \hat{M} \\ 0 \end{bmatrix} \tag{9}$$

where V_1 contains the first r right singular vectors of data M. We have the following theorem about the problem of original problem and feature space, that is, the original problem is completely equivalent to the problem of feature space under certain conditions.

Theorem 1. *If the K-means problem and its extended model satisfy the following two conditions,:*
(1) cluster centere c_j is the linear combination of all data points. and

(2) distance function dist is orthogonal invariant.,

then the K-means problem in the original data space M is equivalent to the K-means problem in the feature space $\widehat{M}$. Thus, the computational complexity of K-means clustering problem is reduced from $\mathrm{O}(ikdn)$ *to* $\mathrm{O}(ikrn)$.

Proof. Without losing generality, it can be assumed that the cluster centere of the original space M is $c_j = \sum_j w_{lj} m_l$, and $\widehat{c}_j = \sum_j w_{lj} \widehat{m}_l$ is defined according to its weight w, where $\widehat{m}$ is the point in the feature space $\widehat{M}$. According to the defineition of M, the cluster centere is transformed into:

$$U^T c_j = U^T \left(\sum_l w_{lj} m_l \right) = \sum_l w_{lj} (U^T m_l) = \sum_l w_{lj} \begin{bmatrix} \widehat{m}_l \\ 0 \end{bmatrix} = \begin{bmatrix} \widehat{c}_j \\ 0 \end{bmatrix} \tag{10}$$

From the orthogonal invariance of the distance function *dist*, the following equation holds:

$$dist(m_i, c_j) = dist(U^T m_i, U^T c_j) = dist(\widehat{m}_l, \widehat{c}_j) \tag{11}$$

Therefore, the objective function of the K-means problem in the original space M is the same as that in the feature space $\widehat{M}$, and the selection method of clustering centere is the same.

Theorem 2. *For the standard K-means problem, the Spherical spherical K-means problem and some K-medoids problems, the clustering in the original space is consistent with that in the feature space.*

Proof. First of all, the clustering centere of the standard K-means problem and spherical K-means problem are the average of all data points, while the clustering centere of the K-medoids problem are some data points, so the first condition in Theorem 1 is satisfied.

Then, the Euclidean distance and cosine difference satisfy the orthogonal condition. Therefore, as long as the Euclidean distance and cosine difference are chosen as the distance function, the K-medoids problem satisfies the second condition of Theorem 1.

According to Theorem 1, this conclusion can be obtained.

Based on the above discussion, when using K-means to deal with Big Data clustering problem, we can first reduce the dimension of the data, and then carry out clustering analysis in the feature space. The algorithm process is shown in below Figure 3.

Algorithm 3: *K*-means for high-dimensional data
Input: $M \in R^{d \times n}, k \in Z_+$
Pre-processing Step: $U_1^T M = \sum_r V_1^T \overset{\Delta}{=} {}^{r \times n}$
Clustering Step: Run Lloyd algorithm on $\widehat{M}$
Output: Clustering results.

Figure 3. The K-means clustering algorithm of for high-dimensional data.

4 Numerical Experiment

In this section, we use artificial data and actual data to test the performance of the algorithm, mainly verifying two aspects: whether the objective function is consistent, whether the running time is reduced. All numerical experiments were run on a desktop computer with an Intel Core i7-3770 CPU at 3.40 GHz with 8 GB RAM under Matlab R2017b.

We construct the following artificial data to verify the accuracy of the algorithm. Firstly, k points in d dimensional space are randomly selected as clustering centeres. Then, Gaussian points with variance of σ are added around the clustering centeres, and the total number of data points is n. Finally, n data points are randomly replaced. In order to test the influence of data dimension, we make d vary from 1,000 to 50,000. The number of samples is n = 1,000, and the number of clusters is k =10.

Table 1 The comparison of the objective functions on the artificial data

Size	*K*-means	Spherical *K*-means	*K*-medoids
d= 1,000	0	0	0
d= 2,000	1.863e–09	0	0
d= 5,000	3.725e–09	2.842e–13	3.275e–09
d= 10,000	3.725e–09	1.137e–13	7.451e–09
d= 20,000	1.490e–08	6.253e–13	0
d= 50,000	2.980e–08	0	5.960e–08

Table 2 The comparison of the run time on the artificial data

Size	*K*-means	Spherical *K*-means	*K*-medoids
d= 1,000	1.01	1.05	1.00
d= 2,000	1.46	1.13	1.30
d= 5,000	4.30	2.59	2.73
d= 10,000	7.97	4.71	5.19
d= 20,000	17.76	8.42	10.09
d= 50,000	44.10	33.60	26.30

Table 1 records the differences of K-means, spherical K-means and K-medoids objective functions between the original space and the feature space. It can be seen that the clustering results of the original space and the feature space are consistent from in any initial value. Table 2 records the running time ratio of K-means, spherical K-means and K-medoids in the original space and feature space. It can be see that when the dimension of d is small (compared with the number of samples), clustering in the feature space does not speed up the algorithm, because clustering in the feature space also needs to calculate the eigenvalues decomposition. However, with the increase of in dimension, K-means clustering in the feature space should to save a lot of time (when d = 50,000, the running time of the algorithm is reduced by 44 times).

Next, we will test the performance of the algorithm on image data and DNA data. These two kinds of data from different fields, have different meanings and the scale of data is also different, which is helpful to verify the advantages and disadvantages of our algorithm in different data. Table 3 records information of various data. The first three data are image data and the last four data are DNA data.

- The AT&T ORL database [39] consists of cropped face images of d = 112 $\times\times$ 92 pixels cropped face images with n = 400 face images from k = 10 different persons, and each person contains 40 sample images captured at different conditions. All images were taken against a dark homogeneous background with the subjects in an upright, frontal position.

- The Yale database [40] consists of cropped face images of d = 100$\times\times$100 pixels cropped face images with n = 165 face images from k = 15 different persons, each of which includes 11 images. They refer to some different facial expressions or configurations, i.e. glasses, happy, normal, sad, sleepy, surprised, and wink.

- The COIL-20 database [41] consists of grey-scale images of d = 128$\times\times$128 pixels gray-scale images with n = 1,440 objects images from k = 20 different objects. The objects were placed on a motorizsed turntable against a black background. The turntable was rotated through 360 degrees to vary object pose with respect to axed camera. Images of the objects were taken at pose intervals of 5 degrees.

- The CMD data [42] consists of d = 7,129 dimensions with n = 60 cotton microsatellites from k = 2 different characters. In addition, CMD displays data for three of the microsatellite projects that have been screened against a panel of core germplasm. The standardizsed panel consists of 12 diverse genotypes

Table 3 The date information in the algorithm test

Name	d	n	k
ORL	4,096	400	20
YALE	4,096	165	15
COIL20	16,384	1,440	20
CMD	7,129	60	2
DLBCL	7,129	77	2
LunG	1,000	197	4
Prostate	12,600	102	2

Table 4 The comparison of the objective functions on the actual data

Name	K-means	Spherical K-means	K-medoids
ORL	2.309e–14	1.705e–13	1.243e–13
YALE	4.263e–14	2.398e–14	7.105e–15
COIL20	2.757e–12	1.121e–11	4.547e–13
CMD	7.105e–15	6.128e–14	1.776e–14
DLBCL	7.105e–15	9.548e–14	3.730e–14
LunG	3.908e–14	4.796e–14	3.553e–14
Prostate	7.105e–15	1.172e–13	3.553e–14

including genetic standards, mapping parents, BAC donors, subgenome representatives, unique breeding lines, exotic introgression sources, and contemporary Upland cottons with significant acreage.

- The DLBCL data [43] consists of d = 7,129 dimensions with n = 77 DLBCL patients from k = 2 different factors. The original data frame with had over more than 8,000 observations (rows) on the following 3three markers (rows) and contained measurements from biopsies of 30 DLBCL patients. Each sample was stained with three antibodies,: CD3, CD5, and CD19.

- The LunG data [44] consists of d = 1,000 dimensions with n = 197 lung cancer patients from k = 4 different factors.

- The Prostate data [45] consists of d = 12,600 dimensions with n = 102 prostate cancer patients from k = 2 different factors. The original data set from contained 97 men who haved prostate cancer and recorded the information of the patients.

Table 4 records the differences in objective functions of K-means, spherical K-means and K-medoids objective functions between the original space and the feature space. It can be seen that the clustering results of the original space and the feature space are consistent from for any initial value. Table 5 records the running time ratio of K-means, spherical K-means and K-medoids in the original space and feature space. It can be see that the acceleration effect of the algorithm for LunG data is not obvious, because the data dimension is not high (d = 1,000) and the number of samples is relatively large (n = 197). The algorithm has the best acceleration effect (more than 10 times) for Prostate data, mainly due to the high dimension and small number of data.

5 Conclusion

In this paper, we aim at the low efficiency of K-means algorithm caused by high-dimensional data. The clustering algorithm in the feature space is proposed to improve the efficiency of the algorithm while ensuring the accuracy. We also point out that as long as the clustering centere and distance function satisfy certain

Table 5 The comparison of the run times on the actual data

Name	*K*-means	Spherical *K*-means	*K*-medoids
ORL	10.77	8.45	2.74
YALE	8.88	2.89	3.96
COIL20	8.86	6.22	5.70
CMD	10.22	4.86	6.61
DLBCL	7.70	4.76	6.92
LunG	5.13	1.21	1.96
Prostate	15.47	10.96	15.44

conditions, *K*-means type problems and corresponding algorithms can be accelerated with our ideas. In addition, we demonstrate in detail that the pre-processing steps (data dimensionality reduction) of our proposed method perfectly match the clustering steps (low-dimensional clustering) for the problem of high-dimensional Big Data problem.

Acknowledgements
The work described in this paper was supported by the Science and Technology Research Program of Chongqing Municipal Education Commission (Grant No. KJ1709207).

References

[1] Y. Y. Tang, Y. Tao and E. C. M. Lam, (2002), New method for feature extraction based on fractal behavior, *Pattern Recognize*, 35, 1071-1081, DOI: 10.1016/S0031-3203(01)00095-4.
[2] Y. Y. Tang, L. Yang and J. Liu, (2000), Characterization of dirac-structure edges with wavelet transform, *IEEE Transactions on Cybernetics*, 30, 93-109, DOI: 10.1109/3477.826950.
[3] T. Zhang, B. Fang, Y. Yuan, Y. Y. Yang, Z. Shang and B. Xu, (2010), Generalized discriminate analysis: A matrix exponential approach, *IEEE Transactions on Cybernetics*, 40, 186-197, DOI: 10.1109/TSMCB.2009.2024759.
[4] Y. Y. Tang and X. You, (2003), Skeletonization of ribbon-like shapes based on a new wavelet function, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 1118-1133, DOI: 10.1109/TPAMI.2003.1227987.
[5] T. Xie, P. Ren, T. Zhang and Y. Y. Tang, (2018), Distribution preserving learning for unsupervised feature selection, *Neurocomputing*, 289, 231-240, DOI: 10.1016/j.neucom.2018.02.032.
[6] T. Zhang, B. Fang, Y. Y. Tang, G. He and J. Wen, (2008), Topology preserving non-negative matrix factorization for face recognition, *IEEE Transactions on Image Processing*, 17, 574-584, DOI: 10.1109/CIS.2007.82.
[7] T. Zhang, Y. Y. Tang, Z. Shang and X. Liu, (2009), Face recognition under varying illumination using gradientfaces, *IEEE Transactions on Image Processing*, 18, 2599-2606, DOI: 10.1109/TIP.2009.2028255.
[8] J. Han, M. Kamber and J. Pei, (2011), Data mining: concepts and technique, *Morgan Kaufmann Press.*
[9] G. Sudipto, R.Rajeev and S. Kyuseok, (2001), CURE: An efficient clustering algorithm for large databases, *Information Systems*, 26, 35–58, DOI: 10.1016/S0306-4379(01)00008-4.
[10] T. Xie and F. Chen, (2018), Non-convex clustering via proximal alternating linearized minimization method, *International Journal of Wavelets, Multisolution and Information Processing*, 16, 13-25, DOI: 10.1142/S0219691318400131.
[11] P. Hoyer, (2004), Nonnegative matrix factorization with sparseness constraints, *Machine Learning Research*, 9, 1457-1469.
[12] D. D. Lee and H. S. Seung, (1999), Learning the parts of objects by nonnegative matrix factorization, *Nature*, 401, 788-791.
[13] B. Ren, P. Laurent, G. B. Zhu and D. Gaspard, (2018), Nonnegative matrix factorization: robust extraction of extended structures, *The Astrophysical Journal*, 852, 104-121.
[14] Y. X. Wang and Y. J. Zhang, (2013), Nonnegative matrix factorization: A comprehensive review, *IEEE Transactions on Knowledge and Data Engineering*, 25, 1336–1353, DOI: 10.1109/TKDE.2012.51.
[15] D. Comaniciu and P. Meer, (2002), Mean shift: A robust approach toward feature space analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 603-619, DOI: 10.1109/34.1000236.
[16] Y. Du, B. Sun, R. Lu, C. Zhang and H. Wu, (2019), A method for detecting high-frequency oscillations using semi-supervised k-means and mean shift clustering, *Neurocomputing*, 350, 102-107, DOI: 10.1016/j.neucom.2019.03.055.
[17] T. Duong, G. Beck, H. Azzag and M. Lebbah, (2016), Nearest neighbor estimators of density derivatives, with appli-

cation to mean shift clustering, *Pattern Recognition Letter,* 80, 224-230, DOI: 10.1016/j.patrec.2016.06.021.
[18] D. Cai and X. Chen, (2015), Large scale spectral clustering via landmark-based sparse representation, *IEEE Transactions on Cybernetics*, 45, 1669-1680, DOI**:** 10.1109/TCYB.2014.2358564.
[19] U. V. Luxburg, (2007), A tutorial on spectral clustering, Statistics and Computing, 17, 395-416, DOI: 10.1007/s11222-007-9033-z.
[20] J. Shi and J. Malik, (2000), Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 888-905, DOI: 10.1109/34.868688.
[21] M. Brbis and I. Kopriva, (2018), Multi-view low-rank sparse subspace clustering, *Pattern Recognition*, 73, 247-258, DOI: 10.1016/j.patcog.2017.08.024.
[22] E. Elhamifar and R. Vidal, (2013), Sparse subspace clustering: algorithm, theory and application, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 2765-2781, DOI: 10.1109/TPAMI.2013.57.
[23] Y. Ma, A. Y. Yang, H. Derksen and R. Fossum, (2008), Estimation of subspace arrangements with applications in modeling and segmenting mixed data, *SIAM Review*, 50, 413-458, DOI: 10.1137/060655523.
[24] A.K. Jain, (2014), Data clustering: 50 years beyond K-means, *Pattern Recognition Letters*, 33, 651-666, DOI: 10.1016/j.patrec.2009.09.011.
[25] S. Lloyd, (1982), Least squares quantization in PCM, *IEEE Transactions on Information Theory*, 28, 129–137, DOI: 10.1109/TIT.1982.1056489.
[26] H. Park and C. Jun, (2009), A simple and fast algorithm for K-medoids clustering, *Expert Systems with Applications,* 36, 3336-3341, DOI: 10.1016/j.eswa.2008.01.039.
[27] S. Yu, S. Chu, C. Wang Y. Chan and T. Chang, (2018), Two improved K-means algorithms, *Applied Soft Computing*, 68, 747-755, DOI: 10.1016/j.asoc.2017.08.032.
[28] D. Arthur and S. Vassilvitskii, (2007), K-means++++: the advantages of careful seeding, *Society for Industrial and Applied Mathematics*, 165, 1027-1035, DOI: 10.1145/1283383.1283494.
[29] O. Bachem, M. Lucic, S. H. Hassani and A. Krause, (2016), Fast and provably good seeding for k-means, *IEEE The 30th Conference on Neural Information Processing Systems*, 2016, 76-85.
[30] M. E. Celebi, H. A. Kingravi and P. A. Vela, (2013), A comparative study of efficient initialization methods for the k-means clustering algorithm, *Expert Systems with Applications*, 40, 200-210, DOI: 10.1016/j.eswa.2012.07.021.
[31] I. S. Dhilon, Y. Guan and B. Kulis, (2004), Kernel k-means, spectral clustering and normalized cuts, *ACM International Conference on Knowledge Discovery and Data Mining*, 2004, 551-556.
[32] G. Ball and D. Hall, ISODATA, (1965), A novel method of data analysis and pattern classification, *Stanford Research Institute Press.*
[33] S. A. E. Rahman, (2015), Hyperspectral imaging classification using ISODATA algorithm: big data challenge, *IEEE the 5th International Conference on e-Learning,* 2015, 271-280.
[34] J. C. Dunn, (1973), A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, *Journal of Cybernetics*, 3, 32-57, DOI: 10.1080/01969727308546046.
[35] I. S. Dhillon and D. S. Modha, (2001), Concept decompositions for large sparse text data using clustering, *Machine Learning*, 42, 143-175.
[36] A. Banerjee, (2004), Clustering with Bregman Divergences, *SIAM International Conference on Data Mining*, 2004, 234–245.
[37] Y. Linde, A. Buzo and R. Gray, (1980), An algorithm for vector quantizer design, *IEEE Transaction Communication*, 28, 84-94, DOI: 10.1109/TCOM.1980.1094577.
[38] J. Mao and A. K. Jain, (1996), A self-organizing network for hyper ellipsoidal clustering, *IEEE Transactions on Neural Networks*, 7, 16-29, DOI: 10.1109/ICNN.1994.374705.
[39] Online, (2019), ORL, http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.
[40] Online, (2019), Yale database, http://cvc.yale edu/projects/yalefaces.html.
[41] Online, (2019), COIL-20 database, ftp://zen.cs.columbia.edu/.
[42] Online, (2019), CMD data, http://www.cottonssr.org/.
[43] Online, (2019), DLBCL data, http://flowrepository.org/id/FR-FCM-ZZYY/.
[44] Online, (2019), LunG data, http://biogps.org/dataset/tag/lung/.
[45] Online, (2019), Prostate data, http://statweb.stanford.edu/~tibs/ElemStatLearn/prostate.data/.