


Research Article

A Robust k -Means Clustering Algorithm Based on Observation Point Mechanism

Xiaoliang Zhang,¹ Yulin He ,^{1,2,3} Yi Jin,^{4,5} Honglian Qin,¹ Muhammad Azhar,¹
and Joshua Zhexue Huang^{1,2,3}

¹College of Computer Science & Software Engineering, Shenzhen University, Shenzhen 518060, China

²National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen 518060, China

³Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University, Shenzhen 518060, China

⁴Department of Trace Inspection Technology, Criminal Investigation Police University of China, Shenyang 110854, China

⁵Key Laboratory of Trace Inspection and Identification Technology of The Ministry of Public Security, Shenyang 110854, China

Correspondence should be addressed to Yulin He; yulinhe@szu.edu.cn

Received 11 September 2019; Revised 31 January 2020; Accepted 4 February 2020; Published 30 March 2020

Academic Editor: Xianggui Guo

Copyright © 2020 Xiaoliang Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The k -means algorithm is sensitive to the outliers. In this paper, we propose a robust two-stage k -means clustering algorithm based on the observation point mechanism, which can accurately discover the cluster centers without the disturbance of outliers. In the first stage, a small subset of the original data set is selected based on a set of nondegenerate observation points. The subset is a good representation of the original data set because it only contains all those points that have a higher density of the original data set and does not include the outliers. In the second stage, we use the k -means clustering algorithm to cluster the selected subset and find the proper cluster centers as the true cluster centers of the original data set. Based on these cluster centers, the rest data points of the original data set are assigned to the clusters whose centers are the closest to the data points. The theoretical analysis and experimental results show that the proposed clustering algorithm has the lower computational complexity and better robustness in comparison with k -means clustering algorithm, thus demonstrating the feasibility and effectiveness of our proposed clustering algorithm.

1. Introduction

Clustering is an important research branch of data mining. The k -means algorithm is one of the most popular clustering methods [1]. When performing k -means clustering, we usually use a local search to find the solution [2, 3], i.e., selecting k points $\mu_1, \mu_2, \dots, \mu_k$ as the initial cluster centers and then optimizing them by an iterative process to minimize the following objective function (see, for example, [4, 5]):

$$E = \sum_{i=1}^k \sum_{X_j \in C_i} \|X_j - \mu_i\|_2^2, \quad (1)$$

where X_j is the j -th data point belonging to the i -th cluster C_i . It is well known that the solution of equation (1) is affected by the initial values of μ_i ($i = 1, 2, \dots, k$).

In order to choose μ_i properly, the k -means++ algorithm [6] picks out a set of points as the initial center points whose distances between each other are as large as possible. However, this method for choosing the initial center points is sensitive to outliers [7–9]. Some methods use the subsets of the original data set to determine μ_i . For instance, the CLARA [10] and CLARANS [11] algorithms use PAM [12] to calculate the initial cluster centers from the random subsets of the original data set. The sampling-based methods weaken the sensitivity because the sampling process can discard some outliers in the original data set, but it cannot guarantee all outliers to be ignored in the sampling process. Therefore, the remaining outliers in subsets still affect the clustering results.

The automatic clustering algorithms are attracting more and more attention from the academic community, e.g., the

density-based spatial clustering of applications with noise (DBSCAN) algorithm [13–15], depth difference-based clustering algorithm [16], and Tanir’s method [17]. Recently, a new automatic clustering algorithm named I-nice was proposed in [18]. Inspired by the observation point mechanism of I-nice algorithm, we propose a two-stage k -means clustering algorithm in this paper to find the cluster centers from a subset of the original data set with all outliers removed. In the first stage, we select a small subset of original data set based on a set of nondegenerate observation points. The subset contains only all the higher density points of the original data set and does not have the outliers. Therefore, it is a good representation of the original data set for finding the proper cluster centers. In the second stage, we perform the k -means algorithm on the subset to obtain a set of cluster centers and then the other points in the original data set can be clustered accordingly.

Selecting the subset in the first stage is based on a set of $d + 1$ nondegenerate observation points that are assigned to the data space \mathbb{R}^d , where d is the dimension of data points. For each observation point, we compute a set of distances between it and all data points in the original data set. The set of distances generates a distance distribution with respect to the observation point. From the distance distribution, we identify the dense areas and extract the subset of data points in the dense areas. Then, we take the intersection of all $d + 1$ subsets of data points in all dense areas from those $d + 1$ distance distributions. After refining this intersection subset of data points, we obtain a subset without outliers of the original data set. Therefore, it can be used to find the proper cluster centers. Finally, we conduct some convictive experiments to validate the effectiveness of our proposed algorithm and the experimental results demonstrate that our proposed algorithm is robust to outliers.

The remainder of this paper is organized as follows. We describe the related mathematical principles of our algorithm in Section 2. The details of two-stage k -means clustering algorithm and its pseudocode are presented in Section 3. In Section 4, we present a series of experiments to validate the feasibility of our proposed algorithm. Finally, we summarize the conclusions and future work in Section 5.

2. Mathematical Principles

Definition 1. Suppose $\mathbb{D} = \{X_1, X_2, \dots, X_N\}$ is a data set including N data points with d dimensions. Given an observation point $O \in \mathbb{R}^d$, we say $\tilde{\mathbb{D}} = \{d(X_i, O) : i = 1, 2, \dots, N\}$ is a generated distance set of \mathbb{D} with respect to the observation point O , where $d(X, Y)$ denotes the Euclidean distance between X and Y .

Given a data set \mathbb{D} and an observation point O , we have

$$|d(X, O) - d(Y, O)| \leq d(X, Y), \quad (2)$$

by the triangle inequality for every $X, Y \in \mathbb{D}$. Hence, the distance between two data points in \mathbb{D} is larger than the difference of their corresponding two distances in $\tilde{\mathbb{D}}$. Therefore, for any positive number r and a point $X \in \mathbb{D}$, the number of points in \mathbb{D} with distances to X less than r is not

greater than the number of points in $\tilde{\mathbb{D}}$ whose distances to $d(X, O)$ are less than r . In particular, if X is a proper cluster center in \mathbb{D} , $d(X, O)$ will be a data point in $\tilde{\mathbb{D}}$ which has more points close to it. That is to say, if X is a dense point in \mathbb{D} , it is also corresponding to a dense point in $\tilde{\mathbb{D}}$.

Unfortunately, the converse is not true. Because two points in $\tilde{\mathbb{D}}$ which have a small distance may correspond to two points in \mathbb{D} that have a large distance, a proper cluster center of $\tilde{\mathbb{D}}$ may not be corresponding to a proper cluster center of \mathbb{D} . Hence, we can deduce that $\tilde{\mathbb{D}}$ retains the partial clustering information of \mathbb{D} . In order to obtain more clustering information of \mathbb{D} , one possible way is to choose more observation points to generate more distance sets and then combines all those different clustering information together. This is the main idea behind our new algorithm. We provide the following two theorems to guarantee the correctness of the abovementioned statements.

Definition 2. Given a set of $d + 1$ points $\{O_0, O_1, \dots, O_d\}$ where $O_i = (a_{i1}, a_{i2}, \dots, a_{id})$ for $i = 0, 1, 2, \dots, d$, define the generating matrix A of $\{O_0, O_1, \dots, O_d\}$ as

$$A = \begin{pmatrix} a_{11} - a_{01} & a_{12} - a_{02} & \cdots & a_{1d} - a_{0d} \\ a_{21} - a_{01} & a_{22} - a_{02} & \cdots & a_{2d} - a_{0d} \\ \vdots & \vdots & & \vdots \\ a_{d1} - a_{01} & a_{d2} - a_{02} & \cdots & a_{dd} - a_{0d} \end{pmatrix}. \quad (3)$$

If the determinant of A is not equal to zero, we say A is nondegenerate and $\{O_0, O_1, \dots, O_d\}$ is a set of nondegenerate points.

Theorem 1. Suppose $\{O_0, O_1, \dots, O_d\}$ is a set of nondegenerate points and let $X_1, X_2 \in \mathbb{R}^d$. If

$$d(X_1, O_i) = d(X_2, O_i), \quad (4)$$

for all $i = 0, 1, \dots, d$, then $X_1 = X_2$.

Proof. Assume $X_1 = (x_{11}, x_{12}, \dots, x_{1d})$ and $X_2 = (x_{21}, x_{22}, \dots, x_{2d})$. For each $i = 1, 2, \dots, d$, we have

$$[d(X_1, O_i)]^2 - [d(X_1, O_0)]^2 = [d(X_2, O_i)]^2 - [d(X_2, O_0)]^2, \quad (5)$$

i.e.,

$$\sum_{t=1}^d (x_{1t} - a_{it})^2 - \sum_{t=1}^d (x_{1t} - a_{0t})^2 = \sum_{t=1}^d (x_{2t} - a_{it})^2 - \sum_{t=1}^d (x_{2t} - a_{0t})^2. \quad (6)$$

By simplifying equation (6), we can obtain

$$\sum_{t=1}^d (2x_{1t} - a_{it} - a_{0t})(a_{0t} - a_{it}) = \sum_{t=1}^d (2x_{2t} - a_{it} - a_{0t})(a_{0t} - a_{it}). \quad (7)$$

Thus, we have

$$\sum_{t=1}^d (a_{it} - a_{0t})(x_{1t} - x_{2t}) = 0, \quad i = 1, 2, \dots, d. \quad (8)$$

Since the coefficient matrix of (8) is nondegenerate, we can get that

$$x_{1t} - x_{2t} = 0, \quad t = 1, \dots, d. \quad (9)$$

Thus, $X_1 = X_2$. \square

Remark 1. For the convenience of calculation, we can choose $O_0 = (0, 0, \dots, 0)$, $O_i = (0, \dots, 0, 1, 0, \dots, 0)$, $i = 1, 2, \dots, d$ as the set of nondegenerate observation points. In fact, if $X = (x_1, x_2, \dots, x_d)$, then for each i , it has

$$\begin{aligned} d(X, O_i)^2 &= \sum_j (x_j - o_{ij})^2 = \sum_{j \neq i} (x_j - 0)^2 + (x_i - 1)^2 \\ &= \sum_j x_j^2 - 2 * x_i + 1 = d(X, O_0)^2 - 2 * x_i + 1. \end{aligned} \quad (10)$$

Thus, if we have obtained the distance between X and O_0 , then computing the square of the distance between X and O_i ($i = 1, 2, \dots, d$) will convert to addition operation three times, which will decrease the time complexity in generating those distance sets.

Remark 2. If the number of observation points is less than $d + 1$, Theorem 1 does not hold true. For example, if we choose

$$\begin{aligned} O_0 &= (0, 0), O_1 = (1, 0), O_2 = (0, 1), X_1 = (0.2, 0.2), \\ X_2 &= (0.6, 0.6), \end{aligned} \quad (11)$$

then, it has

$$\begin{aligned} d(X_1, O_1) &= d(X_2, O_1), \\ d(X_1, O_2) &= d(X_2, O_2), \end{aligned} \quad (12)$$

but $X_1 \neq X_2$. By Theorem 1, we can confirm that all different clustering points can be distinguished by choosing a set of nondegenerate points as the observation points. Thus, $d + 1$ is the minimum number of the observation points to distinguish all the cluster centers of the original data set.

Theorem 2. Suppose $O_0 = (0, 0, \dots, 0)$, $O_i = (0, \dots, 0, 1, 0, \dots, 0)$, $i = 1, 2, \dots, d$. Let $X_1, X_2 \in \mathbb{R}^d$ and set

$$d(X_j, O_i) = d_{ji}, \quad (13)$$

for $i = 0, 1, \dots, d$ and $j = 1, 2$. If, for each $i = 0, 1, \dots, d$,

$$\begin{aligned} |d_{1i} - d_{2i}| &< r, \\ M &= \max_{j=1,2, i=0,1,\dots,d} \{d_{ji}\}, \end{aligned} \quad (14)$$

then,

$$d(X_1, X_2) \leq 2\sqrt{d}Mr. \quad (15)$$

Proof. For $j = 1, 2$, we set $X_j = (x_{j1}, x_{j2}, \dots, x_{jd})$. We have

$$\begin{cases} \sum_{t=1}^d x_{jt}^2 = d_{j0}^2, & j = 1, 2; \\ \sum_{t=1}^d x_{jt}^2 - 2x_{ji} + 1 = d_{ji}^2, & i = 1, 2, \dots, d, \quad j = 1, 2. \end{cases} \quad (16)$$

Solving the system of equation (16) results in

$$x_{ji} = \frac{1}{2}(1 + d_{j0}^2 - d_{ji}^2), \quad i = 1, 2, \dots, d, \quad j = 1, 2. \quad (17)$$

Then, we can obtain

$$\begin{aligned} d(X_1, X_2)^2 &= \sum_{t=1}^d (x_{1t} - x_{2t})^2 \\ &= \sum_{t=1}^d \left[\frac{1}{2}(1 + d_{10}^2 - d_{1t}^2) - \frac{1}{2}(1 + d_{20}^2 - d_{2t}^2) \right]^2 \\ &= \frac{1}{4} \sum_{t=1}^d [(d_{10}^2 - d_{20}^2) - (d_{1t}^2 - d_{2t}^2)]^2 \\ &\leq \frac{1}{4} \sum_{t=1}^d [r(d_{10} + d_{20} + d_{1t} + d_{2t})]^2 \\ &\leq 4dM^2r^2, \end{aligned} \quad (18)$$

which yields $d(X_1, X_2) \leq 2\sqrt{d}Mr$. \square

Remark 3. If we normalize the original data set \mathbb{D} , for example, we perform the *min-max normalization* on \mathbb{D} , and we can deduce that $M \leq \sqrt{d}$.

Remark 4. Suppose \mathbb{A} is a generated distance set of \mathbb{D} with respect to the observation point O . We cannot confirm whether two elements in \mathbb{A} which have a small difference are corresponding to two data points in \mathbb{D} which also have a small distance. But by Theorem 2, if all the $d + 1$ pairs of generated distances of X and Y have a small difference, then X must have a small distance to Y . This can be used to adjust the dense of the selected subset.

Remark 5. The observation point mechanism aims to transform the original multidimensional data points into one-dimensional distance points, which is different from the landmark point or representative point mechanisms. The landmark point [19] is the core of landmark-based spectral clustering (LSC) algorithm which generates some representative data points as the landmarks and represents the remaining data points as linear combinations of these landmarks. The representative points [20] are the subset of original data set and used in the ultrascaleable spectral clustering (U-SPEC) algorithm to alleviate the huge computational burden of spectral clustering. The observation points are designed to enhance the robustness of k -means

clustering, while the landmark points or representative points are used to speed up the spectral clustering.

3. The Proposed Two-Stage k -Means Clustering Algorithm

Given a data set \mathbb{D} with N objects, we want to partition \mathbb{D} into k clusters. The main idea of our two-stage k -means clustering algorithm is that we only need to deal with a small subset of \mathbb{D} which has a similar clustering structure to \mathbb{D} . In order to select a proper subset with the abovementioned property, we need to discard all outliers in \mathbb{D} and retain a portion of those points that are close to the cluster centers.

3.1. Description of Algorithm

3.1.1. Generating $d + 1$ Distance Sets in the First Stage. First of all, we conduct the normalization operation on the original data set \mathbb{D} . Set $\mathbb{D} = \{X_1, X_2, \dots, X_N\}$, where $X_j = (x_{j1}, x_{j2}, \dots, x_{jd})$ for $j = 1, 2, \dots, N$. Suppose

$$M = \max_{j=1,2,\dots,N; i=1,2,\dots,d} \{x_{ji}\} \text{ and } m = \min_{j=1,2,\dots,N; i=1,2,\dots,d} \{x_{ji}\}. \quad (19)$$

Then, we transform \mathbb{D} into $\tilde{\mathbb{D}}$ with $X_j = (x_{j1}, x_{j2}, \dots, x_{jd})$ corresponding to $\tilde{X}_j = (((x_{j1} - m)/(M - m)), ((x_{j2} - m)/(M - m)), \dots, ((x_{jd} - m)/(M - m)))$. Obviously, the transformation on \mathbb{D} is a composition of a translation transformation and a dilation transformation. The dilation factor $1/(M - m)$ is the same for each dimension; hence, the dilation transformation does not change the cluster structure. Because the translation transformation also does not change the cluster structure of a dataset, the cluster structure of \mathbb{D} is totally the same as that of $\tilde{\mathbb{D}}$. We also note that the value of every component of \tilde{X}_j is in the interval $[0, 1]$.

Let

$$O_0 = (0, 0, \dots, 0), \quad (20)$$

$$O_i = (0, \dots, \overset{i-1}{0}, \overset{i}{1}, \overset{i+1}{0}, \dots, 0), \quad i = 1, 2, \dots, d,$$

be the set of observation points. Denote $\tilde{\mathbb{D}}_j$ the generated distance set of $\tilde{\mathbb{D}}$ with respect to the observation point O_j ($j = 0, 1, \dots, d$), and we get $d + 1$ sets $\tilde{\mathbb{D}}_0, \tilde{\mathbb{D}}_1, \dots, \tilde{\mathbb{D}}_d$. For each data point \tilde{X} , we actually have mapped it to a $(d + 1)$ -dimensional vector $(d(\tilde{X}, O_0), d(\tilde{X}, O_1), \dots, d(\tilde{X}, O_d))$. Theorem 1 shows that we can identify \tilde{X} by the $(d + 1)$ -dimensional vector, and hence, it is reasonable to expect that the clustering structure about $\tilde{\mathbb{D}}$ can be deduced by those $d + 1$ distance sets.

3.1.2. Selecting a Representative Subset of $\tilde{\mathbb{D}}$ in the First Stage. For each $\tilde{\mathbb{D}}_i$, we can get a set S_i consisting of all candidate higher density points of $\tilde{\mathbb{D}}_i$, by using the *grid-based clustering methods* (e.g., [21]). For example, first, we arrange $\tilde{\mathbb{D}}_i$ in the ascending order. Second, a fixed value δ_i is selected to be a quantile of $\text{diff}(\tilde{\mathbb{D}}_i)$. Third, for each s in $\tilde{\mathbb{D}}_i$, we counter the

number of elements of $\tilde{\mathbb{D}}_i$ in the interval $(s - \delta_i, s + \delta_i)$. Thus, we obtain a positive integer sequence, where each member indicates the relative size of the density of the corresponding element of $\tilde{\mathbb{D}}_i$. Finally, we select out those s in $\tilde{\mathbb{D}}_i$ such that the corresponding integer number is either a local maximum or beyond a threshold.

In the following experiments, we will set δ_i as two times the p -th percentile of set $\text{diff}(\mathcal{D}_i)$ for some p , where \mathcal{D}_i is the rearrangement of $\tilde{\mathbb{D}}_i$ in the ascending order and $\text{diff}(\mathcal{D}_i)$ is the sequence of the first-order difference on \mathcal{D}_i . Denote N as the cardinality of $\tilde{\mathbb{D}}_i$. If N is small, we usually choose a smaller p ; for example, $p = 75$. If N is very large, we choose a bigger p ; e.g., $p = 99$. Otherwise, we can choose a proper p between them; e.g., $p = 90$.

Now, we have obtained $d + 1$ sets S_0, S_1, \dots, S_d with each one containing all the higher-density points of the corresponding distance set. By the triangle inequality, we have the following property:

If there is an $i \in \{0, 1, \dots, d\}$ such that $d_i = d(X, O_i)$ is not in S_i , then X cannot be a higher-density point of $\tilde{\mathbb{D}}$.

According to this property, we can select a subset \mathbb{S} of $\tilde{\mathbb{D}}$ whose distances to the i -th observation point are in S_i for all $i \in \{0, 1, \dots, d\}$.

For each point $X \in \tilde{\mathbb{D}}$, we have mapped it to a $(d + 1)$ -dimensional vector. By Remark 4 of Theorem 2, all the $d + 1$ pairs of the corresponding components of two points that belong to the same cluster will have a little difference between them. But it is possible that there are some data points which have some components that have the little difference with that of one cluster center and have the other components that have the little difference with that of another cluster center. In such case, few outliers may be missed by the above selection criterion. To discard those few outliers in \mathbb{S} and decrease the number of elements of \mathbb{S} , we need to refine \mathbb{S} . We have the following criterion according to Remark 4 of Theorem 2.

Suppose X_1 has been selected. Given a data point X_2 , if, for every $i \in \{0, 1, \dots, d\}$, $d_{1i} = d(X_1, O_i)$ and $d_{2i} = d(X_2, O_i)$ have a small difference, then we discard X_2 .

We denote $\mathbb{S} = \{Y_1, Y_2, \dots, Y_m\}$ and set

$$\begin{aligned} \mathbb{S}_c &= \emptyset, \\ \mathbb{S}_a &= \emptyset. \end{aligned} \quad (21)$$

We also need a counter to indicate the density of each data point of \mathbb{S}_a . Firstly, we let $Y_1 \in \mathbb{S}_a$ and make the indicative number of Y_1 to be 1. We then sequentially choose the data points in \mathbb{S} and dynamically construct \mathbb{S}_c and \mathbb{S}_a according to the following process. Suppose we choose Y_i from \mathbb{S} , then we compute the distance between Y_i and each data point in \mathbb{S}_a . If there are some distances less than a threshold value δ , we add 1 to each of the counter of data point that corresponds to these distances and then discard Y_i . Meanwhile, if the counter number of a data point in \mathbb{S}_a is bigger than another threshold value n , then we remove this data point from \mathbb{S}_a and add it into \mathbb{S}_c . But if each one of the point in \mathbb{S}_a has distance to Y_i bigger than δ , we continue to check whether there is a point in \mathbb{S}_c that has distance to Y_i less than δ ; we will discard Y_i if there is any and we will add

Y_i to \mathbb{S}_c if not. Finally, we obtain a set \mathbb{S}_c that closely represents the original data set and the size of it is smaller than that of the original data set. Furthermore, all outliers of original data set are not included in this selected subset.

3.1.3. Clustering \mathbb{S}_c and $\tilde{\mathbb{D}}$ in the Second Stage. Since the selected set has discarded all outliers and has a smaller size than the original data set, the running time will decrease significantly when performing the k -means algorithm on the selected subset. Furthermore, because the subset closely represents the original data set, the cluster centers will also be suitable to be chosen as the cluster centers of the original data set. When we have identified the cluster centers, it is easy to cluster the whole data set. The pseudo-code of our proposed algorithm is presented in Algorithm 1.

3.2. Analysis of Computational Complexity. In this section, we analyze the computational complexity of the proposed algorithm. When running the classical k -means algorithm, each iteration needs to compute the distances between each data point in the whole data and those new modified cluster centers, which has a time complexity of $O(Nkd)$. In our algorithm, the time cost in the first stage mainly consists of four parts. The first part is to generate $d + 1$ one-dimensional data sets, which has a time complexity of $O(Nd)$. The second one is to find those intervals which contain the local maximum of distances, which has a time complexity of $O(N)$. The third part is to select \mathbb{S} , which has the time complexity $O(Nd)$. In the fourth part, we refine \mathbb{S} and obtain the subset \mathbb{S}_c , which has the time complexity $O(\tilde{N}^2)$, where \tilde{N} denotes the cardinality of \mathbb{S} . Thus, the time complexity of the first stage is $O(Nd + \tilde{N}^2)$. At the second stage, we will perform the k -means algorithm on \mathbb{S}_c and each iteration will have a time complexity less than $\tilde{N}_1 kd$, where \tilde{N}_1 denotes the cardinality of \mathbb{S}_c . Because $\tilde{N}_1 < \tilde{N}$, the total time complexity of the new algorithm is $O(Nd + \tilde{N}kd + \tilde{N}^2)$.

We note that the time complexity of the fourth part in the first stage is usually much less than $O(\tilde{N}^2)$. Since many data points have been discarded when constructing the sets \mathbb{S}_c and \mathbb{S}_a , we do not have to compute the distances with all \tilde{N} data points in \mathbb{S} .

4. Experimental Results and Analysis

We have conducted a series of experiments on 6 synthetic data sets and 3 benchmark data sets (UCI [22] and KEEL [23]) to validate the effectiveness of the proposed two-stage k -means clustering algorithm in this section. The synthetic data sets can be downloaded from BaiduPan (<https://pan.baidu.com/s/1MfS8JfQdJLHYSlpZdndLUQ>) with the extraction code "p3mc." We first present the clustering results of our proposed algorithm and the k -means algorithm on two synthetic data sets, i.e., the data set #1 and data set #2. The experimental results are shown in Figures 1 and 2. For simplicity, we only use the experimental results on data set #1 to explain the advantage of our proposed algorithm. There are two clusters in data set #1, where each cluster

includes 41 data points. The data points obey the 2-dimensional normal distributions with mean vectors (3, 11) and (12, 5) and covariance matrices $\begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix}$ and $\begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}$, respectively. There are also two outliers in data set #1.

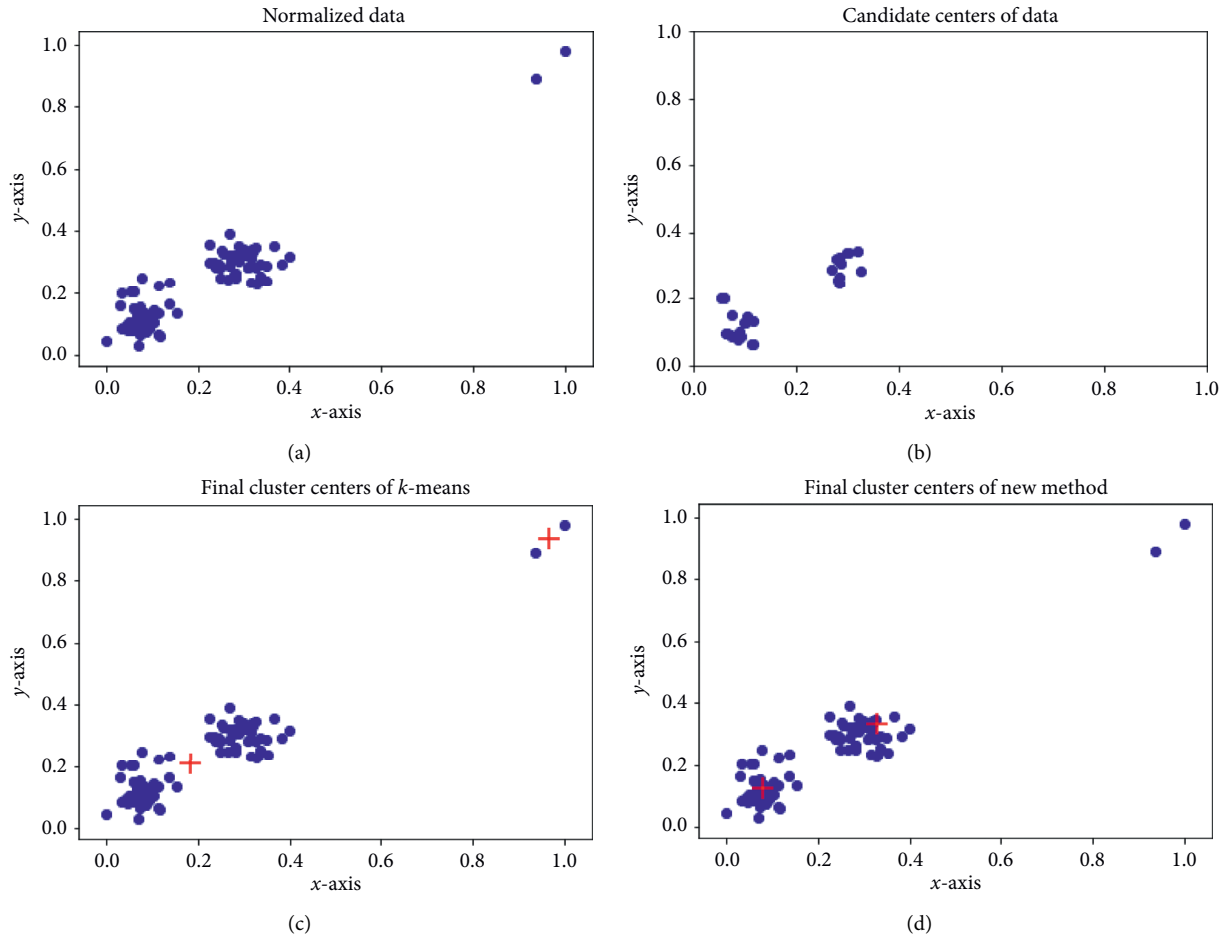
Figure 1(b) gives the selected data points of normalized data points corresponding to the data set #1 as shown in Figure 1(a). In Figure 1(b), we can find that outliers have been removed in the first stage of our proposed method. Figure 1(c) shows the clustering result of the k -means algorithm. We can see that outliers seriously impact the clustering result of the k -means algorithm, although there are only two outlier data points in the data set #1. The clustering result of our proposed method is presented in Figure 1(d), where the cluster center can be found correctly without the disturbance of outliers. The similar results can be found in Figure 2 for the data set #2 which includes 7 clusters and 10 outliers. The experimental results reflect that our proposed two-stage k -means clustering algorithm is not sensitive to outliers and can obtain the better clustering results than that of k -means clustering algorithm.

Furthermore, we choose another four synthetic data sets as shown in Figure 3 (only 2-dimensional illustration) and three real-world data sets to compare the clustering performances of our proposed algorithm with the k -means algorithm. The details of these data sets and experimental results are summarized in Table 1, where N is the number of the elements of the data set, t is the proportion of the outlier in the data set, k is the number of clusters, d is the dimension of data point, p is the percentile number, n_c is the cardinality of selected subset, $\text{ARI}_{k\text{means}}$ and $\text{Time}_{k\text{means}}$ are the adjusted Rand index (ARI) and time consumption of k -means algorithm, and ARI_{our} and Time_{our} are ARI and time consumption of our proposed algorithm. In Table 1, we can see that our proposed algorithm obtains the larger ARIs with the lower time consumption in comparison with k -means clustering algorithm on these synthetic data sets. For the real data sets without outliers, our algorithm can obtain the ARIs comparable to that of k -means algorithm. Nevertheless, the ARIs of k -means algorithm are severely degraded when the outliers are deliberately arranged in the real data sets, while the experimental results in Table 1 demonstrate that our proposed clustering algorithm is robust to the outliers. Table 2 shows the details of comparison on four large-scale synthetic data sets. The variables in Table 2 have the same meaning as that in Table 1. The comparison of time complexity between our proposed algorithm and k -means algorithm in Table 2 reflects that our algorithm has less time consumption than k -means algorithm. Especially, we can find that the superiority of our proposed method on time consumption is more obvious for data set with the larger size and dimension. Furthermore, the most time-consuming procedure in our algorithm, i.e., the selection of high-density distances for each generated distance set can be ran in the parallel way, which make our algorithm to be easily extended to perform the clustering task for large-scale data set.

Input:
 The number of clusters: k ;
 The number of percentile: p ;
 The original data set: $\mathbb{D} = \{X_i = (x_{i1}, x_{i2}, \dots, x_{id}) : i = 1, 2, \dots, N\}$;

Method:
 Normalize \mathbb{D} and generate $\tilde{\mathbb{D}} = \{\tilde{X}_i : i = 1, 2, \dots, N\}$;
for $t = 0$ to d **do**
 Set $\tilde{\mathbb{D}}_t = \{d_{it} : i = 1, 2, \dots, N\}$ where $d_{it} = d(\tilde{X}_i, O_t)$;
 Generate \mathcal{D}_t by rearranging $\tilde{\mathbb{D}}_t$ in ascending order and set δ_t be p -th percentile of $\text{diff}(\mathcal{D}_t)$;
 Set $m_t = \min \mathcal{D}_t$ and $M_t = ((\max \mathcal{D}_t) / \delta_t + 1) * \delta_t$;
 Equally divide the interval $[m_t, M_t]$ into intervals with length δ_t ;
 Let S_t be the union of those intervals that contain local maximum number of elements in \mathcal{D}_t ;
end for
 Select out a subset \mathbb{S} : $X_i \in \mathbb{S}$ if and only if $d_{it} \in S_t$ for all $t = 0, 1, \dots, d$;
 Refine \mathbb{S} and obtain the subsets \mathbb{S}_c ;
 Perform the k -means algorithm on \mathbb{S}_c ;
 Assign each $\tilde{X}_i (i = 1, 2, \dots, N)$ to the nearest center of the obtained cluster of \mathbb{S}_c ;

Output:
 The result of clustering.

ALGORITHM 1: Two-stage k -means clustering algorithm.FIGURE 1: Experimental results on synthetic data set #1 (the red '+'s are the cluster centers). (a) Normalized data. (b) Selected subset. (c) Clustering centers of k -means algorithm. (d) Clustering centers of our proposed algorithm.

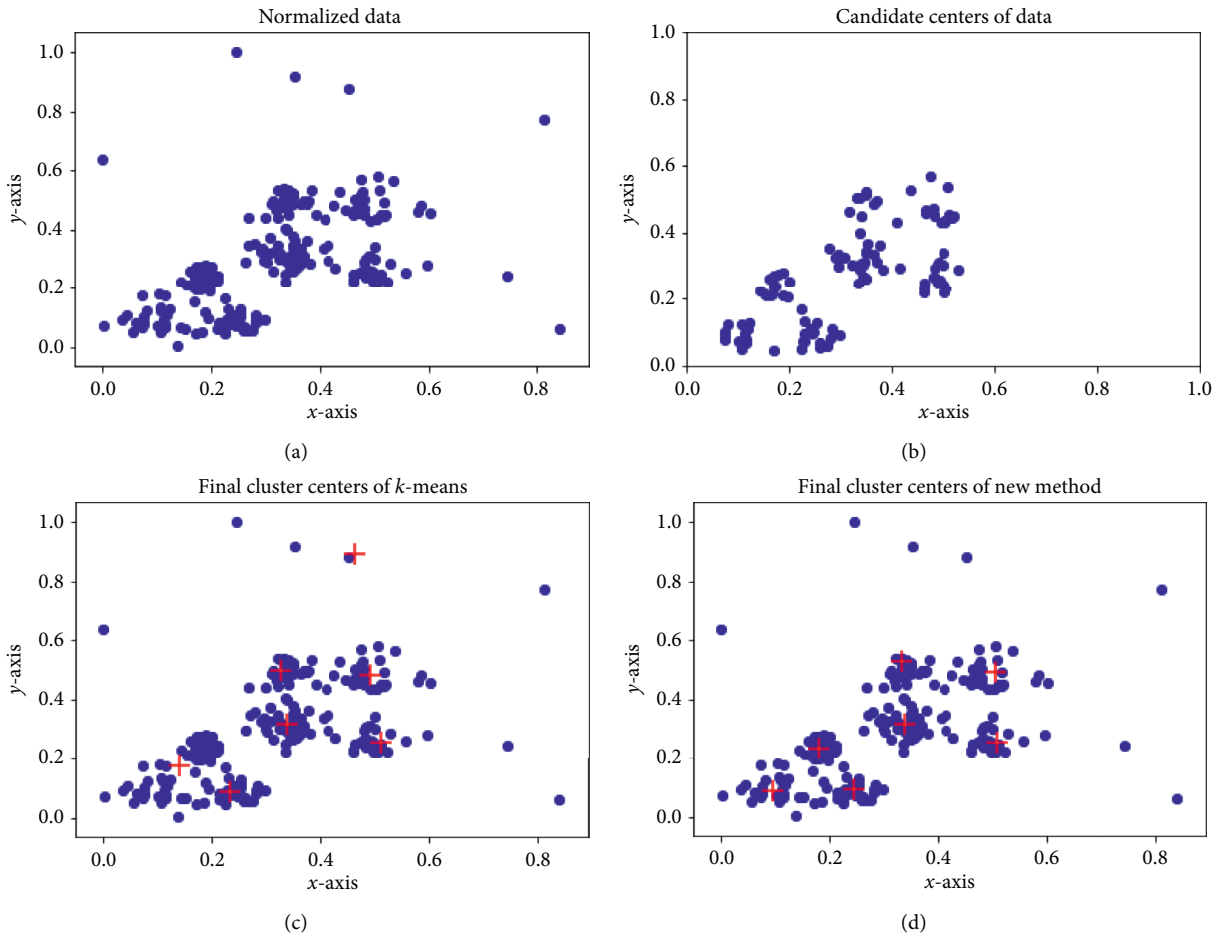


FIGURE 2: Experimental results on synthetic data set #2 (the red '+'s are the cluster centers). (a) Normalized data. (b) Selected subset. (c) Clustering centers of k -means algorithm. (d) Clustering centers of our proposed algorithm.

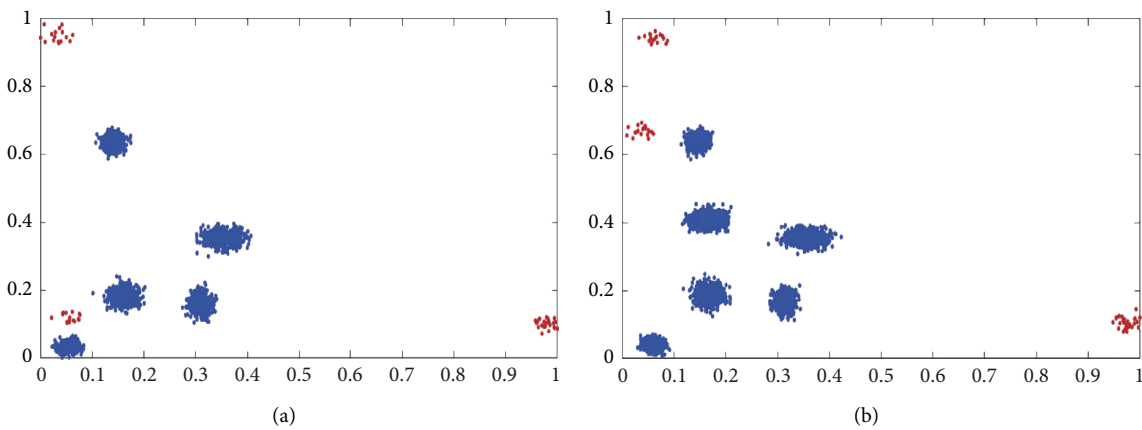


FIGURE 3: Continued.

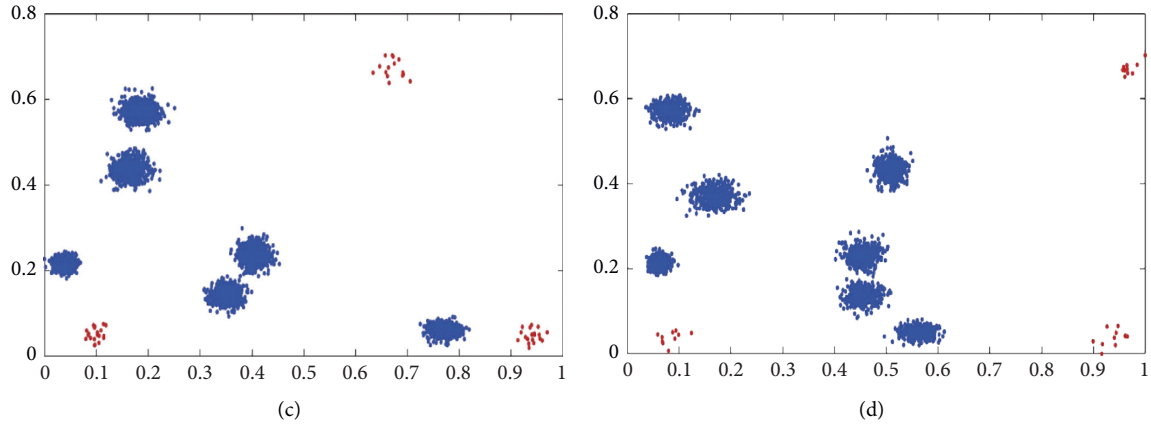


FIGURE 3: Two-dimensional illustrations corresponding to four synthetic data sets (the red data points are outliers). (a) Data set 3 (five clusters). (b) Data set 4 (six clusters). (c) Data set 5 (six clusters). (d) Data set 6 (seven clusters).

TABLE 1: Comparison between k -means clustering algorithm and our proposed clustering algorithm.

Data sets	N	t (%)	k	d	p	n_c	ARI_{kmeans}	ARI_{our}	$Time_{kmeans}$	$Time_{our}$
Synthetic #1	84	2.3	2	2	75	27	0.084	0.954	0.020	0.000
Synthetic #2	236	4.2	7	2	80	107	0.791	0.860	0.063	0.016
Synthetic #3	2557	2.2	5	3	92	761	0.774	0.972	0.047	0.031
Synthetic #4	3670	1.9	6	4	96	656	0.815	0.977	0.188	0.063
Synthetic #5	3655	1.5	6	5	96	573	0.816	0.982	0.313	0.078
Synthetic #6	2830	1.1	7	6	95	296	0.848	0.988	0.250	0.063
Iris	150	0	3	4	80	27	0.730	0.730	0.031	0.016
Iris*	152	1.3	3	4	82	26	0.531	0.743	0.031	0.016
Seeds	210	0	3	7	80	37	0.717	0.728	0.047	0.016
Seeds*	212	0.9	3	7	80	37	0.462	0.694	0.047	0.016
Wine	178	0	3	13	81	6	0.870	0.850	0.031	0.016
Wine*	180	1.1	3	13	81	10	0.365	0.882	0.031	0.016

*The real data set which includes two synthetic outliers as shown in our BaiduPan.

TABLE 2: Comparison between k -means clustering algorithm and our proposed clustering algorithm on large-scale data sets.

Data sets	N	k	d	p	n_c	ARI_{kmeans}	ARI_{our}	$Time_{kmeans}$	$Time_{our}$
Synthetic #7	22501	5	4	99	1413	1	1	0.858	0.718
Synthetic #8	50001	5	4	99.5	1148	1	1	1.732	1.248
Synthetic #9	60001	6	4	99.5	1257	1	1	2.262	1.466
Synthetic #10	60001	6	5	99.5	1038	1	1	2.387	1.810



FIGURE 4: Application of tyre inclusion identification. (a) Tyre #1. (b) Tyre #2.

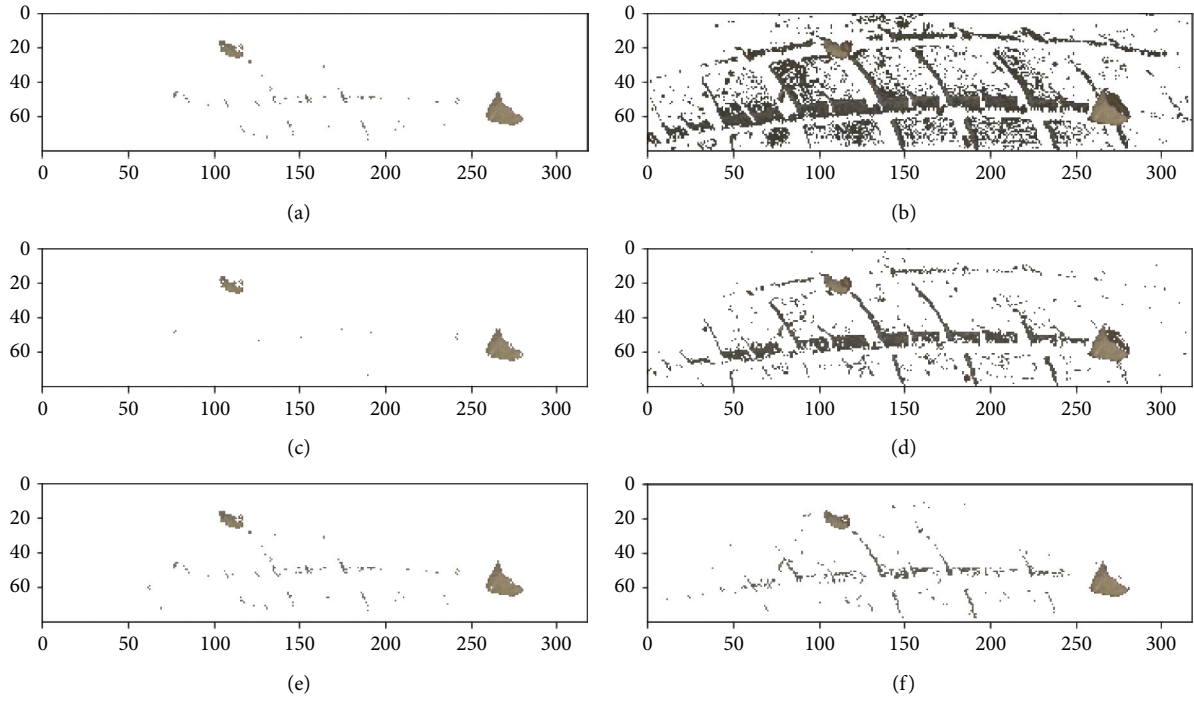


FIGURE 5: Clustering results on Tyre #1 (318×80 pixels, $p = 99$, $N = 25440$, and $n_c = 7696$). (a) Our algorithm with $k = 2$. (b) k -means with $k = 2$. (c) Our algorithm with $k = 3$. (d) k -means with $k = 3$. (e) Our algorithm with $k = 4$ (f) k -means with $k = 4$.

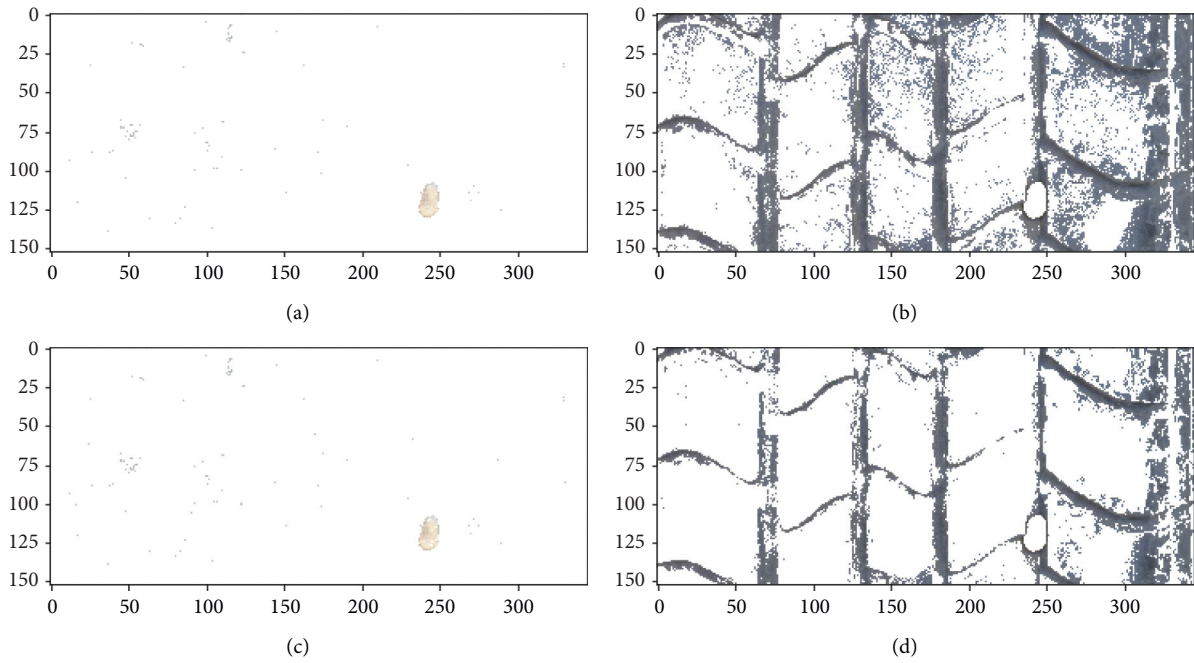


FIGURE 6: Continued.

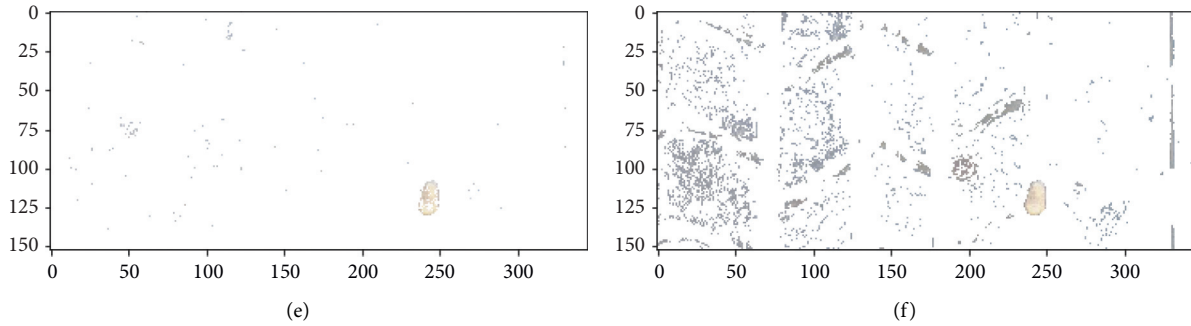


FIGURE 6: Clustering results on Tyre #2 (345×152 pixels, $p = 99$, $N = 52440$, and $n_c = 16852$). (a) Our algorithm with $k = 2$. (b) k -means with $k = 2$. (c) Our algorithm with $k = 3$. (d) k -means with $k = 3$. (e) Our algorithm with $k = 4$. (f) k -means with $k = 4$.

In addition, we provide a real application, i.e., the tyre inclusion identification, to validate the clustering performance of our proposed clustering algorithm. Figure 4 shows two tyres with different kinds of inclusions, where each picture includes 1027×768 pixels. Figures 5 and 6 present the clustering results of our proposed algorithm and k -means clustering algorithm on Tyre #1 and Tyre #2, respectively. In these figures, we can see that our proposed method can accurately identify the cluster centers without the disturbance of outliers. The inclusions can be clearly recognized by our proposed algorithm in the tyres, while the k -means clustering algorithm does not find the inclusions distinctly, e.g., Figures 6(b), 6(d), and 6(f) include not only the inclusions but also the tyre traces. Above all, the experimental results demonstrate the better clustering performance in comparison with the classical k -means clustering algorithm when handling the clustering tasks with the disturbance of outliers.

5. Conclusions and Future Work

In this paper, we proposed a robust two-stage k -means clustering algorithm which can accurately identify the cluster centers without the disturbance of outliers. As the direct application of the observation point mechanism of I-nice [18], we select a small subset from the original data set based on a set of nondegenerate observation points in the first stage. In the second stage, we use the k -means clustering algorithm to cluster the selected subset and make these cluster centers as the true cluster centers of the original data set. The theoretical analysis and experimental verification demonstrate the feasibility and effectiveness of proposed clustering algorithm. The future studies will be focused on three directions. First, we will try to use the k -nearest neighbors (k NN) method to improve the selection of observation points. Second, we will seek the real applications for the two-stage k -means clustering algorithm. Third, we will extend our proposed algorithm to cluster big data based on the random sample partition model [24].

Data Availability

The data used in our manuscript can be accessed by readers via our BaiduPan (<https://pan.baidu.com/s/1MfS8JfQdJLHYSlpZdndLUQ>) with the extraction code “p3mc.”

1MfS8JfQdJLHYSlpZdndLUQ) with the extraction code “p3mc.”

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper was supported by the National Key R&D Program of China (2017YFC0822604-2), Basic Research Foundation of Strengthening Police with Science and Technology of the Ministry of Public Security (2017GABJC09), and Scientific Research Foundation of Shenzhen University for Newly-introduced Teachers (2018060).

References

- [1] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, Berkeley, CA, USA, 1965.
- [2] E. W. Forgy, “Cluster analysis of multivariate data: efficiency versus interpretability of classifications,” *Biometrics*, vol. 21, pp. 768–769, 1965.
- [3] S. Lloyd, “Least squares quantization in PCM,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [4] F. Gibou and R. Fedkiw, “A fast hybrid k -means level set algorithm for segmentation,” in *Proceedings of the 4th Annual Hawaii International Conference on Statistics and Mathematics*, pp. 281–291, Honolulu, HI, USA, August 2005.
- [5] R. Herwig, A. J. Poustka, C. Muller et al., “Large-scale clustering of cdna-fingerprinting data,” *Genome Research*, vol. 9, no. 11, pp. 1093–1105, 1999.
- [6] D. Arthur and S. Vassilvitskii, “K-means++: the advantage of careful seeding,” in *Proceedings of the 18th Symposium on Discrete Algorithms*, pp. 1027–1035, New Orleans, LA, USA, January 2007.
- [7] R. J. Campello, D. Moulavi, A. Zimek et al., “Hierarchical density estimates for data clustering, visualization, and outlier detection,” *ACM Transactions on Knowledge Discovery from Data*, vol. 10, no. 1, 2015.
- [8] F. Jiang, G. Liu, J. Du, and Y. Sui, “Initialization of K-modes clustering using outlier detection techniques,” *Information Sciences*, vol. 332, pp. 167–183, 2016.

- [9] S. Salloum, J. Z. Huang, and Y. L. He, "Exploring and cleaning big data with random sample data blocks," *Journal of Big Data*, vol. 6, no. 1, p. 45, 2019.
- [10] Z. Huang, "Extensions to the k -means algorithm for clustering large data sets with categorical values," *Data Mining and Knowledge Discovery*, vol. 2, no. 3, pp. 283–304, 1998.
- [11] R. T. Ng and J. Han, "CLARANS: a method for clustering objects for spatial data mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 5, pp. 1003–1016, 2002.
- [12] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for k -medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.
- [13] D. Arlia and M. Coppola, "Experiments in parallel clustering with DBSCAN," *Euro-Par 2001 Parallel Processing*, vol. 2150, pp. 326–331, 2001.
- [14] M. Ester, H. P. Kriegel, J. Sander et al., "Density-based spatial clustering of applications with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, Portland, OR, USA, 1996.
- [15] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: why and how you should (still) use DBSCAN," *ACM Transactions on Database Systems*, vol. 42, no. 3, pp. 1–21, 2017.
- [16] C. Patil and I. Baidari, "Estimating the optimal number of clusters k in a dataset using data depth," *Data Science and Engineering*, vol. 4, no. 2, pp. 132–140, 2019.
- [17] D. Tanir and F. Nuriyeva, "On selecting the initial cluster centers in the k -means algorithm," in *Proceedings of 2017 IEEE International Conference on Application of Information and Communication Technologies*, pp. 1–5, Moscow, Russia, September 2017.
- [18] M. A. Masud, J. Z. Huang, C. Wei, J. Wang, I. Khan, and M. Zhong, "I-nice: a new approach for identifying the number of clusters and initial cluster centres," *Information Sciences*, vol. 466, pp. 129–151, 2018.
- [19] D. Cai and X. L. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE Transactions on Cybernetics*, vol. 45, no. 8, pp. 1669–1680, 2014.
- [20] D. Huang, C. D. Wang, J. Wu et al., "Ultra-scalable spectral clustering and ensemble clustering," *IEEE Transactions on Knowledge and Data Engineering*, p. 1, 2019.
- [21] W. Wang, J. Yang, and R. Muntz, "STING: A statistical information grid approach to spatial data mining," in *Proceedings of 1997 International Conference on Very Large Data Bases*, pp. 186–195, 1997.
- [22] M. Lichman, *UCI Machine Learning Repository*, University of California, Oakland, CA, USA, 2013.
- [23] I. Triguero, S. González, J. M. Moyano et al., "Keel 3.0: an open source software for multi-stage analysis in data mining," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, pp. 1238–1249, 2017.
- [24] S. Salloum, J. Z. Huang, and Y. He, "Random sample partition: a distributed data model for big data analysis," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 11, pp. 5846–5854, 2019.