Pattern Recognition Letters 90 (2017) 8–14

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

k-means clustering with outlier removal

Guojun Gan[a,*], Michael Kwok-Po Ng[b]

[a] Department of Mathematics, University of Connecticut, 341 Mansfield Road, Storrs, CT, 06269-1009, USA
[b] Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong, China

ARTICLE INFO

Article history:
Received 5 July 2016
Available online 8 March 2017

MSC:
62H30
68T10
91C20
62P10

Keywords:
Data clustering
k-means
Outlier detection

ABSTRACT

Outlier detection is an important data analysis task in its own right and removing the outliers from clusters can improve the clustering accuracy. In this paper, we extend the *k*-means algorithm to provide data clustering and outlier detection simultaneously by introducing an additional "cluster" to the *k*-means algorithm to hold all outliers. We design an iterative procedure to optimize the objective function of the proposed algorithm and establish the convergence of the iterative procedure. Numerical experiments on both synthetic data and real data are provided to demonstrate the effectiveness and efficiency of the proposed algorithm.

1. Introduction

The goal of data clustering is to identify homogeneous groups or clusters from a set of objects. In other words, data clustering aims to divide a set of objects into groups or clusters such that objects in the same cluster are more similar to each other than to objects from other clusters [3,11]. As an unsupervised learning process, data clustering is often used as a preliminary step for data analytics. For example, data clustering is used to identify the patterns hidden in gene expression data [30], to produce a good quality of clusters or summaries for big data to address the associated storage and analytical issues [9], to select representative insurance policies from a large portfolio in order to build metamodel models [12,14].

Many clustering algorithms have been developed in the past sixty years. Among these algorithms, the *k*-means algorithm is one of the oldest and most commonly used clustering algorithms [22,31]. Despite being used widely, the *k*-means algorithm has several drawbacks. One drawback is that it is sensitive to noisy data and outliers. For example, the *k*-means algorithm is not able to recover correctly the two clusters shown in Fig. 1(a) due to the outliers. As we can see from Fig. 1(b), three points were clustered incorrectly.

Motivated by Dave and Krishnapuram [7], we propose in this paper the KMOR (*k*-means with outlier removal) algorithm by extending the *k*-means algorithm for outlier detection. Dave and Krishnapuram [7] proposed to use an additional "cluster" for the fuzzy *c*-means algorithm to hold all outliers. In the KMOR algorithm, we use the same idea of introducing an additional "cluster" that contains all outliers. Given a desired number of clusters *k*, the KMOR algorithm partitions the dataset into $k+1$ groups, which include *k* clusters and a group of outliers that cannot fit into the *k* clusters. Unlike most existing clustering algorithms with outlier detection, the KMOR algorithm assigns all outliers into a group naturally during the clustering process.

The remaining part of this paper is organized as follows. In Section 2, we give a review of clustering algorithms that can detect outliers. In Section 3, we present the KMOR algorithm in detail. In Section 4, we demonstrate the performance of the KMOR algorithm with numerical results on both synthetic and real datasets. Section 5 concludes the paper with some remarks.

2. Related work

Kadam and Pund [27] and Aggarwal [2, Chapter 8] reviewed several approaches to detect outliers, including the cluster-based approach. Aggarwal [1] devoted a whole book to outlier analysis. Yu et al. [36] proposed the OEDP *k*-means algorithm by removing outliers from the dataset before applying the *k*-means algorithm. Aparna and Nair [5] proposed the CHB-K-Means algorithm by using a weighted attribute matrix to detect outliers. Jiang et al. [24] proposed two initialization methods for the *k*-modes algorithm to choose initial cluster centers that are not outliers.

* Corresponding author.
E-mail address: guojun.gan@uconn.edu (G. Gan).

http://dx.doi.org/10.1016/j.patrec.2017.03.008
0167-8655/

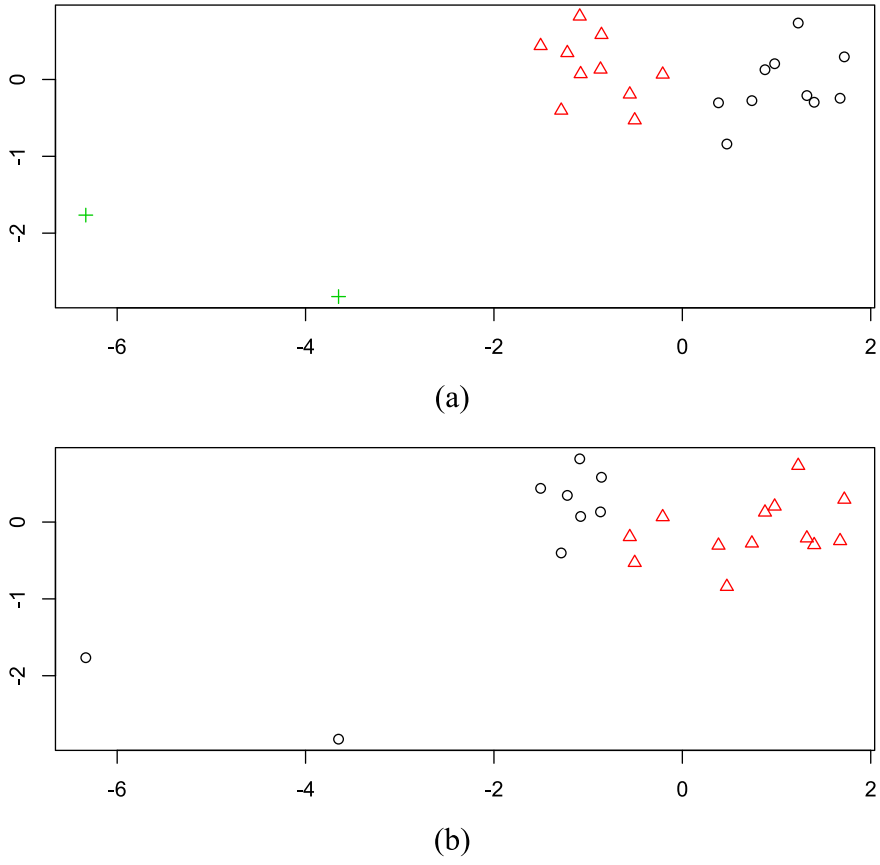

Fig. 1. An illustration showing that the *k*-means algorithm is sensitive to outliers. (a) A data set with two clusters and two outliers. The two clusters are plotted by triangles and circles, respectively. The two outliers are denoted by plus signs. (b) Two clusters found by the *k*-means algorithm. The two found clusters are plotted by triangles and circles, respectively.

Although much work has been done on outlier analysis, few of them perform clustering and detect outliers simultaneously. In this section, we focus on clustering methods with the built-in mechanism of outlier detection and give a review of those methods.

Jiang et al. [25] proposed a two-phase clustering algorithm for outlier detection. In the first phase, the *k*-means algorithm is modified to partition the data in such a way that a data point is assigned to be a new cluster center if the data point is far away from all clusters. In the second phase, a minimum spanning tree is constructed based on the cluster centers obtained from the first phase. Clusters in small sub trees are considered as outliers. He et al. [19] introduced the concept of cluster-based local outlier and designed a measure, called cluster-based local outlier factor (CBLOF), to identify such outliers.

Hautamäki et al. [18] proposed the ORC (Outlier Removal Clustering) algorithm to identify clusters and outliers from a dataset simultaneously. The ORC algorithm consists of two consecutive stages: the first stage is a purely *k*-means algorithm; the second stage iteratively removes the data points that are far away from their cluster centroids. Rehm et al. [34] defined outliers in terms of noise distance. The data points that are about the noise distance or further away from any other cluster centers get high membership degrees to the outlier cluster.

Jiang and An [26] also proposed a two-stage algorithm, called CBOD (Clustering Based Outlier Detection), to detect outliers from datasets. In the first stage, an one-pass clustering algorithm is applied to divide a dataset into hyper spheres with almost the same radius. In the second stage, outlier factors for all clusters obtained from the first stage are calculated and the clusters are sorted according to their outlier factors. Clusters with high outlier factors are considered outliers. Zhou et al. [38] proposed a three-stage *k*-means algorithm to cluster data and detect outliers. In the first stage, the fuzzy *c*-means algorithm is applied to cluster the data. In the second stage, local outliers are identified and the cluster centers are recalculated. In the third stage, certain clusters are merged and global outliers are identified. Zhang et al. [37] introduced a measure called Local Distance-based Outlier Factor (LDOF) to measure the outlier-ness of objects in scattered datasets. Pamula et al. [33] used the *k*-means algorithm to prune some points around the cluster centers and the LDOF measure to identify outliers from the remaining points. Jayakumar and Thomas [23] proposed an approach to detect outliers based on the Mahalanobis distance.

Ahmed and Naser [4] proposed the ODC (Outlier Detection and Clustering) algorithm to detect outliers. The ODC algorithm is a modified version of the *k*-means algorithm. In the ODC algorithm, a data point that is at least p times the average distance away from its centroid is considered as an outlier. Chawla and Gionis [6] proposed the *k*-means– algorithm to provide data clustering and outlier detection simultaneously. The *k*-means– algorithm requires two parameters: k and l, which specify the desired number of clusters and the desired number of top outliers, respectively.

Ott et al. [32] extended the facility location formulation to model the joint clustering and outlier detection problem and proposed a subgradient-based algorithm to solve the resulting optimization problem. The model requires pairwise distances of the dataset and the number of outliers as input. Whang et al. [35] proposed the NEO-*k*-means (Non-exhaustive Overlapping *k*-means) algorithm, which is also able to identify outliers during the clustering process.

Some of the aforementioned algorithms perform clustering and outlier detection in stages. In these algorithms, a clustering algorithm is used to divide the dataset into clusters and some measure is calculated for the data points based on the clusters to identify outliers. The ODC algorithm, the *k*-means– algorithm, and the NEO-*k*-means algorithm integrate outlier detection into the clustering process. However, data points that are removed as outliers during the iterative process of the ODC algorithm cannot be used as normal points again when the centroids are updated. Determining the parameters α and β in the NEO-*k*-means algorithm is time consuming.

3. The KMOR algorithm

In the KMOR algorithm, a data point that is at least $\gamma \times d_{avg}$ away from all the cluster centers is considered as an outlier, where γ is a multiplier and d_{avg} is the average distance calculated dynamically during the clustering process.

To describe the KMOR algorithm, let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a numerical dataset containing n data points, each of which is described by d numerical attributes. Let k be the desired number of clusters. Let $U = (u_{il})_{n\times(k+1)}$ be an $n \times (k+1)$ binary matrix (i.e., $u_{il} \in \{0, 1\}$) such that for each $i = 1, 2, \ldots, n$,

$$\sum_{l=1}^{k+1} u_{il} = 1. \tag{1}$$

The binary matrix U has $k+1$ columns. The last column of U is used to indicate whether a data point is an outlier. If $\mathbf{x}_i$ is an outlier, then $u_{i,k+1} = 1$. If $\mathbf{x}_i$ is a normal point, then $u_{i,k+1} = 0$. If $u_{i,k+1} = 0$, then $u_{il} = 1$ for some $l \in \{1, 2, \ldots, k\}$, where l is the index of the cluster to which $\mathbf{x}_i$ belongs. The binary matrix U is a partition matrix that divides the dataset X into $k+1$ groups, which include k normal clusters and one special "cluster" that contains the outliers.

The KMOR algorithm divides X into k clusters and a group of outliers by minimizing the following objective function

$$P(U, Z) = \sum_{i=1}^{n}\left(\sum_{l=1}^{k} u_{il}\|\mathbf{x}_i - \mathbf{z}_l\|^2 + u_{i,k+1}D(U, Z)\right), \tag{2}$$

subject to

$$\sum_{i=1}^{n} u_{i,k+1} \le n_0, \tag{3}$$

where $0 \le n_0 < n$ is a parameter, $Z = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k\}$ is a set of cluster centers, $\|\cdot\|$ is the L^2 norm and

$$D(U, Z) = \left(\frac{\gamma}{n - \sum_{j=1}^{n} u_{j,k+1}}\right)\sum_{l=1}^{k}\sum_{j=1}^{n} u_{j,l}\|\mathbf{x}_j - \mathbf{z}_l\|^2. \tag{4}$$

Here $\gamma \ge 0$ is also a parameter. The parameters n_0 and γ are used to control the number of outliers. How to select appropriate values for n_0 and γ is discussed in the end of this section. The first term of $P(U, Z)$ is employed in the standard *k*-means algorithm to put similar data points into a cluster. The second term of $P(U, Z)$ is used to check how to assign a point to be an outlier based on the average distance calculated from $D(U, Z)$.

The condition given in Eq. (3) limits the number of outliers to be at most n_0. In fact, the purpose of this condition is to make sure that

$$\sum_{i=1}^{n} u_{i,k+1} < n.$$

It is worth noting that this condition is necessary for the objective function to be nontrivial. Without the condition given in Eq. (3), the objective function reaches zero with $u_{i,k+1} = 1$, $u_{i,l} = 0$ for $i = 1, 2, \ldots, n$ and $l = 1, 2, \ldots, k$. In other words, the objective function without the condition is minimized when all data points are put into the group of outliers. When $n_0 = 0$, the condition given in Eq. (3) implies that $u_{i,k+1} = 0$ for $i = 1, 2, \ldots, n$. In this case, the KMOR objective function becomes the standard *k*-means objective function. When $0 < n_0 < n$, the KMOR objective function also becomes the standard *k*-means objective function when $\gamma \to \infty$. In this setting, the second term $u_{i,k+1}D(U, Z)$ is more dominant, the minimization procedure will favour $u_{i,k+1}$ to be zero, i.e., no points will be assigned to the group of outliers. Also it is obvious in the KMOR objective function that we do not allow $n_0 \ge n$ in order to prevent the algorithm from assigning all points to the group of outliers.

Like the *k*-means algorithm, the KMOR algorithm starts with a set of k initial cluster centers and then keeps updating U and Z until some stopping criterion is achieved. However, the objective function of the KMOR algorithm involves the interaction of the two set of variables: $u_{i,l}(l = 1, 2, \ldots, k)$ and $u_{i,k+1}$. As a result, the iterative process of the KMOR algorithm is different from that of the standard *k*-means algorithm. To describe the iterative process of the KMOR algorithm, we define

$$Q(U, V, Z) = \sum_{i=1}^{n}\left(\sum_{l=1}^{k} u_{i,l}\|\mathbf{x}_i - \mathbf{z}_l\|^2 + u_{i,k+1}D(V, Z)\right), \tag{5}$$

where $V = (v_{i,l})_{n\times(k+1)}$ is an $n \times (k+1)$ binary matrix that satisfies conditions given in (1) and (3). Comparing Eqs. (2) and (5), we have $Q(U, U, Z) = P(U, Z)$. According to $Q(U, V, Z)$, we can optimize Q by solving three subproblems: $Q(U, \cdot, \cdot)$, $Q(\cdot, V, \cdot)$ and $Q(\cdot, \cdot, Z)$ iteratively. The pseudo-code of the KMOR algorithm is given in Algorithm 1. For each subproblem, we have the following theorems to guarantee the optimality.

Algorithm 1: Pseudo-code of the KMOR algorithm, where σ and N_{max} are two parameters used to terminate the algorithm.

Input: X, k, γ, n_0, δ, N_{max}
Output: Optimal values of U and Z
Initialize $Z^{(0)} = \{\mathbf{z}_1^{(0)}, \mathbf{z}_2^{(0)}, \ldots, \mathbf{z}_k^{(0)}\}$ by selecting k points from X randomly;
Update $U^{(0)}$ by assigning $\mathbf{x}_i$ to its nearest center for $i = 1, 2, \ldots, n$;
$s \leftarrow 0$;
$P^{(0)} \leftarrow 0$;
while *True* **do**
  Update $U^{(s+1)}$ by minimizing $Q(U, U^{(s)}, Z^{(s)})$ according to Theorem 1;
  Update $Z^{(s+1)}$ by minimizing $Q(U^{(s+1)}, U^{(s+1)}, Z)$ according to Theorem 3;
  $s \leftarrow s + 1$;
  $P^{(s+1)} \leftarrow P\left(U^{(s+1)}, Z^{(s+1)}\right)$;
  if $\left|P^{(s+1)} - P^{(s)}\right| < \delta$ *or* $s \ge N_{max}$ **then**
    Break;
  end
end

Theorem 1. *Let $V = V^*$ and $Z = Z^*$ be fixed. Let $m_1, m_2, \ldots, m_n \in \{1, 2, \ldots, k\}$ such that*

$$d_{i,m_i} = \min_{1\le l\le k} d_{i,l},$$

where $d_{i,l} = \|\mathbf{x}_i - \mathbf{z}_l^*\|^2$, that is, m_i is the index of the center to which the point $\mathbf{x}_i$ is closest. Let $(i_1, i_2, \ldots, i_n)$ be a permutation of

$(1, 2, \ldots, n)$ *such that*

$d_{i_1,m_{i_1}} \geq d_{i_2,m_{i_2}} \geq \cdots \geq d_{i_n,m_{i_n}}$.

Define

$$O^* = \{i_1, i_2, \ldots, i_{n_0}\} \cap \{i \in \{1, 2, \ldots, n\} : d_{i,m_i} > D(V^*, Z^*)\},$$

where n_0 is a parameter of the KMOR algorithm. Then the binary matrix U^* that satisfies the conditions (1) and (3) and minimizes the function given in Eq. (5) is given as:

$$u^*_{i,l} = \begin{cases} 1, & \text{if } i \notin O^* \text{ and } l = m_i, \\ 0, & \text{if } i \in O^*, \end{cases} \tag{6}$$

for $i = 1, 2, \ldots, n$ and $l = 1, 2, \ldots, k$. For $l = k + 1$, we have $u^*_{i,k+1} = 1 - \sum_{s=1}^{k} u^*_{is}$.

Proof. We only need to show that for any binary matrix U satisfying the conditions (1) and (3), we have

$$Q(U^*, V^*, Z^*) \leq Q(U, V^*, Z^*).$$

To do that, we let U be an arbitrary binary matrix that satisfies conditions (1) and (3) and let

$$O = \{i \in \{1, 2, \ldots, n\} : u_{i,k+1} = 1\}$$

and $t = |O|$. Let

$$a^*_i = \sum_{l=1}^{k} u^*_{il} d_{il} + u^*_{i,k+1} D(V^*, Z^*), \quad i = 1, 2, \ldots, n$$

and

$$a_i = \sum_{l=1}^{k} u_{i,l} d_{il} + u_{i,k+1} D(V^*, Z^*), \quad i = 1, 2, \ldots, n.$$

Then we have $a^*_{i_j} = D(V^*, Z^*)$ for $j = 1, 2, \ldots, t^*$ and $a^*_{i_j} = d_{i_j,m_{i_j}}$ for $j = t^* + 1, \ldots, n$, where $t^* = |O^*|$.

Let $(s_1, s_2, \ldots, s_n)$ be a permutation of $(1, 2, \ldots, n)$ such that $a_{s_j} = D(V^*, Z^*)$ for $j = 1, 2, \ldots, t$ and

$$a_{s_{t+1}} \geq a_{s_{t+2}} \geq \cdots \geq a_{s_n}.$$

Let us first consider the case when $t = t^*$. In this case, we have $a^*_{i_j} = a_{s_j} = D(V^*, Z^*)$ for $j = 1, 2, \ldots, t$ and $a^*_{i_j} \leq a_{s_j}$ for $j = t + 1, t + 2, \ldots, n$. Hence we have

$$Q(U^*, V^*, Z^*) = \sum_{j=1}^{n} a^*_{i_j} \leq \sum_{j=1}^{n} a_{s_j} = Q(U, V^*, Z^*).$$

Now let us consider the case when $t < t^*$. In this case, we have $a^*_{i_j} = a_{s_j} = D(V^*, Z^*)$ for $j = 1, 2, \ldots, t$. For $j = t + 1, t + 2, \ldots, t^*$, we have

$$a_{s_j} = \sum_{l=1}^{k} u_{s_j l} d_{s_j l} \geq d_{s_j,m_{s_j}} > D(V^*, Z^*) = a^*_{i_j}.$$

For $j = t^* + 1, \ldots, n$, we have $a^*_{i_j} \leq a_{s_j}$. Hence we have

$$Q(U^*, V^*, Z^*) = \sum_{j=1}^{n} a^*_{i_j} \leq \sum_{j=1}^{n} a_{s_j} = Q(U, V^*, Z^*).$$

For the case when $t > t^*$, we have $t^* < t \leq n_0$ because U satisfies the condition (3). In this case, we have $a^*_{i_j} = a_{s_j} = D(V^*, Z^*)$ for $j = 1, 2, \ldots, t^*$. For $j = t^* + 1, \ldots, t$, we have

$$a^*_{i_j} = d_{i_j,m_{i_j}} \leq D(V^*, D^*) = a_{s_j}.$$

For $j = t + 1, t + 2, \ldots, n$, we have $a^*_{i_j} \leq a_{s_j}$. In this case, we also have

$$Q(U^*, V^*, Z^*) = \sum_{j=1}^{n} a^*_{i_j} \leq \sum_{j=1}^{n} a_{s_j} = Q(U, V^*, Z^*).$$

This completes the proof. □

According to Theorem 1, the assignment of a point to a cluster can be determined similar to that of the standard k-means algorithm except we remove outliers in the clustering process.

Theorem 2. *Let $V = V^*$ and $Z = Z^*$ be fixed. Let U^* be the binary matrix defined in Eq. (6). Then by Theorem 1, we know that U^* satisfies the conditions (1) and (3) and minimizes the function given in Eq. (5). Suppose that*

$$\sum_{i=1}^{n} u^*_{i,k+1} = \sum_{i=1}^{n} v^*_{i,k+1}.$$

Then

$$Q(U^*, U^*, Z^*) \leq Q(U^*, V^*, Z^*).$$

Proof. We only need to show that

$$D(U^*, Z^*) \leq D(V^*, Z^*). \tag{7}$$

By Theorem 1, we know that U^* is a binary matrix that satisfies conditions (1) and (3) and minimizes $Q(U, V^*, Z^*)$. Hence we have

$$Q(U^*, V^*, Z^*) \leq Q(V^*, V^*, Z^*)$$

or

$$\sum_{i=1}^{n} \left(\sum_{l=1}^{k} u^*_{i,l} \|\mathbf{x}_i - \mathbf{z}^*_l\|^2 + u^*_{i,k+1} D(V^*, Z^*) \right) \leq \sum_{i=1}^{n} \left(\sum_{l=1}^{k} v^*_{i,l} \|\mathbf{x}_i - \mathbf{z}^*_l\|^2 + v^*_{i,k+1} D(V^*, Z^*) \right). \tag{8}$$

From the assumption that $\sum_{i=1}^{n} u^*_{i,k+1} = \sum_{i=1}^{n} v^*_{i,k+1}$ and Eq. (8), we get

$$\sum_{i=1}^{n} \sum_{l=1}^{k} u^*_{i,l} \|\mathbf{x}_i - \mathbf{z}^*_l\|^2 \leq \sum_{i=1}^{n} \sum_{l=1}^{k} v^*_{i,l} \|\mathbf{x}_i - \mathbf{z}^*_l\|^2.$$

Eq. (7) follows from the above inequality, the assumption, and the definition of $D(V, Z)$. This completes the proof. □

According to Theorem 2, we can set V^* equal to U^*, and guarantee that the objective function value is always non-increasing.

Theorem 3. *Let $U = U^*$ and $V = U^*$ be fixed. Then the cluster centers Z^* that minimizes the function (5) is given by*

$$z^*_{l,s} = \frac{\sum_{i=1}^{n} u^*_{i,s} x_{i,s}}{\sum_{i=1}^{n} u^*_{i,s}} \tag{9}$$

for $l = 1, 2, \ldots, k$ and $s = 1, 2, \ldots, d$, where $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \cdots, x_{i,d}]$.

Proof. By combining Eqs. (4) and (5), we get

$$Q(U^*, U^*, Z) = \left[1 + \frac{\gamma \sum_{j=1}^{n} u^*_{j,k+1}}{n - \sum_{j=1}^{n} u^*_{j,k+1}} \right] \sum_{i=1}^{n} \sum_{l=1}^{k} u^*_{i,l} \|\mathbf{x}_i - \mathbf{z}_l\|^2.$$

Minimizing the above equation with respect to Z is equivalent to minimizing the following function

$$f(Z) = \sum_{i=1}^{n} \sum_{l=1}^{k} u^*_{i,l} \|\mathbf{x}_i - \mathbf{z}_l\|^2.$$

Taking derivative of the above equation with respect to z_{ls} and equating the derivative to zero lead to the result. This completes the proof. □

According to Theorem 3, we see that the update of cluster centers is the same as that in the standard k-means algorithm.

By using the results in Theorems 1–3, we can show that the KMOR algorithm converges. In particular, we have

$$P(U^{(s+1)}, Z^{(s+1)}) = Q(U^{(s+1)}, U^{(s+1)}, Z^{(s+1)})$$
$$\le Q(U^{(s+1)}, U^{(s+1)}, Z^{(s)}) \le Q(U^{(s+1)}, U^{(s)}, Z^{(s)})$$
$$\le Q(U^{(s)}, U^{(s)}, Z^{(s)}) = P(U^{(s)}, Z^{(s)}).$$

We see that the objective is non-increasing. As U is finite and the objective function value is bounded below by zero, the algorithm will terminate as the objective function value is not changed after a finite number of iterations.

As shown in Algorithm 1, the KMOR algorithm requires three main parameters k, n_0, and γ. The first parameter k is the desired number of clusters. The second parameter n_0 is the maximum number of outliers. The purpose of this parameter is to prevent the algorithm from assigning all the points to the group of outliers. The third parameter γ is used to classify normal points and outliers. In general, when the value of γ increases, the number of outliers decreases. The two additional parameters δ and N_{max} are used to terminate the algorithm.

The parameters n_0 and γ are used together to control the number of outliers. If we know the percentage of outliers in a dataset, then we can set n_0 to that number and set γ to be 1 so that the algorithm will identify a group of n_0 outliers and divide the remaining data points into k clusters. If we do not know the percentage of outliers in a dataset, then we can set n_0 to a reasonable large number (e.g., $0.5n$) and set γ appropriately to capture the outliers. For example, we can set γ in such a way that the scaled average distance $D(U, Z)$ is approximately equal to the maximum of $\sum_{l=1}^{k} u_{i,l}\|\mathbf{x}_i - \mathbf{z}_l\|^2$ $(1 \le i \le n)$. Suppose that

$$L = \max_{1\le i\le n} \sum_{l=1}^{k} u_{i,l}\|\mathbf{x}_i - \mathbf{z}_l\|$$

and $\sum_{l=1}^{k} u_{i,l}\|\mathbf{x}_i - \mathbf{z}_l\|$ $(1 \le i \le n)$ are uniformly distributed on [0, L]. Then such γ can be derived from the following relation:

$$\frac{\gamma}{n_1}\sum_{s=1}^{n_1}\left(\frac{s}{n_1}L\right)^2 \approx L^2,$$

where $n_1 = n - \sum_{j=1}^{n} u_{j,k+1}$. Noting that

$$\sum_{s=1}^{n_1} s^2 = \frac{n_1^3}{3} + \frac{n_1^2}{2} + \frac{n_1}{6},$$

we get $\gamma \approx 3$.

If we do not know the percentage of outliers in a dataset, setting $\gamma = 3$ is a good initial guess. To apply the KMOR algorithm, we use the following default values for parameters: $\gamma = 3$, $n_0 = 0.1n$, $\delta = 10^{-6}$, and $N_{max} = 100$.

4. Numerical experiments

In this section, we demonstrate the performance of the KMOR algorithm using both synthetic and real datasets. We shall compare the KMOR algorithm with the ODC algorithm [4], the k-mean– algorithm [6], and the NEO-k-means algorithm [35], which are clustering algorithms that perform clustering and outlier detection simultaneously.

To measure the performance of the KMOR algorithm, we use the following two measures: the corrected Rand index [15,21] and the distance of a classifier on the Receiver Operating Curve graph from the perfect classifier [4]. The first measure, denoted by R, is used to measure the overall accuracy of the clustering algorithm in terms of clustering and outlier detection. The value of R ranges from −1 to 1. A value of 1 indicates a perfect agreement between the two partitions; while a negative value indicates agreement by chance. The second measure, denoted by M_E, is used to measure the performance of the clustering algorithm in terms of outlier detection. The measure M_E ranges from 0 to $\sqrt{2}$. A smaller value of M_E indicates a better result.

Table 1
Average statistics of 100 runs of the four algorithms on synthetic datasets. The runtime is measured in seconds. (a) Results on the first synthetic dataset with $k = 2$. (b) Results on the second synthetic dataset with $k = 7$.

	KMOR	ODC	k-means–	NEO-k-means
R	0.91	0.87	0.32	0.34
M_E	0.05	0.16	0.95	0.94
Outliers	6.14	5.04	53	52.27
Runtime	0.003	0.003	0.019	0.015

(a)

	KMOR	ODC	k-means–	NEO-k-means
R	0.562	0.782	0.291	0.292
M_E	0.707	1	0.709	0.717
Outliers	272.37	0	408	407.43
Runtime	0.029	0.01	0.044	0.078

(b)

4.1. Experiments on synthetic data sets

To show that the proposed algorithm works, we generated two synthetic datasets with some outliers. The two synthetic datasets are shown in Fig. 2. The first synthetic dataset contains 106 data points, including 2 clusters and 6 outliers. The second synthetic dataset contains 816 data points, including 7 clusters and 16 outliers.

The KMOR algorithm has two main parameters: n_0 and γ. The parameter n_0 specifies the maximum number of outliers. The parameter γ specifies the multiplier of the average squared distance for outlier detection. In our experiments, we used $n_0 = 0.5n$ and $\gamma = 3$ as discussed in the end of Section 3. For the ODC algorithm, the parameter p is used to control the number of outliers. A smaller value of p leads to more outliers. In our experiments, we set $p = 6$ for ODC. For the k-mean– algorithm, the parameter l refers to the top number of outliers. In our experiment, we set $l = 0.5n$ for k-means– that is the same as the parameter n_0 in KMOR. In the NEO-k-means algorithm, α captures the degree of overlap and βn is the maximum number of outliers. For comparison purpose, we set $\alpha = 0$ and $\beta = 0.5$ for NEO-k-means. Since all four algorithms can be affected by the cluster center initialization problem, we run each algorithm 100 times with different initial cluster centers selected randomly from the datasets.

Table 1 summarizes the average accuracy and runtime when the four algorithms are applied to the two synthetic datasets. From Table 1(a), we see that the KMOR algorithm performs the best among the four algorithms in terms of overall accuracy as measured by the corrected Rand index. The k-means– algorithm and the NEO-k-means algorithm produced similar results. In addition, the KMOR algorithm identified 6.14 outliers on average, which is close to the actual number of outliers in the first synthetic dataset. The average number of outliers identified by the k-means– algorithm and the NEO-k-means algorithm is close to the specified number of outliers.

Table 1(b) shows the average statistics of 100 runs of the four algorithms on the second synthetic dataset. For the second synthetic dataset, the ODC algorithm achieved the best overall performance. However, the ODC algorithm did not identify any outliers as the average number of identified outliers is zero. Again, the number of outliers identified by the k-means– algorithm and the NEO-k-means algorithm is close to the specified number of outliers. The KMOR algorithm identified 272.37 outliers on aver-

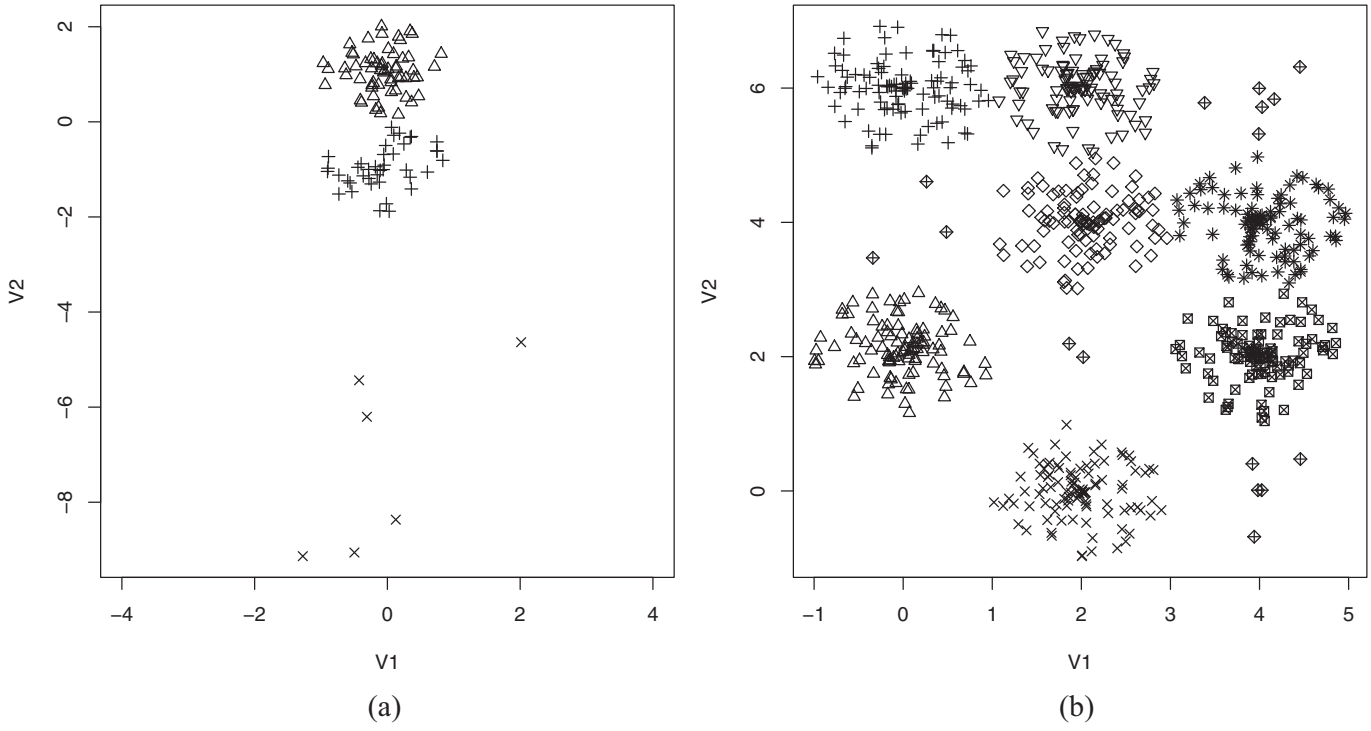

Fig. 2. Two synthetic datasets with outliers.

age, which is less than the specified number of outliers n_0. We can increase γ to decrease the number of outliers.

In term of runtime, the k-means– algorithm and the NEO-k-means algorithm were slower than the KMOR algorithm and the ODC algorithm. For the second synthetic dataset, the NEO-k-means algorithm is the slowest because it needs to sort the distances between all points and all cluster centers.

4.2. Experiments on real data sets

To test the performance of the proposed algorithm, we also obtained real datasets from UCI machine learning Repository [10]: the WBC dataset and the Shuttle dataset. The WBC dataset contains 699 records, each of which is described by 9 numerical attributes. The WBC dataset contains 2 clusters: malignant and benign. The benign cluster contains 458 records and the malignant cluster contains 241 records. We treat the benign records as normal and the malignant records are outliers. The WBC dataset was used to study outlier detection by He et al. [19], Jiang and An [26], and Duan et al. [8]. The shuttle dataset contains 58,000 records, which are described by 9 numerical features. The shuttle dataset consists of a training set and a test set. We use the training set in our experiments. The training set contains 43,500 records and 7 classes. The largest three classes contain 99.57% of the points. We treat the points in the three largest classes as normal points and points in the reset four classes as outliers. The shuttle dataset was used to study outliers by Chawla and Gionis [6].

We applied KMOR, ODC, k-means–, and NEO-k-means to the real datasets 100 times with different initial cluster centers, which are selected randomly from the datasets. The average corrected Rand index, the average M_E measure, the average number of outliers, and the average runtime of these 100 runs on the real datasets are summarized in Table 2(a) and (b). For the WBC dataset, we used the parameter values mentioned before. Since the shuttle dataset is a large dataset, we used a larger value for γ and a smaller value for n_0 in order to control the number of outliers. In particular, we use $n_0 = 0.1n$ and $\gamma = 9$ for KMOR. Similar to the way we select parameter values for the synthetic datasets, we use $p = 9$ for ODC, $l = 0.1n$ for k-means–,and $\alpha = 0$ and $\beta = 0.1$for NEO-k-means.

Table 2
Average statistics of 100 runs of the four algorithms on real datasets. (a) Results on the WBC dataset with $k = 1$. (b) Results on the Shuttle dataset with $k = 3$.

	KMOR	ODC	k-means–	NEO-k-means
R	0.695	0	0.477	0.481
M_E	0.127	1	0.236	0.234
Outliers	299	0	349	348
Runtime	0.023	0.009	0.037	0.021

(a)

	KMOR	ODC	k-means–	NEO-k-means
R	0.46	0.44	0.36	0.36
M_E	0.99	1.002	1.009	1.009
Outliers	1106.7	72.07	4350	4349.96
Runtime	1.262	1.646	2.689	5.26

(b)

From Table 2(a), we see that each of the four algorithms produced identical clustering results for the 100 runs. Since the desired number of clusters is 1 for the WBC dataset, all the 100 runs produced the same results. The standard deviations of the corrected Rand index, the M_E measure, and the number of outliers are zero. The runtime of each run was different due to the operating system. By comparing the average corrected Rand indices and the average classifier distances, we see that the KMOR algorithm achieved the best performance in terms of overall accuracy and outlier detection.

Table 2(b) summarizes the performance of the four algorithms on the shuttle dataset. From this table we see that the KMOR algorithm achieved the best performance in terms of overall accuracy. This test shows the KMOR algorithm is able to converge fast for large datasets. The NEO-k-means algorithm was the slowest algorithm due to the fact that it needs to sort nk distances for record assignments.

In summary, the tests on both synthetic data and real data have shown that the KMOR algorithm is able to cluster data and detect outliers simultaneously. In addition, the tests also show that the KMOR algorithm is able to outperform the ODC algorithm, the k-means– algorithm, and the NEO-k-means algorithm in terms of overall accuracy and outlier detection. For large datasets, the KMOR algorithm is also able to outperform other algorithms in term of speed.

5. Conclusions

Both clustering and outlier detection are important data analysis tasks. In this paper, we proposed the KMOR algorithm by extending the k-means algorithm to provide data clustering and outlier detection simultaneously. In the KMOR algorithm, two parameters n_0 and γ are used to control the number of outliers. The parameter n_0 is the maximum number of outliers the proposed algorithm will produce regardless the value of γ. For fixed n_0, a larger value of γ leads to less number of outliers. We can also estimate the two parameters within the algorithm. For example, we can follow the approach proposed in [35] by running the traditional k-means algorithm on a dataset to estimate n_0 and γ.

We compared the KMOR algorithm, the ODC algorithm [4], the k-means– algorithm [6], and the NEO-k-means algorithm [35]. The experiments on both synthetic data and real data have shown that the KMOR algorithm is able to cluster data and detect outliers at the same time. The tests have also shown that the KMOR algorithm was able to outperform other algorithms in terms of accuracy and runtime. Since outlier detection in the KMOR algorithm is natural part of the clustering process, points can move between normal clusters and the outlier cluster. In the ODC algorithm, however, points assigned to the outlier cluster cannot be reassigned to a normal cluster.

In future, we would like to extend the KMOR algorithm in the following directions. First, we would like to investigate other ways to control the number of outliers. Currently we control the number of outliers by the parameter n_0. Second, we would like to extend the KMOR algorithm for subspace clustering [13,15–17,20]. Currently the KMOR algorithm was not designed to identify clusters embedded in subspaces of the original data space. Finally, it is also interesting to investigate how to select an appropriate value for the parameter k required by the KMOR algorithm [28,29]. In the current version of the algorithm, we assume that k is given.

References

[1] C.C. Aggarwal, Outlier Analysis, Springer, New York, NY, 2013.
[2] C.C. Aggarwal, Data Mining: The Textbook, Springer, New York, NY, 2015.
[3] C.C. Aggarwal, C.K. Reddy (Eds.), Data Clustering: Algorithms and Applications, CRC Press, Boca Raton, FL, USA, 2013.
[4] M. Ahmed, A. Naser, A novel approach for outlier detection and clustering improvement, in: Proceedings of the 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2013, pp. 577–582.
[5] K. Aparna, M.K. Nair, Computational Intelligence in Data Mining, vol. 2, Springer, pp. 25–35.
[6] S. Chawla, A. Gionis, k-means: a unified approach to clustering and outlier detection, SIAM, pp. 189–197.
[7] R. Dave, R. Krishnapuram, Robust clustering methods: a unified view, IEEE Trans. Fuzzy Syst. 5 (2) (1997) 270–293.
[8] L. Duan, L. Xu, Y. Liu, J. Lee, Cluster-based outlier detection, Ann. Oper. Res. 168 (1) (2009) 151–168.
[9] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, A. Y.Zomaya, I. Khalil, S. Foufou, A. Bouras, A survey of clustering algorithms for big data: taxonomy amp; empirical analysis, IEEE Trans. Emerg. Top Comput. PP (99) (2014). 1–1
[10] A. Frank, A. Asuncion, UCI machine learning repository, 2010. Available at http://archive.ics.uci.edu/ml.
[11] G. Gan, Data Clustering in C++: An Object-Oriented Approach, Data Mining and Knowledge Discovery Series, Chapman & Hall/CRC Press, Boca Raton, FL, USA, 2011.
[12] G. Gan, Application of data clustering and machine learning in variable annuity valuation, Insurance: Math. Econ. 53 (3) (2013) 795–801.
[13] G. Gan, K. Chen, A soft subspace clustering algorithm with log-transformed distances, Big Data and Inf. Anal. 1 (1) (2016) 93–109.
[14] G. Gan, S. Lin, Valuation of large variable annuity portfolios under nested simulation: a functional data approach, Insurance: Math. Econ. 62 (2015) 138–150.
[15] G. Gan, M.K.-P. Ng, Subspace clustering using affinity propagation, Pattern Recognit. 48 (4) (2015) 1451–1460.
[16] G. Gan, M.K.-P. Ng, Subspace clustering with automatic feature grouping, Pattern Recognit. 48 (11) (2015) 3703–3713.
[17] G. Gan, J. Wu, Z. Yang, A fuzzy subspace algorithm for clustering high dimensional data, in: X. Li, S. Wang, Z. Dong (Eds.), Lecture Notes in Artificial Intelligence, 4093, Springer-Verlag, 2006, pp. 271–278.
[18] V. Hautamäki, S. Cherednichenko, I. Kärkkäinen, T. Kinnunen, P. Fränti, Improving k-means by outlier removal, in: Proceedings of the 14th Scandinavian Conference on Image Analysis, SCIA'05, 2005, pp. 978–987.
[19] Z. He, X. Xu, S. Deng, Discovering cluster-based local outliers, Pattern Recognit. Lett. 24 (9–10) (2003) 1641–1650.
[20] 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.
[21] L. Hubert, P. Arabie, Comparing partitions, J. Classification 2 (1985) 193–218.
[22] A. Jain, Data clustering: 50 years beyond k-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.
[23] G.S.D.S. Jayakumar, B.J. Thomas, A new procedure of clustering based on multivariate outlier detection, J. Data Sci. 11 (2013) 69–84.
[24] F. Jiang, G. Liu, J. Du, Y. Sui, Initialization of k-modes clustering using outlier detection techniques, Inf. Sci. 332 (2016) 167–183.
[25] M. Jiang, S. Tseng, C. Su, Two-phase clustering process for outliers detection, Pattern Recognit. Lett. 22 (6–7) (2001) 691–700.
[26] S.-Y. Jiang, Q. An, Clustering-based outlier detection method, in: Fifth International Conference on Fuzzy Systems and Knowledge Discovery, 2, 2008, pp. 429–433.
[27] N.V. Kadam, M.A. Pund, Joint approach for outlier detection, Int. J. Comput. Sci. Appl. 6 (2) (2013) 445–448.
[28] S.-S. Kim, Variable selection and outlier detection for automated k-means clustering, Commun. Statis. Appl. Methods 22 (1) (2015) 55–67.
[29] D. Lei, Q. Zhu, J. Chen, H. Lin, P. Yang, Information Engineering and Applications: International Conference on Information Engineering and Applications (IEA 2011), Springer London, London, pp. 363–372.
[30] J.D. MacCuish, N.E. MacCuish, Clustering in Bioinformatics and Drug Discovery, CRC Press, Boca Raton, FL, 2010.
[31] J. Macqueen, Some methods for classification and analysis of multivariate observations, in: L. LeCam, J. Neyman (Eds.), Proceedings of the 5th Berkeley Symposium on Mathematical Statistics andProbability, 1, University of California Press, Berkely, CA, USA, 1967, pp. 281–297.
[32] L. Ott, L. Pang, F.T. Ramos, S. Chawla, On integrated clustering and outlier detection, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Weinberger (Eds.), Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014, pp. 1359–1367.
[33] R. Pamula, J. Deka, S. Nandi, An outlier detection method based on clustering, in: Second International Conference on Emerging Applications of Information Technology, 2011, pp. 253–256.
[34] F. Rehm, F. Klawonn, R. Kruse, A novel approach to noise clustering for outlier detection, Soft Comput. 11 (5) (2007) 489–494.
[35] J. Whang, I.S. Dhillon, D. Gleich, Non-exhaustive, overlapping k-means, in: SIAM International Conference on Data Mining (SDM), 2015.
[36] Q. Yu, Y. Luo, C. Chen, X. Ding, Outlier-eliminated k-means clustering algorithm based on differential privacy preservation, Applied Intelligence (2016) 1–13.
[37] K. Zhang, M. Hutter, H. Jin, A new local distance-based outlier detection approach for scattered real-world data, in: Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, PAKDD '09, 2009, pp. 813–822.
[38] Y. Zhou, H. Yu, X. Cai, A novel k-means algorithm for clustering and outlier detection, in: Second International Conference on Future Information Technology and Management Engineering, 2009, pp. 476–480.