

A novel outlier cluster detection algorithm without top-n parameter



Jinlong Huang, Qingsheng Zhu*, Lijun Yang, DongDong Cheng, Quanwang Wu

Chongqing Key Lab. of Software Theory and Technology, College of Computer Science, Chongqing University, Chongqing 400044, China

ARTICLE INFO

Article history:

Received 12 August 2016

Revised 6 January 2017

Accepted 7 January 2017

Available online 9 January 2017

Keywords:

Outlier detection

Outlier clusters

Top-n problem

Mutual neighbor

ABSTRACT

Outlier detection is an important task in data mining with numerous applications, including credit card fraud detection, video surveillance, etc. Outlier detection has been widely focused and studied in recent years. The concept about outlier factor of object is extended to the case of cluster. Although many outlier detection algorithms have been proposed, most of them face the top-n problem, i.e., it is difficult to know how many points in a database are outliers. In this paper we propose a novel outlier cluster detection algorithm called ROCF based on the concept of mutual neighbor graph and on the idea that the size of outlier clusters is usually much smaller than the normal clusters. ROCF can automatically figure out the outlier rate of a database and effectively detect the outliers and outlier clusters without top-n parameter. The formal analysis and experiments show that this method can achieve good performance in outlier detection.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Outlier detection is an important data mining activity with numerous applications, including credit card fraud detection, discovery of criminal activities in electronic commerce, video surveillance, weather prediction, and pharmaceutical research [1–9].

An outlier is an observation that deviates so much from other observations so that it arouses that it is generated by a different mechanism [8]. In many fields, outliers are more important than the normal data, as they may demonstrate either deviant behavior or the beginning of a new pattern, and may cause damage to the user. At present, the studies on outlier detection is very active. Many outlier detection algorithms have been proposed. Outlier detection algorithms can be roughly divided into distribution-based methods, depth-based methods, distance-based methods, density-based methods and clustering-based methods etc.

Each type of the outlier detection algorithms has its advantages and disadvantage. In distribution-based methods, the observations that deviate from a standard distribution are considered as outliers [7]. Hence distribution-based methods can simply and effectively detect out the outliers, if we know the distribution of the datasets. However, distribution-based methods are not applicable to datasets that are multidimensional or where the distribution is unknown. The depth-based [10,11] methods can address this problem. Depth-based methods relies on the computation of different layers of k-d

convex hulls. In this way, outliers are objects in the outer layer of these hulls. However, the efficiency of depth-based algorithms is low on 4-dimensional or more than 4-dimensional datasets.

The distance-based algorithms are widely used for its effectiveness and simplification. In paper [4], a distance-based outlier is described as the object that with pct% of the objects in database having a distance of more than d_{\min} away from it. However, distance-based algorithms don't take into account the changes of local density, so distance-based algorithms can only detect the global outliers, fail to detect the local outliers. The density-based methods can solve this problem well. Many density-based outlier detection algorithms have been proposed, such as LOF [12], INFLO [13] and INS [14].

However, all of the above outlier detection algorithms only regard the outlier as a single point that deviates from a certain cluster. Therefore, in order to analyze the outliers, researchers must further process these outliers after outlier detection, such as paper [15] does. In order to solve the problem of outlier cluster, many cluster-based outlier detection algorithms are proposed. However, although cluster-based outlier detection algorithm can detect the outlier clusters, these methods need too many parameters. For instance, Duan et al. proposed a cluster-based outlier detection algorithm (CBOF) [20] need four parameters. In order to solve the parameter selection problem, A non-parameter outlier detection algorithm called NOF based on Natural Neighbor is proposed in paper [16]. Nonetheless, almost all of the existing outlier detection algorithms, even NOF, must need a parameter n to specify the number of the outliers or α that the percentage of outliers in a dataset, which is known as the top-n problem. However, it is well known

* Corresponding author.

E-mail address: qszyu@cqu.edu.cn (Q. Zhu).

that researchers are hard to know the number of outliers contained in a dataset.

In order to overcome these above mentioned problems, we propose a novel outlier cluster detection algorithm called ROCF which does not require the top-n parameter. Firstly, we propose a preliminary clustering algorithm that is devoted to outlier cluster detection based on MUTual Neighbors Graph (MUNG) constructed by connecting each point to its mutual neighbors. After that, we propose an outlier cluster detection approach based on the idea that the size of outlier clusters is usually much smaller than the normal clusters. Then we detect which clusters are outlier clusters via Decision Graph instead of parameter n or α by manually set. We also figure out the outlier rate of database, finally output the outliers and outlier clusters. Therefore the proposed algorithm can detect outliers and outlier clusters. Moreover, although ROCF need parameter k , the number of neighbors, to construct mutual neighbor graph, ROCF doesn't need parameter n or α .

The paper is organized as follows. In Section 2, we present the existing definition and our motivation. In Section 3, the proposed outlier clusters detection algorithm and its related definitions will be described in detail. In Section 4, a performance evaluation is made and the results are analyzed. Section 5 concludes this paper.

2. Related work

As a primary method of data mining and data analysis, outlier detection get more and more attentions. A lot of outlier detection algorithms have been proposed. However, most of the existing outlier detection algorithms are hard to detect the outlier clusters, such as distribution-based, depth-based, distance-based and density-based algorithms. Cluster-based algorithms can solve this problem well, and many cluster-based outlier detection algorithms have been proposed. OFP [17], FindOut [18], FindCBLOF [19] and CBOF [20] are the representative cluster-based outlier detection algorithms. Recently some cluster-based outlier detection algorithms were proposed as well. For example, Min et al. [21] proposed an efficient outlier detection algorithm based on data clustering over massive data, and Jobe et al. [22] proposed a cluster-based outlier detection scheme for multivariate data.

In the following contents, we will briefly introduce the concept of CBOF. To compute cluster based outlier factor of point p , denoted as $CBOF(p)$, CBOF must cluster the datasets firstly. Therefore, CBOF needs a clustering algorithm, and LDBSCAN [23] is used in CBOF. LDBSCAN computes the local outlier factor (LOF) of each point of dataset firstly, then defines the core point, as the following definition.

Definition 1. (Core point): A point p is a core point w.r.t. LOFUB if $LOF(p) \leq LOFUB$.

Here, LOFUB is a parameter which should be manually set. If $LOF(p)$ is small enough, it means that point p is not an outlier and must belong to a cluster. Therefore, it can be regarded as a core point. Then LDBSCAN continues to extend from Core point until all of the points are visited. After clustering the datasets, CBOF finds the boundary between normal and abnormal clusters.

Definition 2. (Upper bound of the cluster-based outlier): Suppose D is the dataset, and $C = \{C_1, C_2, \dots, C_k\}$ is the set of clusters in the sequence that $|C_1| \geq |C_2| \geq \dots \geq |C_k|$. Given parameter $alpha$, the value of UBCBO is i if $(|C_1| + |C_2| + \dots + |C_{(i-1)}|) \geq |D| * alpha$ and $(|C_1| + |C_2| + \dots + |C_{(i-2)}|) \leq |D| * alpha$.

For example, if $alpha$ is set to 90%, we intend to regard clusters which contain 90% of data points as normal clusters, and the others are abnormal clusters.

Definition 3. (Cluster-based outlier): Let C_1, C_2, \dots, C_k be the clusters of the database D discovered by LDBSCAN. Cluster-based outliers are the clusters in which the number of the objects is no more than $|C_{UBCBO}|$.

After discovering the cluster-based outliers, CBOF also computes the cluster-based outlier factor as follows.

Definition 4. (Distance between two clusters): Let C_1, C_2 be the clusters of the database D . The distance between C_1 and C_2 is defined as

$$dist(C_1, C_2) = \min\{dist(p, q) | p \in C_1, q \in C_2\} \quad (1)$$

Definition 5. (Cluster-based outlier factor): Let C_1 be a cluster-based outlier and C_2 be the nearest non-outlier of C_1 . The cluster-based outlier factor of C_1 is defined as

$$CBOF(C_1) = |C_1| * dist(C_1, C_2) * \sum_{p_i \in C_2} \frac{lrd(p_i)}{|C_2|} \quad (2)$$

Here, $|C|$ is the number of the objects in C , $lrd(p_i)$ is the local reachability density of p_i . The outlier factor of cluster C_1 captures the degree to which we call C_1 an outlier cluster.

Though above definitions and analysis, we can see that the computation of CBOF is based on the clustering result. Therefore, once the result of clustering is undesirable, the CBOF(p) is unrepresentative, which lead to the result of outliers detecting is undesirable. Moreover, there are too many parameters in CBOF. The first step of CBOF needs three parameters (k , pct and LOFUB) to cluster the database. Then CBOF also need parameter alpha to discovery the cluster-based outliers. However, it is well known that these parameters are hard to set by researchers.

In this paper, we propose a novel outlier cluster detection algorithm called ROCF. First, ROCF briefly clusters the dataset via constructing Mutual Neighbors Graph, then constructs the Decision Graph. Finally, ROCF detects out the outliers and outlier clusters though Decision Graphs instead of parameter n or α . The detailed introduction will be made in the following section.

3. The proposed algorithm

In this section, the proposed algorithm (ROCF), and its related concept will be introduced in detail. Let D be a database, p and q be some objects in D , and k be a positive integer to indicate the number of neighbors of each object.

Definition 6. (Mutual Neighbor (MN)): If p is the neighbor of q , and q is the neighbor of p at the same time. Then we call p is a Mutual Neighbor of q , and similarly, q is a Mutual Neighbor of p .

Definition 7. (MUTual Neighbor Graph (MUNG)): MUTual Neighbor Graph can be constructed by connecting each point to its mutual neighbors.

Note that the number of neighbors of each point is k . However, it is possible that different points have different numbers of mutual neighbors. As shown in Fig. 1, point A that lies in the dense region possesses more mutual neighbors than points that lies in the sparse region. Moreover, point B is a local outlier and does not have any mutual neighbors. In addition, from Fig. 1, we can see that the number of mutual neighbors of some points, such as point A, will rise up with the value of k increasing. However the numbers of mutual neighbor of some points, such as point B and the three points located to the right of B, is unchanged. The reason is that point B is a local outlier, and these three points form a outlier cluster that will be defined in Definition 10.

In order to detect out the outlier clusters, we first need to cluster the datasets. From Fig. 1, we can see that MUNG has roughly

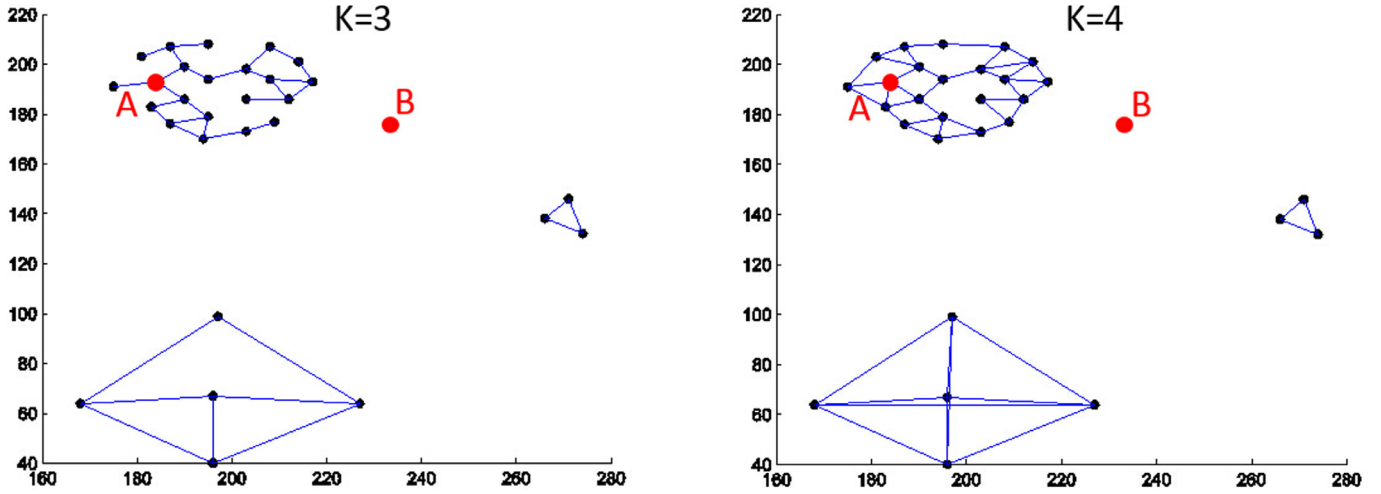


Fig. 1. The example of Mutual Neighbors ($k = 3, 4$).

clustered the datasets. Therefore, in this paper, we propose a rough clustering algorithm that is devoted to detect the outliers and outlier clusters based on MUNG, as shown in Algorithm 1.

Algorithm 1 RoughlyCluster(D, k).

• **Output:** The rough clustering results $C = \{C_1, C_2, \dots, C_n\}$

- (1) Constructing the Mutual Neighbor Graph; $n=1$;
 - (2) Randomly select an object x ; $visited(x)=true$; $C_n = C_n \cup MN(x)$;
 - (3) while exist $y \in C_n$ and $visited(y) \neq true$
 - a. then $visited(y) = true$; $C_n = C_n \cup MN(y)$;
 - (4) if exist $z \in D$ and $visited(z) \neq true$ then $n=n+1$; goto Step2;
-

In fact, some clustering algorithms that based on mutual neighbor graph have been proposed. For example, in paper [25], M.R. et al. proposed a clustering and outlier detection algorithm (CMKNNG) based on the Connectivity of the Mutual K-Nearest-Neighbor Graph. The Mutual K-Nearest-Neighbor Graph of CMKNNG needs to be connected while the mutual neighbor graph of Algorithm 1 does not. Moreover, the clustering procedure of Algorithm 1 is simpler than CMKNNG. It is important to note that the clustering results of Algorithm 1 are not necessarily right. A large complex manifold cluster may be divided into two or more than two clusters by Algorithm 1, when the parameter k is set to a small value. However, this result does not influence outlier cluster detection. Since once a cluster is normal, even it comes from a large cluster, the size of this cluster must be much bigger than outlier cluster.

In this paper, after clustering the datasets, we propose a novel outlier cluster detection algorithm (ROCF) based on the idea that the size of outlier clusters is usually much smaller than the normal clusters. For example, consider the 2d data set in Fig. 2. There are four clusters in this figure, $C_1(20)$, $C_2(30)$, $C_3(300)$ and $C_4(10000)$. Obviously, C_1 and C_2 should be regarded as outlier clusters.

Definition 8. (Transition Level (TL)): Suppose $C = \{C_1, C_2, \dots, C_n\}$ is the set of clusters in the sequence that $|C_1| \leq |C_2| \leq \dots \leq |C_n|$. The Transition Level of C_i , denoted as $TL(C_i)$, is defined as the ratio of the size of C_{i+1} and the size of C_i . The formulation as the following equation.

$$TL(C_i) = \frac{|C_{i+1}|}{|C_i|}, i = 1, 2, \dots, n-1 \quad (3)$$

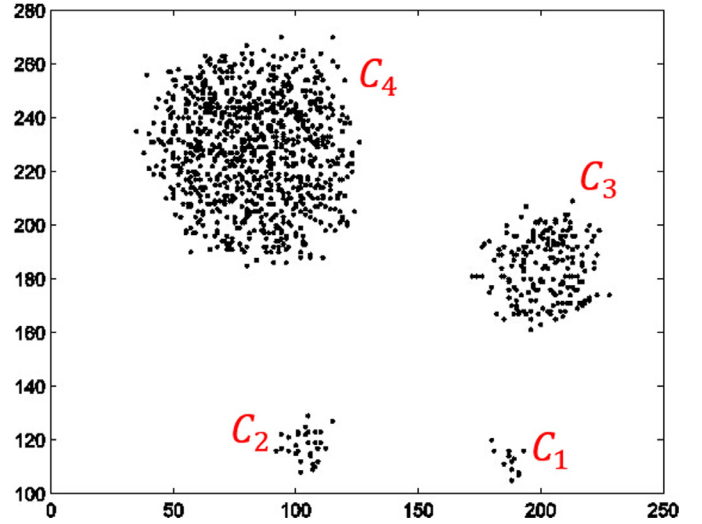


Fig. 2. The example data of normal clusters and outlier clusters.

Based on the above definition, in Fig. 2, we can obtain that $TL(C_1) = 1.5$, $TL(C_2) = 10$ and $TL(C_3) = 33.3$. The size of C_4 is the maximum. Therefore, it is impossible that C_4 is outlier cluster. We can see that the value of $TL(C_2)$ is much larger than $TL(C_1)$. However, a large value of $TL(C_i)$ does not definitely means that C_i is outlier cluster. For example, $TL(C_3)$ is the largest, but C_3 is not an outlier cluster. Hence, we define the Relative Outlier Cluster Factor (ROCF) as the following definition.

Definition 9. (Relative Outlier Cluster Factor (ROCF)): The relative outlier cluster factor of cluster C_i , denoted as $ROCF(C_i)$, is defined as follows.

$$ROCF(C_i) = 1 - e^{-\frac{n(C_i)}{|C_i|}} = 1 - e^{-\frac{|C_{i+1}|}{|C_i|^2}}, i = 1, 2, \dots, n-1 \quad (4)$$

The range of $ROCF(C_i)$ is (0,1). A high value of $ROCF(C_i)$ indicates that C_1, C_2, \dots, C_i are good candidates for outlier clusters. Note that $ROCF(C_i)$ is relative to C_{i+1} . Fig. 3 shows the Decision Graph (DG) of the dataset in Fig. 2. From Fig. 3, we can obviously find that C_1 and C_2 are outlier clusters. And the outlier clusters is defined as the follow definition.

Definition 10. (Outlier Clusters): Let C_1, C_2, \dots, C_n be the clusters of the database D discovered by Algorithm 1. If $ROCF(C_b) =$

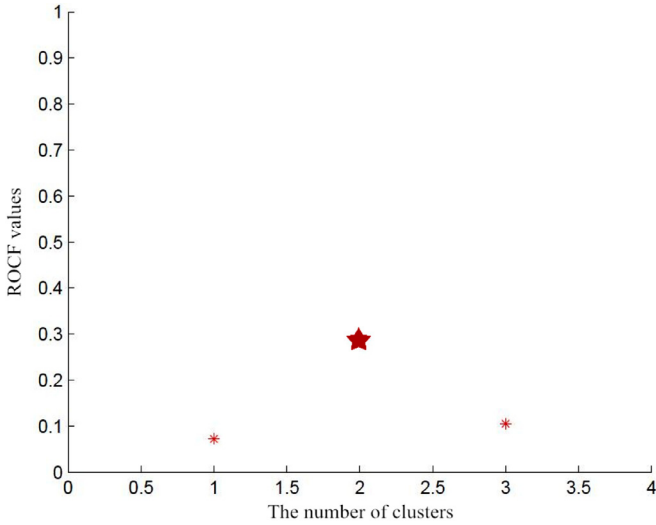


Fig. 3. The Decision Graph of the example data.

$\max\{ROCF(C_i)\}$ and $ROCF(C_b) > 0.1$, then C_1, C_2, \dots, C_b are Outlier clusters.

Note that the value scope of b is $[0, n-1]$. The proposed algorithm ROCF presumes that the largest cluster C_n must be a normal cluster. The definition of outlier clusters is applied to any dataset that contains outlier clusters. If all of clusters, discovered by Algorithm 1, are normal clusters, then the value of b is equal to 0, and the change of ROCF is very little and the value of ROCF is small. Through a number of experiments, generally, we find that the value of b is less than 0.1 or even smaller when all clusters in a dataset are normal. Because, if $ROCF(C_b) < 0.1$, then we can prove that $|C_{b+1}|/|C_b|^2 < 0.1$. $|C_{b+1}|/|C_b|^2 < 0.1$ implies that the change of size from C_b to C_{b+1} is little. Therefore, we think that there are no outlier clusters when $ROCF(C_b) < 0.1$. As shown in Fig. 4.b, for instance, we can see that the original dataset includes some scattered outliers, but don't contains outlier clusters. So, from Fig. 4.a, we can see that the value of ROCF of all clusters is small (smaller than 0.04) and not much changes. Thus we can judge whether

a database contains outlier clusters and detect out these outlier clusters via Decision Graph. The outliers, marked as red color in Fig. 4.b, don't assigned to any cluster when clustering the database using Algorithm 1. So ROCF can detect the scattered outliers too.

Definition 11. (Outlier Rate): Let C_1, C_2, \dots, C_n be the clusters of dataset D discovered by Algorithm 1, and C_1, C_2, \dots, C_b be Outlier clusters. Then the Outlier Rate(OR) of dataset D is defined as the following equation.

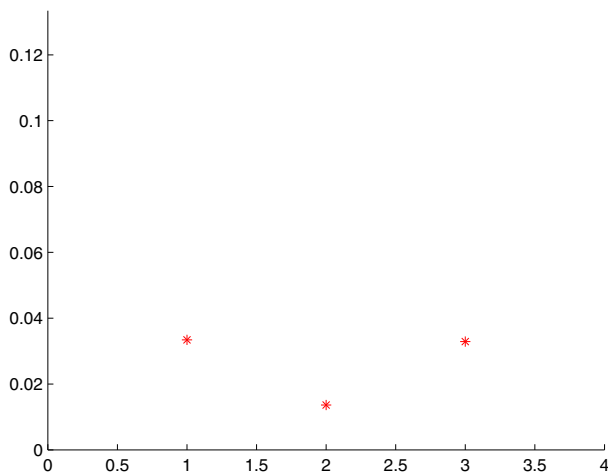
$$OR = 1 - \frac{\sum_{i=b+1}^n |C_i|}{|D|} \quad (5)$$

OR is the percentage of outliers and outlier clusters in a database. The significance of OR is that it can be used in any outlier detection algorithm. In other words, all of the existing outlier detection algorithms need the parameter α or n , but the value of α or n is hard to set by researchers. Now, the value of α or n that needed by the other outlier detection algorithm can be set equal to the value of OR.

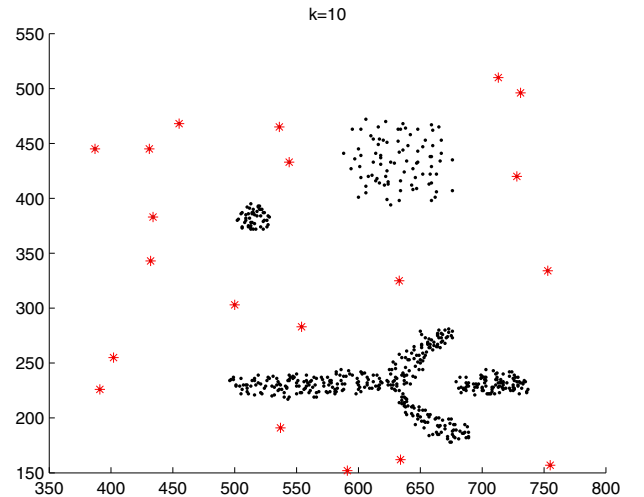
Based on the above concept and definitions, we propose a novel outlier clusters detection algorithm(ROCF) without top- n parameter, as shown in Algorithm 2.

In algorithm ROCF, $C = \{C_1, C_2, \dots, C_n\}$ is the rough clustering result discovered by Algorithm 1. Parameter k is number of neighbors of each point same as in Algorithm 1. So, indeed, after clustering the datasets, ROCF doesn't need any parameters when detecting the outlier clusters.

The proposed algorithm ROCF first uses Algorithm 1 to roughly cluster the dataset, and Then comparing the size of each cluster with k . If the size of C_i is smaller than k , then C_i is marked as outlier cluster and deleted from C , so ROCF can detect the isolated outliers. Afterwards, the proposed algorithm computing the ROCF of clusters which still remain in C and construct the Decision Graph. At last, ROCF finds the boundary b between outlier clusters and normal clusters via Decision Graph, and outputs the outlier clusters $OC = \{C_1, C_2, \dots, C_b\}$. In this way, ROCF can not only detect the isolated outliers, but also the outlier clusters. Moreover, the greatest strength of ROCF is that ROCF doesn't need parameter n or α .



(a) The Decision Graph



(b) The detection result

Fig. 4. The instance that no outlier clusters.

Algorithm 2 ROCF(C,k).
$$\setminus \setminus C = \{C_1, C_2, \dots, C_n\}$$

• **Output:** $OC = \{C_1, C_2, \dots, C_b\} \setminus \setminus b = 0, 1, 2, \dots, n - 1$

- (1) for $\forall C_i \in C$
 - a. if $\text{size}(C_i) < k$ then C_i is marked as outlier cluster and delete C_i from C ;
- (2) for $i = 1 : \text{size}(C) - 1$;
 - a. $TL(C_i) = \frac{|C_{i+1}|}{|C_i|}$;
 - b. $ROCF(C_i) = 1 - e^{-\frac{TL(C_i)}{|C_i|}}$;
- (3) Construct and display the Decision Graph.
- (4) Find the value of b that $ROCF(C_b) = \max\{ROCF(C_i)\}$.
 - a. if $ROCF(C_b) < 0.1$ then $b=0$;
- (5) Compute the value of $OR = 1 - \frac{\sum_{i=b+1}^n |C_i|}{|D|}$.
- (6) Output OR and the outlier clusters $OC = \{C_1, C_2, \dots, C_b\}$.

4. Performance evaluation

In order to show the effectiveness of the proposed method, performance evaluation based on both synthetic datasets and real-world datasets is conducted. In the experiment, we compared our method with two representative approaches (LOF and CBOF), since almost all of the existing outlier detection algorithms face the top-n problem.

4.1. Metrics for measurement

For performance evaluation of the algorithms, we use two metrics, namely Recall and Precision [24], to evaluate the detection results. Let PN be the number of the true outliers that dataset D contains. Let TP be the number of instances that are correctly classified as outliers by an algorithm, and FP be the number of instances that are wrongly classified as outliers. Then the Recall (Re) and Precision (Pr) are defined as follows.

$$Re = \frac{TP}{PN} \quad (6)$$

$$Pr = \frac{TP}{TP + FP} \quad (7)$$

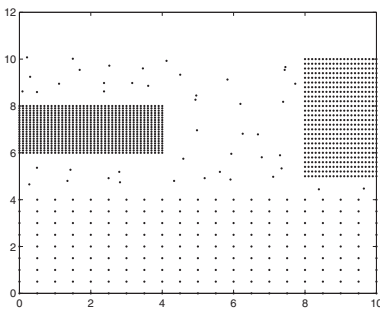
The possible maximum value of Re and Pr is 1, and the possible minimum value of Re and Pr is 0. The bigger the value of Re and Pr is, the better the results of outlier detection are.

4.2. Synthetic examples

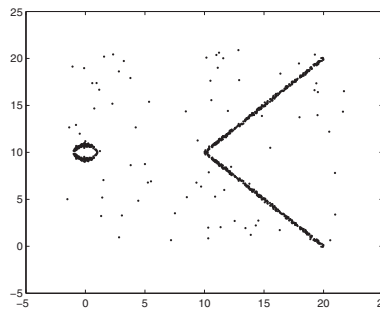
We first conduct comparison experiments based on three synthetic datasets. Fig. 5 shows the three original datasets. Synthetic dataset D1, taken from paper [14], contains different degrees of cluster density and size, and some sparse outliers. D2, taken from paper [14], contains various cluster patterns and some dense outliers. D3 contains one normal cluster and three small outlier clusters. In agreement with the previous results, the outlier or outlier clusters detected by each method are colored red in the following experiments. Note that the value of parameter α of CBOF is the rate of normal objects.

The experimental results on D1 are shown in Fig. 6. A total of 1641 objects are included in D1, of which 45 objects are outliers or outlier clusters. In the experiment of LOF, we set the value of top-n = 45. Thus, the experimental result of LOF on D1 is good, and the value of Re and Pr is 0.93, as shown in Table 1. However, if the value of top-n is smaller (bigger) than 45, the value of $Pr(Re)$ is improved while the value of $Re(Pr)$ becomes lower. The experimental result of CBOF is not as good as LOF. Because some outliers are clustered into the sparse cluster during clustering, so these outliers can not be detected. Although $Pr = 1$, the value of Re of CBOF is the lowest(0.24). In order to improve Re of CBOF, we set $\alpha = 0.85$. But the value of Pr becomes undesirable, only 0.19. As shown in Fig. 6.c, we obtain the boundary b that $ROCF(b)$ is the maximum via Decision Graph. Here $b = 2$ means that there are two outlier clusters. Then output the outliers and outlier clusters as shown in Fig. 6.d. The values of Re and Pr of ROCF are 0.933 and 1. The greatest strength is that the OR of D1, obtained by ROCF algorithm, is 0.026, which is very close to the real outlier rate (0.027). So ROCF algorithm solves the top-n problem via constructing the Decision Graph.

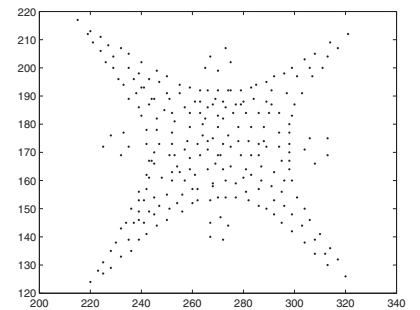
The experimental results on D2 are shown in Fig. 7. A total of 879 objects are included in D2, of which 79 objects are outliers or outlier clusters. As shown in Fig. 7.a, we set top-n = 50 for LOF. The real outliers of D2 is 79 that we don't know previously. Therefore the results of LOF is undesirable, and the value of Re is only 0.63, as shown in Table 2. However, if we set the value of top-n with correct value, the result of LOF is well ($Re = 0.93, Pr = 0.93$). As the experiment on D1, if the value of top-n is larger than the number (79) of real outliers, the value of Pr becomes lower (0.77). As shown in Fig. 7.b, we set $\alpha = 0.9$ of CBOF. CBOF algorithm detect out 87 outliers that 12 normal objects are mistakenly regarded as outliers. The value of Re of CBOF is 0.95, while the value of Pr is only 0.86. As same as experiment on D1, we first construct the Decision Graph of D2, then find the boundary b between normal clusters and outlier clusters via Decision Graph. Here the value of b is 2. So D2 is detected out two outlier cluster by ROCF algorithm. Finally, output the outlier detecting results of ROCF al-



(a) D1



(b) D2



(c) D3

Fig. 5. The original datasets.

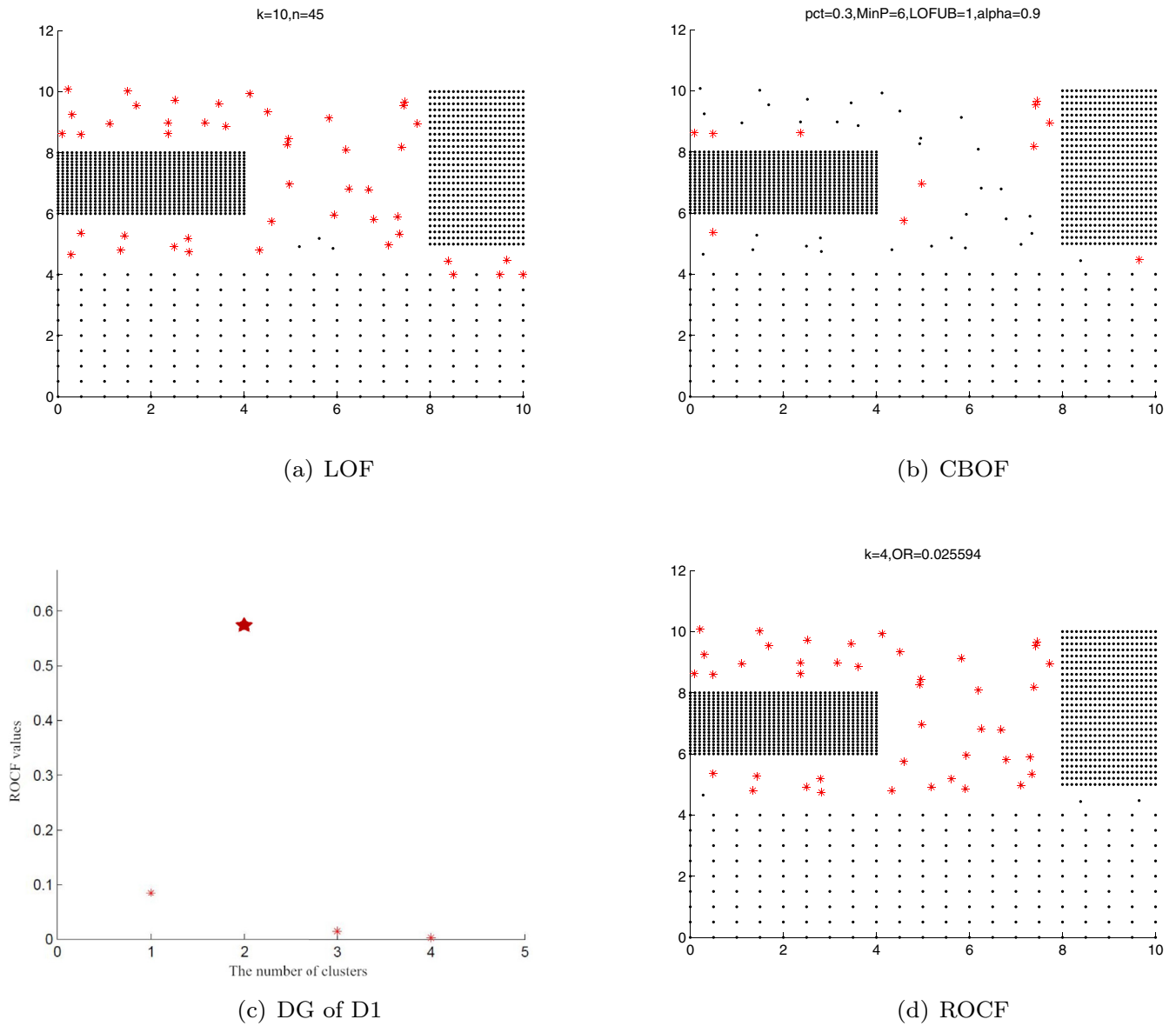


Fig. 6. Detection results and Decision Graph of D1.

Table 1

Recall and Precision of the three methods experiment on D1.

LOF				CBOF				ROCF			
k	n	Re	Pr	k	alpha	Re	Pr	k	OR	Re	Pr
10	30	0.67	1	6	0.95	0.24	1	4	0.026	0.93	0.93
	45	0.93	0.93		0.90	0.24	1				
	60	0.96	0.71		0.85	1	0.19				

Table 2

Recall and Precision of the three methods experiment on D2.

LOF				CBOF				ROCF			
k	n	Re	Pr	k	alpha	Re	Pr	k	OR	Re	Pr
15	50	0.63	1	15	0.95	0.95	0.86	10	0.085	0.94	0.94
	79	0.93	0.93		0.90	0.95	0.86				
	100	0.96	0.77		0.85	0.95	0.86				

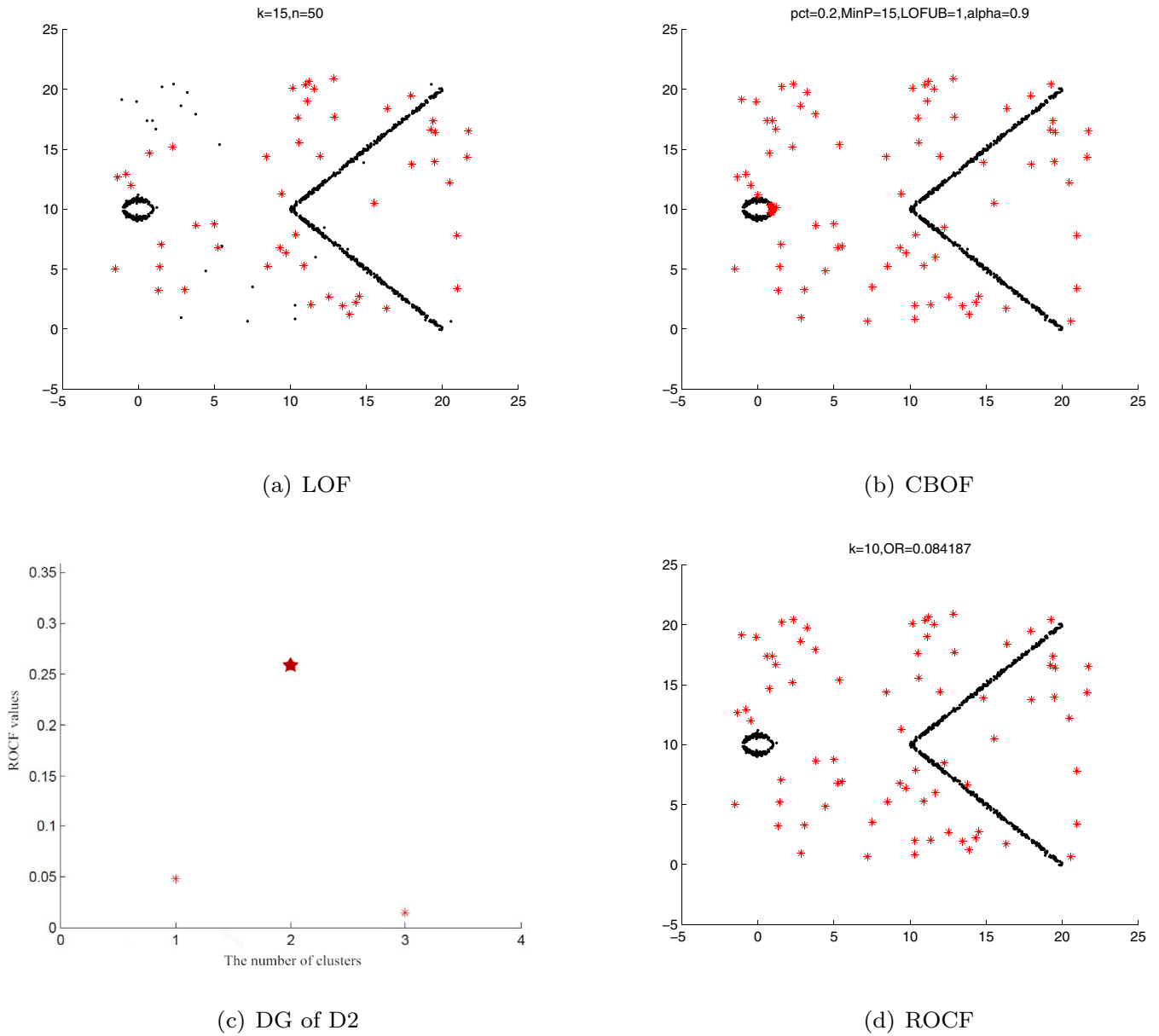


Fig. 7. Detection results and Decision Graph of D2.

gorithm that shown in Fig 7.d, ROCF detect out 74 outliers. The value of Re and Pr are 0.94. Although the Re of ROCF is lower than CBOF, ROCF doesn't need the top- n parameter that CBOF needed, and the Pr of ROCF is higher than CBOF. Moreover, ROCF can adaptively compute the $OR = 0.085$ that is very close to the real outlier rate (0.090) of D2.

The experimental results on D3 are shown in Fig. 8. A total of 254 objects are included in D3, of which 20 objects are outliers or outlier clusters. From the result shown in Fig. 8.a, we set parameter top- $n = 30$ for LOF, we can see that some normal objects are detected as outliers. Moreover, since mistaken value of top- n parameter, although the Re of LOF is 0.9, the Pr of LOF is low, only 0.57, as shown in Table 3. Even the value of top- n is 20 that the correct value, the Pr is only 0.70. If the value of top- n becomes smaller (10) than 20, the value of Re would become smaller (0.35). So, through the above analysis, we can conclude that LOF is not applicable to detect the outlier clusters. As shown in Fig. 8.b, the result of CBOF is better than LOF. CBOF algorithm correctly detect out the four outlier clusters. However, CBOF still mistakenly re-

gard some normal objects as outliers. Therefore, the value of Re of CBOF is 1 while the value of Pr of CBOF is only 0.87. As same as the above two experiments, we first find the value of boundary b between normal clusters and outlier clusters via Decision Graph, shown in Fig. 8.c. Here the value of b is 4. So D3 is detected out 4 outlier clusters by RCOF algorithm. Finally output the experimental result that shown in Fig. 8.d. The greatest strength is that the value of Re and Pr of ROCF are the highest, equal 1, and the $OR = 0.079$ that obtained by ROCF equal to the value of real outlier rate (0.079) of D3.

4.3. Real data examples

We also applied the proposed method to real-world datasets (Iris) that obtained from the University of California, Irvine (UCI) machine learning repository. The iris datasets contains 150 objects that are grouped into 3 classes ("setosa", "versicolor" and "virginica"). In this experiment, we select "setosa" class as normal cluster and respectively select 10 objects from "versicolor" and

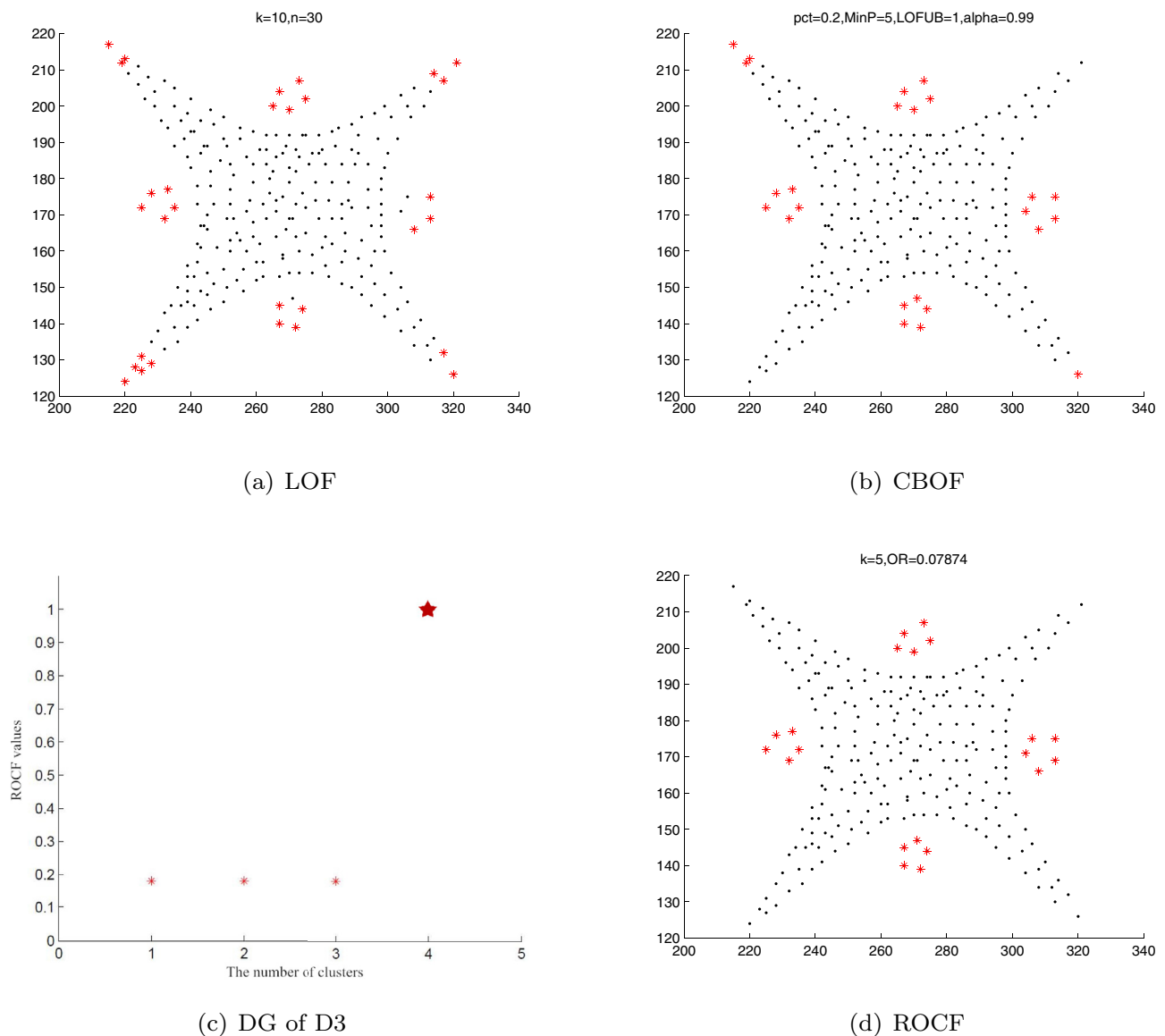


Fig. 8. Detection results and Decision Graph of D3.

Table 3
Recall and Precision of the three methods experiment on D3.

LOF				CBOF				ROCF			
k	n	Re	Pr	k	alpha	Re	Pr	k	OR	Re	Pr
10	10	0.35	0.70	5	0.99	1	0.87	5	0.079	1	1
	20	0.70	0.70		0.90	1	0.87				
	30	0.90	0.57		0.85	1	0.87				

“virginica” classes as two small outlier clusters. So, in this real-world-datasets experiment, the real datasets contains 70 objects that consist of one normal cluster (50 objects) and two small outlier clusters. Each small outlier cluster contains 10 objects and each objects is 4 dimensions.

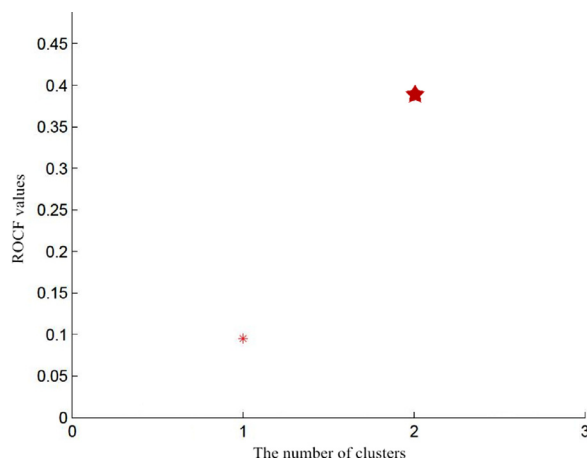
The experimental results are shown in Table 4. From the results of Table 4, we can see that $Re = 0.25$ and $Pr = 0.5$ of LOF when top- $n = 10$ that smaller than the real number of outliers. Although the Re and Pr of LOF are both improved when top- $n = 20$, the value of Re and Pr is still only 0.70. If we further increase the value of top- $n(30)$, though the value of Re is 1, the value of Pr of LOF is decreased to 0.67. This result further demonstrate that LOF is not applicable to detect the outlier clusters. The result of CBOF shows

that $Re = 0$ and $Pr = 0$ when $alpha = 0.9$ and 0.8 . However, when $alpha = 0.68$ means that outlier rate is equal to $OR(0.32)$, the value of Re is 1 and Pr is 0.95. But we know that the value of $alpha$ is very difficult to know in advance. As same as the experiment on Synthetic datasets, ROCF first construct the Decision Graph shown in Fig. 9. Then ROCF find the largest value of $ROCF(C_b)$ that b is the boundary between normal clusters and outlier clusters. Here $b = 2$ means that ROCF algorithm successfully detect out two outlier clusters. Moreover, the value of OR that obtained by ROCF is 0.32 that very close to the value(0.29) of real outlier rate of the dataset. The $Re(1)$ and $Pr(0.5)$ of ROCF are the maximum value in this experiment.

Table 4

Recall and Precision of the three methods experiment on Iris.

LOF				CBOF				ROCF			
k	n	Re	Pr	k	alpha	Re	Pr	k	OR	Re	Pr
20	10	0.25	0.50	8	0.90	0	0	9	0.32	1	0.95
	20	0.70	0.70		0.80	0	0				
	30	1	0.67		0.68	1	0.95				

**Fig. 9.** The Decision Graph of Iris.

5. Conclusions

In this study, we propose a novel outlier cluster detection algorithm (ROCF) without top-n parameter. The proposed outlier detection algorithm is cluster-based, so ROCF can detect the outlier clusters that are hard to detect out by other distance-based or density-based outlier detection algorithms, such as LOF. Unlike most of cluster-based outlier detection algorithms that need many parameters, ROCF only need one parameter k to indicate the number of neighbors. Moreover, ROCF can automatically figure out the outlier rate of a dataset via constructing the Decision Graph. Therefore, the proposed algorithm can detect the outliers and outlier clusters without parameter top-n (α) to specify the number of outliers, or the percentage of outliers in a database. Through the above experimental analysis, we confirmed that the proposed method can accurately detect outliers and outlier clusters without parameter top-n, and automatically figure out the value of outlier rate of the datasets.

Acknowledgment

The authors would like to greatly thank the editors and the anonymous reviewers for their insightful and helpful comments, which resulted in substantial improvements to this work.

This research was supported by the National Natural Science Foundation of China (No. 61272194) and the Project (No. KJZH17104).

References

- [1] W. Jin, A.K. Tung, J. Han, Mining top-n local outliers in large databases, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2001.
- [2] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, Elsevier, 2011.
- [3] T. Pang-Ning, M. Steinbach, V. Kumar, Introduction to data mining, Library of Congress, 2006.
- [4] E.M. Knorr, R.T. Ng, A unified notion of outliers: Properties and computation, KDD, 1997.
- [5] E.M. Knorr, R.T. Ng, V. Tucakov, Distance-based outliers: algorithms and applications, VLDB J. Int. J. Very Large Data Bases 8 (3–4) (2000) 237–253.
- [6] S. Shekhar, S. Chawla, A Tour of Spatial Databases, Prentice Hall Upper Saddle River, New Jersey, 2002.
- [7] D.M. Hawkins, Identification of Outliers, 11, Springer, 1980.
- [8] V. Barnett, T. Lewis, Outliers in Statistical Data, 3, Wiley New York, 1994.
- [9] I. Ruts, P.J. Rousseeuw, Computing depth contours of bivariate point clouds, Comput. Stat. Data Anal. 23 (1) (1996) 153–168.
- [10] T. Johnson, I. Kwok, R.T. Ng, Fast computation of 2-dimensional depth contours, KDD, 1998. Citeseer
- [11] E.M. Knox, R.T. Ng, Algorithms for mining distancebased outliers in large datasets, in: Proceedings of the International Conference on Very Large Data Bases, 1998. Citeseer
- [12] M.M. Breunig, et al., LOF: Identifying density-based local outliers, ACM Sigmod Record, ACM, 2000.
- [13] W. Jin, et al., Ranking outliers using symmetric neighborhood relationship, in: Advances in Knowledge Discovery and Data Mining, Springer, 2006, pp. 577–593.
- [14] J. Ha, S. Seok, J.S. Lee, Robust outlier detection using the instability factor, Knowl. Based Syst. 63 (2014) 15–23.
- [15] Y.-f. Jin, Q.-s. Zhu, X.-l. Zou, Clustering algorithm of outliers based on adjacency graph, Comput. Eng. 11 (2008) 027.
- [16] J. Huang, et al., A non-parameter outlier detection algorithm based on natural neighbor, Knowl. Based Syst. (2015).
- [17] M.-F. Jiang, S.-S. Tseng, C.-M. Su, Two-phase clustering process for outliers detection, Pattern Recognit. Lett. 22 (6) (2001) 691–700.
- [18] D. Yu, G. Sheikholeslami, A. Zhang, Findout: finding outliers in very large datasets, Knowl. Inf. Syst. 4 (4) (2002) 387–412.
- [19] Z. He, X. Xu, S. Deng, Discovering cluster-based local outliers, Pattern Recognit. Lett. 24 (9) (2003) 1641–1650.
- [20] L. Duan, et al., Cluster-based outlier detection, Ann. Oper. Res. 168 (1) (2009) 151–168.
- [21] J.-k. Min, An efficient outlier detection algorithms based on data clustering over massive data, Database Res. 31 (3) (2015) 59–71.
- [22] J.M. Jobe, M. Pokojovy, A cluster-based outlier detection scheme for multivariate data, J. Am. Stat. Assoc. 110 (512) (2015) 1543–1551.
- [23] L. Duan, et al., A local-density based spatial clustering algorithm with noise, Inf. Syst. 32 (7) (2007) 978–986.
- [24] Z.P. Zhang, Y.X. Liang, A data stream outlier detection algorithm based on reverse k nearest neighbors, Advanced Materials Research., Trans Tech Publ., 2011.
- [25] M.R. Brito, E.L. Chavez, A.J. Quiroz, et al., Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection, Stat. Probab. Lett. 35 (1) (1997) 33–42.