Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A local-gravitation-based method for the detection of outliers and boundary points[☆]

Jiang Xie [*], Zhongyang Xiong, Qizhu Dai, Xiaoxia Wang, Yufang Zhang

Chongqing University of Posts and Telecommunications, Chongqing 400065, China

ARTICLE INFO

Article history:
Received 7 January 2019
Received in revised form 29 November 2019
Accepted 30 November 2019
Available online 6 December 2019

Keywords:
Outlier detection
Boundary points
Local resultant force
Nearest neighbors
Data mining

ABSTRACT

Detection of outliers and boundary points represents an effective, interesting and potentially valuable pattern, which may be more important than that of normal points. In order to detect outliers and boundary points, we propose a local-gravitation-based method in which each data point is viewed as an object with both mass and a local resultant force (LRF) generated by its neighbors. With the increase of neighbor, the LRF of outliers, boundary points and interior points varies at different rates. In this paper, the LRF changing rates of points with lower densities have higher scores, namely the changing rate of an outlier is greater than that of a boundary point and inner point. In other words, top-m ranked points can be identified as outliers, and the greater the LRF changing rate of a point is, the more likely it is a boundary point. The main advantage of our proposed method is that it does not depend on the choice of K value, which improves the detection performance. The experimental results on synthetic and real data sets show that the proposed method is better than the existing methods.

1. Introduction

In many applications, data analysis is needed to detect outliers and boundary points. Unlike clustering, classification and pattern analysis aimed at finding general patterns, outlier and boundary detection are often used to identify observations representing effective, interesting and potentially valuable patterns in data [1]. The detecting method of outliers have proven to be efficient in the detection of credit card fraud, telecommunication fraud, network intrusion, etc. [2], while the detecting method of boundary points is widely used in medical processing and image processing. For example, the detecting system of liver disorder may regard normal observations as healthy patients while boundary points as patients who are more likely to have a liver disorder in the future. In a word, the detection of outliers and boundary points can been used to extract useful information from a growing amount of digital datasets [3,4].

Unlike most gravitation-based methods, this paper focuses on the detection method of outliers and boundary points. Based on Newton's third law, Bharti et al. [5] proposed a gravitational outlier detection (GOD) for wireless sensor networks. GOD calculates the centripetal force between the new data points and the central data points. Then, according to the change of the acceleration of the data points, GOD judged whether the data points are outliers. Based on gravity theory, Z. Wang et al. [6] presented a model for distinguishing boundary points, interior points and outliers. By calculating the direction of the resultant forces acting on the three kinds of points, the model can be used to distinguish the boundary points, interior points and outliers. Our work is to use Newton's law of gravitation to calculate the local resultant forces on each data point. Then, the change values of local resultant forces of each data point under different number of neighbors are accumulated, and the type of data point is judged by accumulative change rate. Compared with other methods based on gravity theory [5,6], our method is insensitive to parameter k.

In summary, firstly, we propose a new model to extract information in the detection of three types of points. Secondly, based on the new model, we propose an outlier detection method which can accurately find outliers without any parameters input. Thirdly, based on the new model, we propose a boundary-point detection method which can detect boundary points without k input.

This paper is organized as follows: Section 2 presents Literature review, Section 3 introduces the proposed method, Section 4 demonstrates the effectiveness of our method through experimental results on synthetic and real-world data sets and Section 5 presents conclusions.

[☆] No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.105331.

[*] Corresponding author.
E-mail address: xiejiang@cqu.edu.cn (J. Xie).

https://doi.org/10.1016/j.knosys.2019.105331
0950-7051/© 2019 Elsevier B.V. All rights reserved.

2. Literature review

Outlier is defined by Hawkins as an observation deviating so much from other observations as to arouse suspicion that it was generated by a different mechanism [7,8]. According to three scenarios in which they are applied, outlier detecting methods can be classified as supervised, semisupervised and unsupervised. Besides, according to different models adopted by outlier detecting methods, these methods can be divided in eight categories [1,9]: distance-based models [3], density-based models,[10,11], clustering models [12], probabilistic models [13,14], classification models [15–19], spectral theory models [20] and information-theoretic models [21].

Most distance-based methods firstly compute distances between each object and its nearest neighbors, and then compare the distances of all objects [22]. It is implied that an object far from its neighbors is likely to be an outlier. The outlier scoring method based on k-nearest neighbor (KNN) distances [23] and the local distance-based outlier detection method (LDOF) [24] are two classical examples of distance-based methods. Although distance-based methods are simple and fast, it is difficult to select an appropriate number of neighbors which is critical to the performance of detection. Unlike distanced-based methods, density-based methods compare the density of point with that of its neighbors. It is implied that a point density that is different from its neighbors is likely to be an outlier. Typical density-based methods include the local outlier factor (LOF) [10] and the outlier detections for low density patterns (COF) [25] which perform well on most data sets. However, density-based methods depend on the choice of parameter which is used to determine the number of neighbors. Therefore, it is critical to choose the best parameter, as is the case with distance-based methods.

Boundary points are different from the outliers [26]. Boundary points do not have a standard definition but an universally accepted definition that boundary points are located in the edge of a class region, i.e., near free pattern space. Depending on whether labels for boundary points are available, detecting methods of boundary points can be classified as unsupervised, semi-supervised and supervised [27,28]. For example, the border-edge pattern selection (BEPS) method [13], a classical detecting method of boundary points, requires manual input of four parameters, which has a great impact on detecting results.

The local resultant force (LRF) in the detection of outliers and boundary points is a concept that comes from Newton's theory of gravitation. In 1977, Wright [29] proposed a hierarchical agglomerative algorithm known as the gravitational clustering which updates the position of each data point at each iteration and merges the points into clusters when they are close enough. Gómez et al. [30,31] proposed two clustering methods based on the gravitation model in which each data point is considered as an object with mass and the objects are moved through the gravitational force according to the Newton's second law of motion. Recently, Kundu [32], Zhang and Qu [33], and Sanchez et al. [34] also proposed clustering algorithms in which each point is moved iteratively through the gravitational force. In addition to clustering algorithms, other algorithms also use the gravitation model. For example, based on gravity and the law of motion, Rashedi et al. [35] proposed a gravitational search algorithm (GSA) which is generally classified as a heuristic optimization algorithm. Inspired by the phenomenon of gravitation and black hole, Hatamlou [36] recently proposed a new heuristic optimization approach called the black hole algorithm.

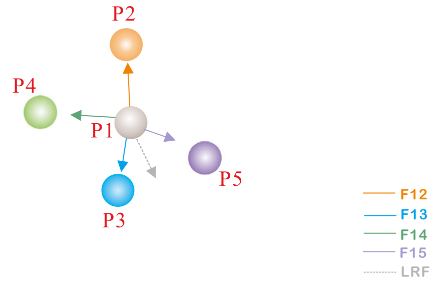


Fig. 1. Illustrates how to compute the local resultant force (LRF) of a data point, and the gray-dashed arrow represents the local resultant force (LRF) of point P1.

3. The proposed method

3.1. A brief description of gravity theory

The local gravitation in data reflects the relation between a point and its neighbors. According to the theory of gravitation, the attractive force between two objects is formulated as follows:

$$\overrightarrow{F}_{ij} = G\frac{m_i m_j}{D_{ij}^2}\hat{D}_{ij} \tag{1}$$

where $\overrightarrow{F}_{ij}$ represents the force between point i and point j. G is the gravitational constant. The distance between mass m_i and mass m_j is D_{ij}, and the unit vector $\hat{D}_{ij}$ is the direction of the line that connects the two objects along which the force acts. If the distances between a point and its different neighbors do not vary significantly in a local region, we can simplify (1) as follows :

$$\overrightarrow{F}_{ij} = Gm_i m_j \hat{D}_{ij} \tag{2}$$

Therefore, as shown in Fig. 1, the resultant force of point i with its k-nearest neighbors (LRF) can be computed as:

$$\overrightarrow{LRF}(i,k) = \sum_{j=1}^{k} \overrightarrow{F}_{ij} = Gm_i \sum_{j=1}^{k} m_j \hat{D}_{ij} \tag{3}$$

The unit vector $\hat{D}_{ij}$ encapsulates the directional information and distance between point i and its neighbors, k is the number of the nearest neighbors, and the set of m_i values weights factors in composing the forces in the neighborhood. In general, points with larger masses provide more influence on their neighbors, and points with smaller masses are more sensitive to the influence from their neighbors. Therefore, as shown in (4), the simplified definition of LRF is proposed to replace Newton's theory of gravitation [6].

$$\overrightarrow{LRF}(i,k) = \frac{1}{m_i} \sum_{j=1}^{k} \hat{D}_{ij} \tag{4}$$

The mass m_i of a point i is defined as follows:

$$m_i = \frac{1}{\sum_{j=1}^{k} D_{ij}} \tag{5}$$

According to (5), points in lower density areas are at longer distances from their neighbors, therefore the masses of these points become smaller. On the contrary, the masses of points in higher density areas become larger.

Furthermore, points in lower density areas (outliers and boundary points) are surrounded by neighboring points in a less uniform way. Therefore, according to (4), directions of forces to points from different neighbors are roughly the same, which results in a large magnitude for $\overrightarrow{LRF}(i,k)$. On the other hand,

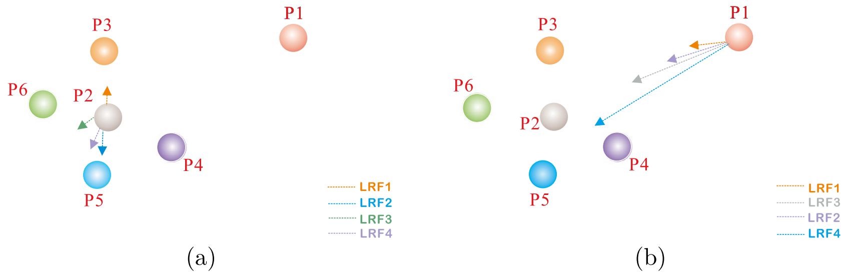


Fig. 2. Distinction of *LRF* on three types of points. (a) The *LRF* of different numbers of neighbors to inner point P2. (b) The *LRF* of different numbers of neighbors to outlier points or boundary points P1.

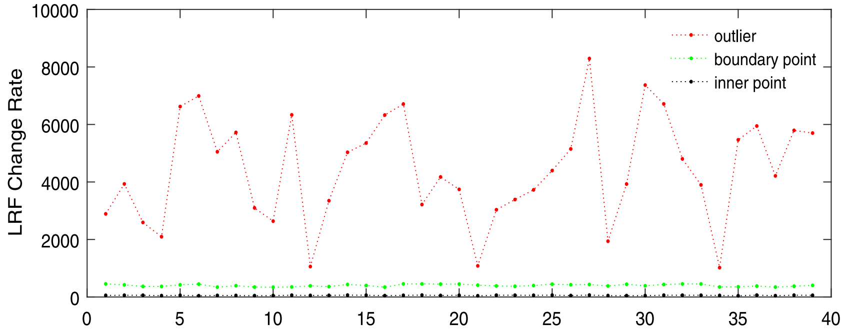


Fig. 3. The *LRF* change rate for different types of points over DS1.

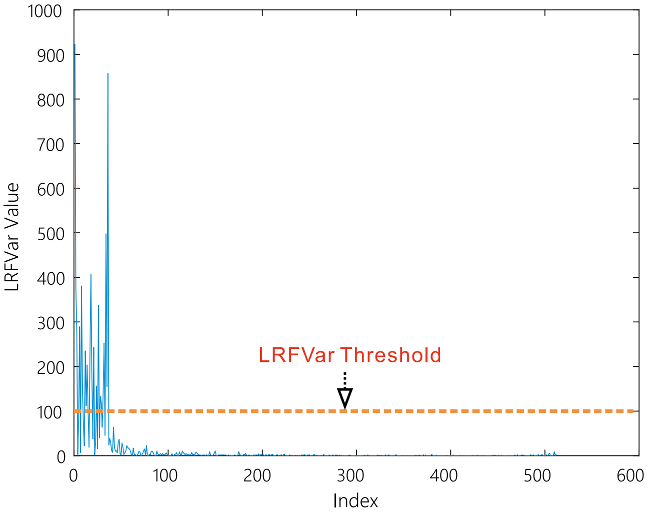


Fig. 4. In the *LRFVarList* plot of DS1, smooth lines symbolize the *LRF* level and sharp waves indicate sharp changes.

Table 1
The characteristics of synthetic datasets.

Synthetic data	Number of instances	Number of attributes	Number of outliers
DS1	1043	2	43
DS2	1000	2	85
DS3	1039	2	41
DS4	1641	2	45
DS5	876	2	77
DS6	1372	2	72
DS7	1037	2	37
DS8	2259	2	159
DS9	1034	2	36
DS10	2042	2	64
DS11	1020	2	26
DS12	1242	2	50

points in higher density areas are surrounded by neighboring points in a more uniform way, which results in a small magnitude for $\overrightarrow{LRF}(i,k)$. Fig. 2 provides an intuitive illustration of this observation. This paper examines the increase of *LRF* by sequentially increasing k. $LRF1$, $LRF2$, $LRF3$ and $LRF4$ respectively represent the *LRF* from one, two, three or four neighbors. Fig. 2 shows that there are differences in the *LRF* variation between Fig. 2(a) and (b). If it is an inner point, there is a small variation in the *LRF*. On the contrary, if it is an outlier or a boundary point, the *LRF* changes a lot. Through comparison of (a) and (b) in Fig. 2, this paper presents the magnitude of the resultant force on point $P2$ is much smaller than that of point $P1$.

As discussed in Section 3.1, the *LRF* of outliers or boundary points is proportional to the number of neighbors because directions of forces are basically the same. On the contrary, the *LRF* of inner points does not change with the increase of the number of neighbors due to the inconsistent directions of forces. Therefore, some quantities are defined in Eq. (6) to measure the *LRF* change rate.

$$\varDelta LRF(i,k) = \left| |\overrightarrow{LRF}(i,k)| - |\overrightarrow{LRF}(i,k+1)| \right|, k = 1, 2, \ldots, K-1 \tag{6}$$

where $\left|\overrightarrow{LRF}(i,k)\right|$ represents the magnitude of *LRF* for point i. K is the maximum number of nearest neighbors and its value ranges from 2 to 100. According to experiments, K of the larger value can be applied to the detection of global outliers, while K of the smaller value can be applied to the detection of local outliers. In this paper, $\varDelta LRF(i,k)$ is used to represent the fluctuation difference among outliers, boundary points and inner points. Therefore, through the summation of all $\varDelta LRF(i,k)$, we can quantify the cumulative fluctuation of *LRF* at a sequentially increasing value of k from 1 to $K-1$. The *LRF* change rate is defined in the following equation.

$$\varTheta LRF(i,K) = \sum_{k=1}^{K-1} \varDelta LRF(i,k) \tag{7}$$

Fig. 3 illustrates the corresponding LRF change rate for three types of points over synthetic dataset 1 (DS1). The detail of DS1 is shown in Fig. 6 and Table 1. As the graph shows, the *LRF* change rate is rather high for an outlier whereas small for boundary points and extremely small for inner points. Unlike other similar detection methods, the proposed detection model is not sensitive to parameter k because the cumulative fluctuation of *LRF* is quantified at a sequentially increasing value of k from 1 to $K-1$.

3.2. Outlier detection

In this section, we propose a novel local-gravitation outlier detection method (LGOD) which is based on the proposed detection model. In order to automatically detect outliers, the level partitioning method is applied in the search of the threshold of the *LRF* change rate variation.

Definition 1 (*The LRF Change Rate Variation*)**.** $LRFVar(i,j)$ namely the *LRF* change rate variation of point i respect to j, is defined as follows:

$$LRFVar(i,j) = |\Theta LRF(i,K) - \Theta LRF(j,K)| \tag{8}$$

Sort *LRF* in ascending order. With sorted *LRF* list, assume its size is n, for each sequence-adjacent $LRF(i)$ and $LRF(i+1)$ in $LRFList$, compute $LRFVar(i,j)$ for each point in dataset and then get $LRFVarList$ (denoted by $LRFVarList$) of size $n-1$. It is important to note that each element in $LRFVarList$ corresponds to two data points in dataset. Plot $LRFVarList$ of DS1 out. (As shown in Fig. 4)

Algorithm 1: *LGOD*.

Input: *D* (the data set), K (the maximum number of the nearest neighbors)

Output: Index1 (index of outliers)

```
Initialize: $k = 1$,Index1= $\varnothing$;
create a $KD$-tree $T$ from the dataset $D$;
For each point $i$ in $D$ using $T$
    For $k$ in 1 to $K$
        find the $k-th$ nearest neighbors of $p$
        Compute $LRF(i,k)$;
          IF $k > 1$
          Compute $\Delta LRF(i,k)$;
          End IF
      End For
      Compute $\Theta LRF(i,k)$;
End For
Get the $LRF$ variation list $LRFVarList$;
Threshold is obtained by formula (9);
For each point $i$ in $LOFVarList$
    IF $LOFVar(i,j) > \beta$
    Index1=[Index1; $i$];
    End IF
End For
Return Index1
```

In the *LRFVarList* plot, there are shape waves and relatively smooth lines. The sharp waves indicate that corresponding points are outliers. Therefore, the *LRFVarList* plot is split into two parts by a straight line and the part higher than the straight line can be treated as outliers. However, there is no need to construct the *LRFVarList* plot. According to this theory, we can get a threshold value and move objects whose *LRFVar* values are bigger than that out of the *LRFVarList*. Based on statistical characteristics [37], the threshold value is defined as:

$$\beta = EX(RVarList) + w \times SD(RVarList) \tag{9}$$

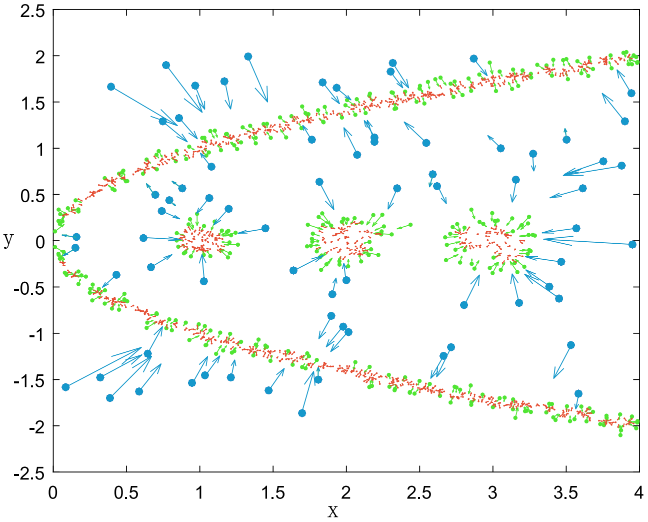


Fig. 5. *LRFs* on three types of points (outliers, boundary points and inner points). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

EX is mathematical expectation, *SD* is standard deviation and ω is a tuning coefficient. As shown in Fig. 4, sharp waves fluctuate slightly around *EX* of the *LRFVarList*, and the width proportion of sharp waves is far less than that of smooth lines. Therefore, we add a positive adjustment in *EX* to detect outliers. ω can be chosen from the range $(0,3]$. According to many experiments, $\omega = 2.5$ is an ideal value for most datasets. Therefore, we determine the value of ω to be 2.5 in LGOD.

Definition 2 (*Outlier*)**.** If the *LRFVar* value of a point is above threshold β, the point is an outlier.

Algorithm 1 summarizes the details of LGOD. Unlike existing density-based and distance-based methods, LGOD is insensitive to k. Furthermore, based on the level partitioning method, outliers can be selected automatically rather than determined manually.

3.3. Boundary detection

As shown in Fig. 5, blue points, green points and red points respectively represent outliers, boundary points and inner points. The lengths of arrows represent the magnitudes of *LRFs*. The magnitudes of *LRFs* are relatively small in central areas, while large in border areas of the clusters. According to Figs. 3 and 5, it is important to note that the *LRF* change rate of an outlier is greater than that of a boundary point. Thus top-m outliers are determined automatically through the level partitioning method. Except for top-m outliers, the greater the value of the LRF change rate is, the more likely the point is a boundary point. Therefore, on the basis of the foregoing, we propose a novel local-gravitation-based detection method of boundary points (LGBD).

Definition 3 (*Boundary Point*)**.** μ is the selection ratio of boundary points. If point i is in the first μ% of the descending order of ΘLRF and not an outlier, it is a boundary point.

Algorithm 2 presents details in the search of boundary points. Because the difference of *LRF* change rate between boundary points and inner points is not obvious, the selection ratio of boundary points needs manually input in LGBD, which is different from LGOD.

Table 2
Descriptive statistics of synthetic data set.

Synthetic Data	Maximum	Minimum	Standard deviation	Variance	Range	Kurtosis	Skewness
DS1	(34.8, 33.5)	(−4.7, −4.8)	(9.4, 9.7)	(88.8, 94.7)	(39.5, 38.2)	(2.1, 2.1)	(−0.6, −0.7)
DS2	(1.0, 9.8)	(10.6, −9.7)	(4.2, 4.2)	(17.9, 17.5)	(20.7, 19.4)	(2.3, 1.9)	(−0.3, 0.4)
DS3	(3.9, 3.5)	(−0.9, −0.8)	(1.0, 0.9)	(1.1, 0.8)	(4.8, 4.4)	(1.7, 2.8)	(−0.5, −1.1)
DS4	(10.0, 10.1)	(0.0, 0.0)	(3.5, 2.0)	(12.1, 4.0)	(10.0, 10.1)	(1.4, 5.2)	(0.3, −1.4)
DS5	(21.7, 20.9)	(−1.5, −0.1)	(7.1, 5.2)	(49.9, 27.6)	(23.2, 21.0)	(2.0, 2.2)	(−0.7, 0.0)
DS6	(4.0, 2.0)	(0.0, −2.1)	(1.1, 1.2)	(1.1, 1.5)	(4.0, 4.1)	(1.9, 1.7)	(0.0, 0.0)
DS7	(29.4, 30.8)	(−18.6, −20.9)	(11.6, 13.0)	(134.3, 168.7)	(48.0, 51.6)	(2.2, 2.5)	(0.3, 0.2)
DS8	(21.7, 20.7)	(−20.9, −21.0)	(7.5, 7.1)	(55.5, 50.0)	(42.6, 41.7)	(3.6, 3.6)	(0.2, −0.2)
DS9	(5.0, 6.0)	(−5.0, −5.9)	(3.0, 3.7)	(8.9, 13.8)	(10.0, 11.9)	(1.8, 1.5)	(0.1, 0.0)
DS10	(5.0, 5.9)	(−5.0, −5.8)	(2.9, 2.9)	(8.5, 8.2)	(10.0, 11.8)	(1.8, 1.7)	(0.0, 0.2)
DS11	(5.0, 6.1)	(−5.0, −5.9)	(2.9, 2.9)	(8.3, 8.4)	(10.0, 12.0)	(1.8, 1.9)	(0.0, 0.0)
DS12	(3.9, 4.0)	(−3.7, −4.0)	(2.0, 2.0)	(4.0, 4.1)	(7.6, 7.9)	(1.8, 1.8)	(0.0, 0.0)

3.4. Complexity analysis

The time complexity of LGOD depends on following parts: (a) the time for searching the kth nearest neighbors through the KD-tree is $O(n * logn)$, and where n is the number of datasets; (b) the computation of $\Theta LRF(i, K)$ needs $O(n * K)$; (c) there are three steps to find outliers in the level partitioning parts: getting the LRF variation, computing the threshold β and determining whether the point is an outlier, and the time complexity is $O(n)$, $O(1)$ and $O(n)$; Because K is the maximum number of neighbors and $(K \ll n)$. Therefore, the above analysis for LGOD reveals that the main cost of computation is the process of searching the kth neighbors. Therefore, the total time complexity of LGOD is $O(n * logn)$.

Algorithm 2: *LGBD*.

```
Input: D (the data set), K (the maximum number of nearest neighbors),
       μ (the selection ration of boundary points ) Index1 (index of outliers)
Output: Index2 (index of boundary points)
Initialize: $k = 1$,Index2= $\varnothing$;
create a $KD$-tree $T$ from the dataset $D$;
For each point $i$ in $D$ using $T$
    For $k$ in 1 to $K$
        find the $k-th$ nearest neighbors of $p$
        Compute $LRF(i, k)$;
          IF $k > 1$
          Compute $\Delta LRF(i, k)$;
          End IF
    End For
    Compute $\Theta LRF$;
End For
Descending the order of $\Theta LRF$.
Index2= (the first $\mu$% index of the descending list of $\Theta LRF$)-Index1;
Return Index2
```

The time complexity of LGBD depends on following parts: (a) the time for searching the kth nearest neighbors through the KD-tree is $O(n * logn)$, and where n is the number of datasets; (b) the computation of $\Theta LRF(i, K)$ needs $O(n*K)$; (c) the time complexity of sorting ΘLRF is $O(n*logn)$. Therefore, the total time complexity of LGBD is also $O(n * logn)$.

4. Experimental analysis

In this section, LGOD is evaluated in synthetic and real-world datasets. LOF [10], COF [25], LDOF [24], KNN [23], INS [22], ROCF [38] and GOD [5] are selected as representative detection methods of outliers, while ABOD [12], BEPS [13] and KNN [23]are chosen as comparative methods of LGBD. The experimental setup is introduced in Section 4.1, experimental results and analysis for LGOD are presented in Sections 4.2.1 and 4.2.2, and that for LGBD is presented in Section 4.2.3. The MATLAB code is available from https://github.com/xjnine/LGOD

Table 3
The characteristics of UCI datasets.

Real data	Number of instances	Number of attributes	Number of outliers	Source
Heart disease	153	13	3	[40]
Lymphography	148	18	6	[40]
Ionosphere	351	32	126	[40]
Breast cancer Wisconsin	683	10	239	[40]
Blood transfusion service center	748	5	178	[40]
SPECTF heart	267	44	55	[40]

4.1. Experimental setup

4.1.1. Datasets for LGOD

As shown in Table 1, there are 12 synthetic datasets [8,22]. Details of synthetic 12 datasets are shown in Fig. 6. The descriptive statistic of 12 synthetic datasets are shown in Table 2. Maximum, minimum, and range can demonstrate the data samples in which the distribution lies. The skewness indicates whether the distribution is symmetric or skewed. The kurtosis measures the thickness of the tails of the distribution and the standard deviation shows how the data samples scatter around the mean [39]. Details of three UCI real datasets are shown in Table 3.

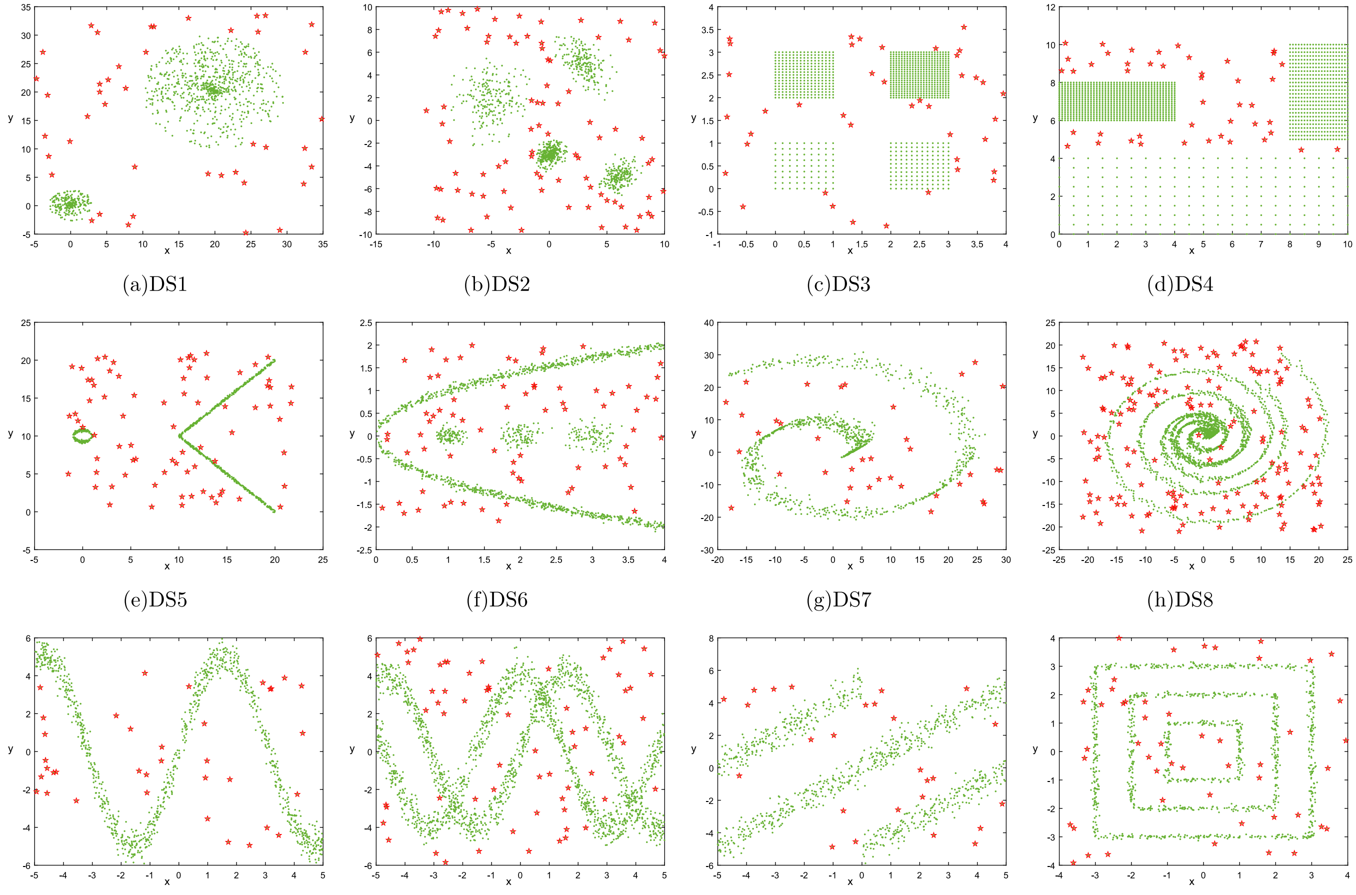
4.1.2. Evaluation measures

The final results of data descriptors can be summarized in four groups: the samples which are truly diagnosed as target ones (TP), the samples which are incorrectly diagnosed as target (FP), the samples correctly detected as outlier (TN), and finally the ones incorrectly recognized as outlier (FN) [41]. F1 is a popular index displaying the percentages of samples which are truly described as the result. The F1 can be computed using equation:

$$F1 = \frac{2 * TP}{2 * TP + FN + FP} \tag{10}$$

The use of overall accuracy might be inappropriate [42,43] since all these data sets are highly imbalanced. Although many evaluation metrics such as recall and precision have been proposed, the most popular evaluation measure in the literature on unsupervised outlier detection is based on a curve known as the Receiver Operating Characteristic (ROC), due to its origin in signal detection.

The curve is obtained by plotting for all possible choices of n the true positive rate (the proportion of outliers correctly ranked among the top n) versus the false positive rate (the proportion of inliers ranked among the top n). A random outlier ranking would result in a curve close to the diagonal, whereas a perfect ranking



(a)DS1 (b)DS2 (c)DS3 (d)DS4

(e)DS5 (f)DS6 (g)DS7 (h)DS8

(i)DS9 (j)DS10 (k)DS11 (l)DS12

Fig. 6. 12 original synthetic datasets.

(in which all outliers are ranked ahead of any inliers) would result in a curve consisting of a vertical line at false positive rate 0 and a horizontal line at the top of the plot (indicating a true positive rate of 1 for every false positive rate> 0) [44].

A ROC curve can be summarized by a single value known as ROC AUC which is defined as the area under the ROC curve (AUC). The AUC value ranges from 0 to 1. A perfect ranking of databased objects would result in an AUC value of 1, whereas an inverted perfect ranking would produce a value approaching 0. A random ranking of database object would result in an AUC value close to 0.5. The AUC is defined as:

$$AUC = \begin{cases} 1, & \text{if score}(o) > \text{score}(i) \\ \frac{1}{2}, & \text{if score}(o) = \text{score}(i) \\ 0, & \text{if score}(o) < \text{score}(i) \end{cases} \quad (11)$$

AUC value corresponds to the probability of pair (o, i), where o is true outlier, and i is inlier which are ordered correctly in the evaluated ranking (that is, with o appearing before i).

4.1.3. Data preprocessing

Data preprocessing has become an essential technique in current knowledge discovery scenarios. It aims at reducing the complexity inherent to real-world datasets, which can be easily processed by current data mining solutions. In data preprocessing phase of LGOD and LGBD, data normalization is achieved by min–max normalization which is defined as follows:

$$X_{norm} = (X - X_{min})/(X_{max} - X_{min}) \quad (12)$$

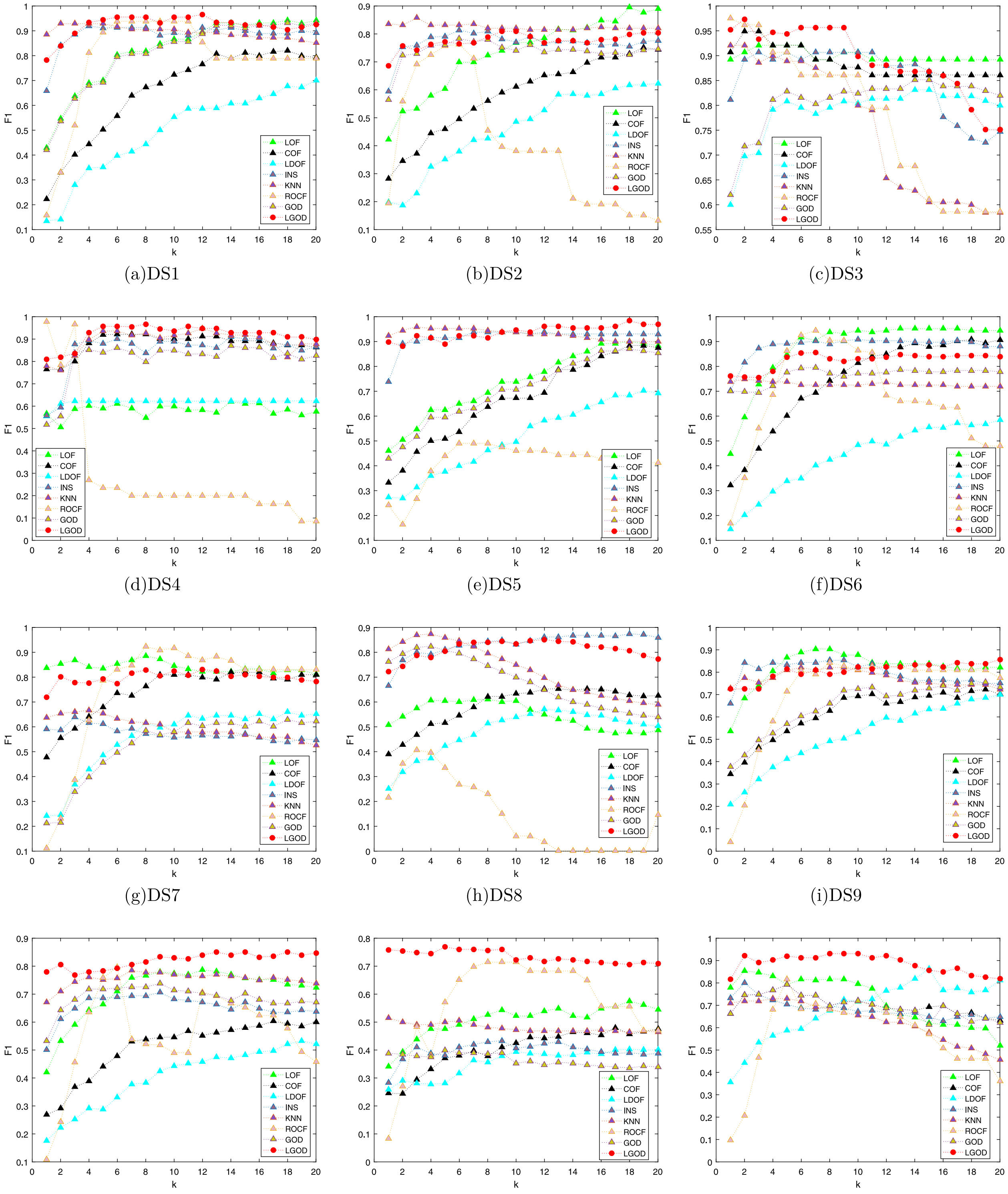
where X_{norm} is the value of point i. X_{max} is the maximum value and X_{min} is the minimum value of each corresponding attribute. Before the data is normalized, the missing attribute value is replaced by the means of attribute set.

4.2. Experimental results

4.2.1. Results of synthetic datasets for LGOD

Since 7 detection methods including the proposed method LGOD have a common model parameter k, the F1 values are computed at various k values from 2 to 20 and the AUC values are computed at various k values from 2 to 100. Experimental results in Fig. 7 show that single detection methods of outliers do not outperform other methods all the time. However, the overall performance and stability of LGOD are better than other methods. Especially, LGOD obviously outperforms other methods in DS1, DS3, DS4, DS5, DS9, DS10, DS11 and DS12. For other three datasets including DS2, DS7 and DS8, the performance of LGOD is as competitive as that of the best detection method.

Experimental results in Fig. 8 also show that single detection methods of outliers do not outperform other methods all the time. Although LGOD is not always the best, it is noteworthy that the overall performance and the stability of LGOD is better than other methods. Particularly, for DS2, DS5, DS9, DS10, DS11 and DS12, LGOD obviously outperforms the other methods. For other three datasets, including DS3, DS4 and DS6, the LGOD method shows competitive performance which is slightly worse than the best detection method. For DS1, AUC that is close to 0.92. In DS3 and DS4, the results of LDOF and LOF are worse than that of other methods. In DS5 and DS6 which have both spherical and manifold



(a)DS1 (b)DS2 (c)DS3

(d)DS4 (e)DS5 (f)DS6

(g)DS7 (h)DS8 (i)DS9

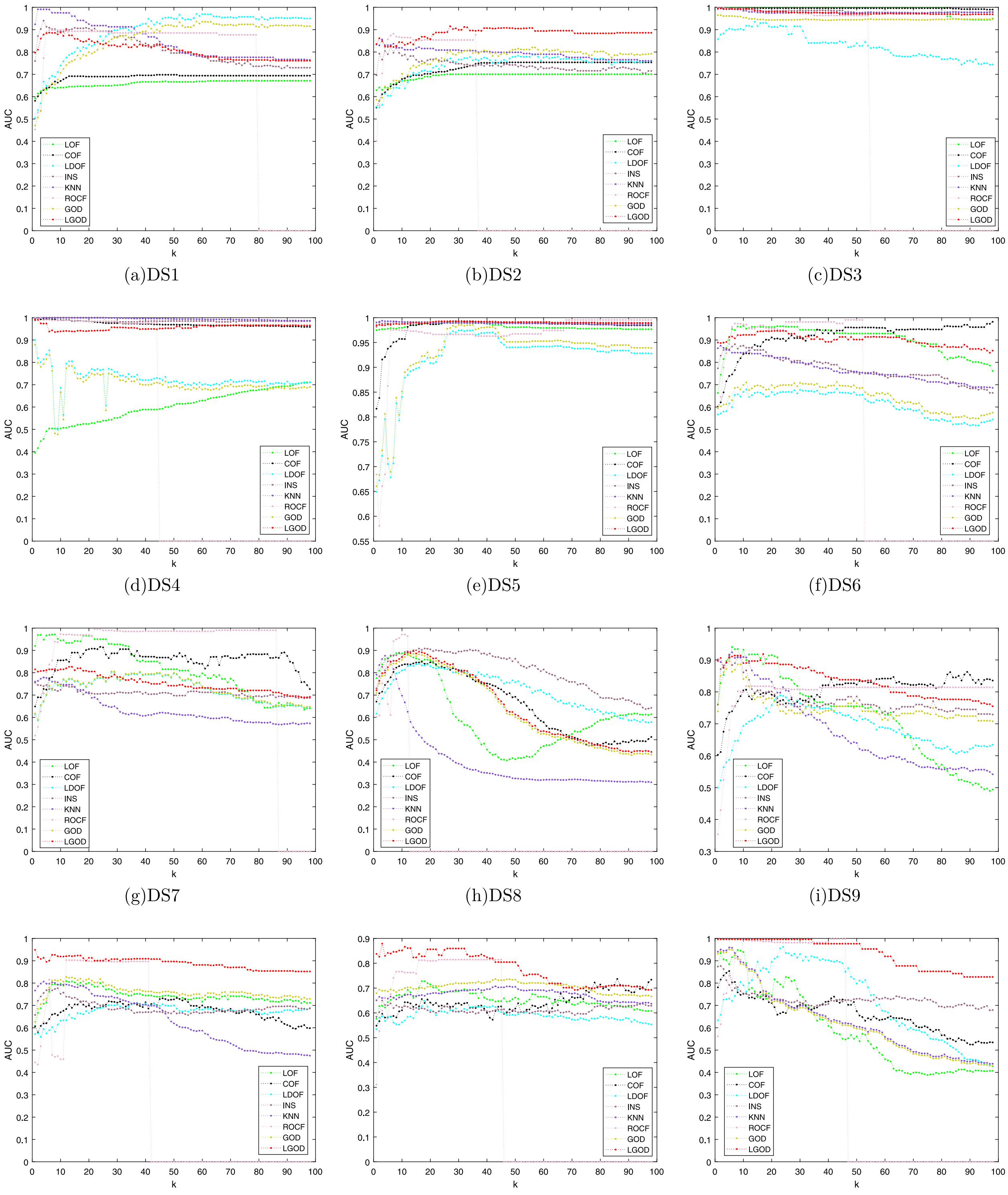
(j)DS10 (k)DS11 (l)DS12

Fig. 7. Detection performance (F1) of 8 methods for 12 original synthetic datasets.

clusters, LOF, COF and LGOD outperform other methods. The detection performances in DS7 are similar to that in DS6. In DS8, the INS method outperforms other methods. Experimental results of DS9 and DS10 indicate good and stable detection performance of LGOD. In DS11, LGOD performs well for small values of k, while other methods show poor detection performance. In DS12, other detection methods except for LGOD perform well for small values of k, but their detection performance radically deteriorates as the value of k increases. As shown in Fig. 9, the resulting AUC values at the given range of k are summarized as boxplots. LGOD shows a high average and small variance of AUC, which indicates that LGOD is precise and robust against the change of the parameter k.

4.2.2. *Results of real datasets for LGOD*

We also conduct outlier detection experiments on real datasets which have been previously used in research literature to evaluate the performance of various outlier detection methods.



(a)DS1 (b)DS2 (c)DS3

(d)DS4 (e)DS5 (f)DS6

(g)DS7 (h)DS8 (i)DS9

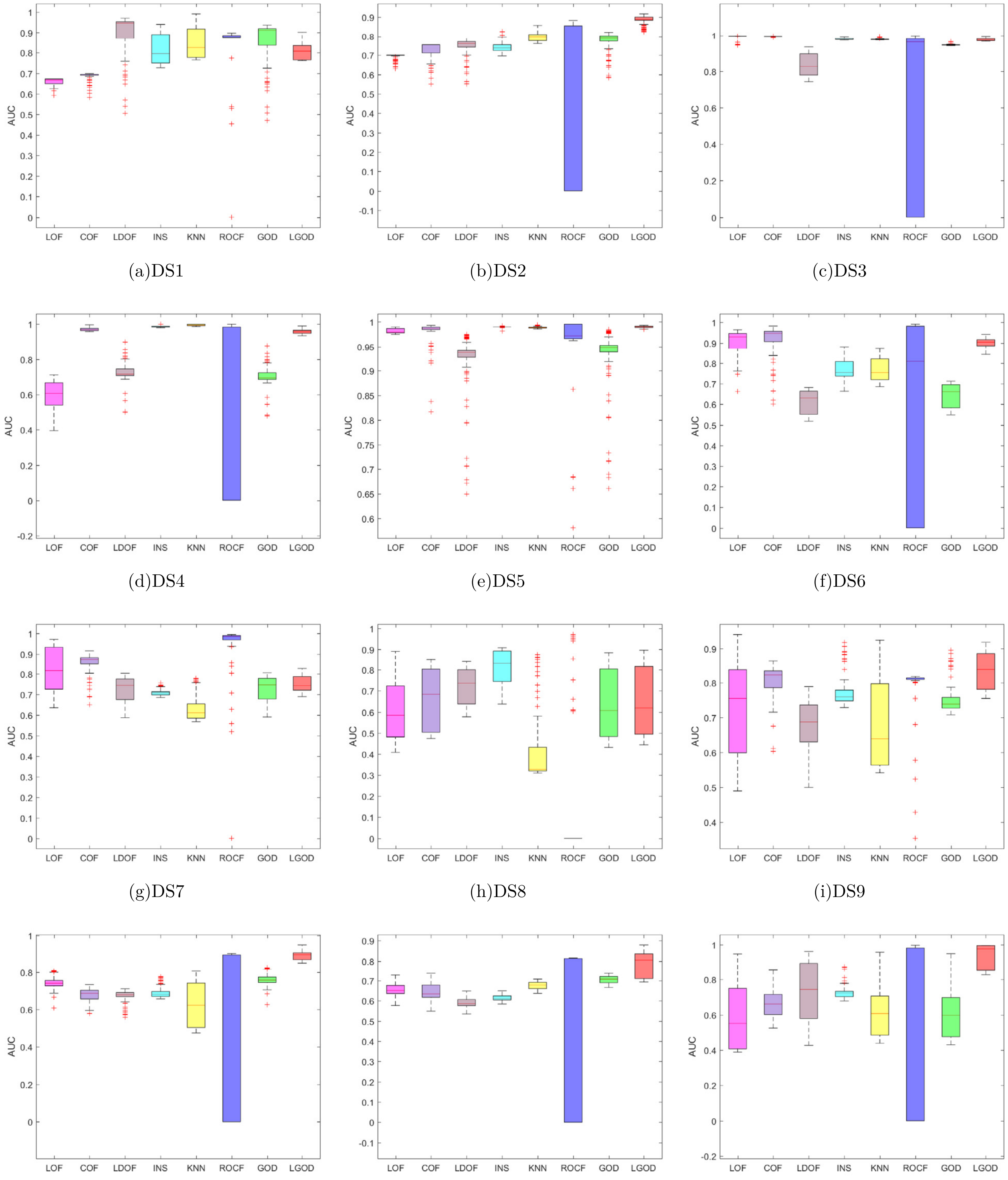
(j)DS10 (k)DS11 (l)DS12

Fig. 8. Detection performance (AUC) of 8 methods for 12 original synthetic datasets.

Figs. 10 and 11 shows experimental results of eight detection methods on real datasets. It is clear that our method LGOD exhibits superior detection performance with the metric of AUC of ROC. Particularly, LGOD achieves the best performance for three real datasets. As shown in Fig. 12, the AUC values are summarized as boxplots at the given range of *k*. The results of Fig. 12 indicate that LGOD shows a high average and small variance of AUC for real datasets, which implies that the proposed method LGOD shows a high and stable performance as compared to other methods.

4.2.3. Results of synthetic datasets for LGBD

To evaluate the performance of boundary point detection, we compare our method with KNN and BEPS methods. Because ABOD can distinguish three kinds of points by the spectra of angles for three types points, we also chose ABOD as a baseline method. Details of synthetic datasets are shown in Fig. 13 and



(a)DS1 (b)DS2 (c)DS3

(d)DS4 (e)DS5 (f)DS6

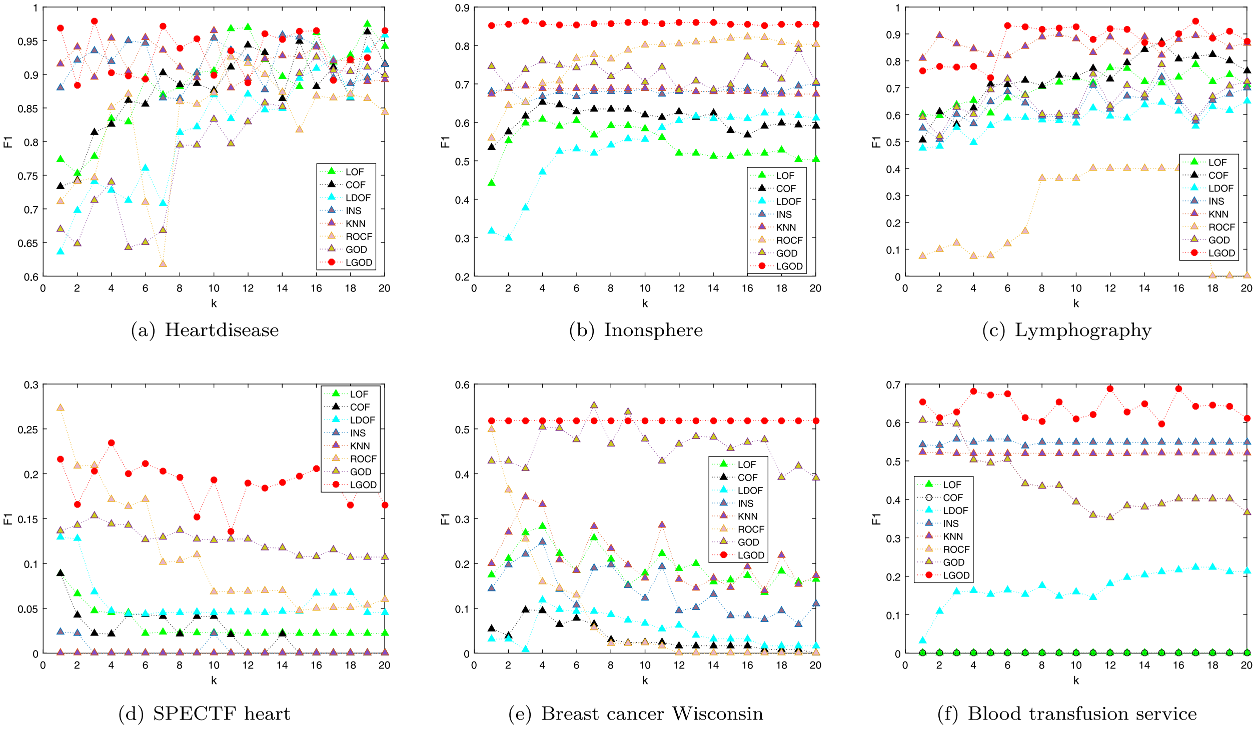
(g)DS7 (h)DS8 (i)DS9

(j)DS10 (k)DS11 (l)DS12

Fig. 9. AUC boxplots of 8 detection methods for 12 original synthetic datasets.

the parameter setting of each boundary point detection method in the two synthetic datasets is displayed in Table 4. μ is the selection ratio and k is the number of the nearest neighbors. BEPS has four parameters: the number of the nearest neighbors k, the number of NNs used for identifying edge patterns ke, ratio used to decide boundary patterns rb and the ratio used to decide boundary patterns rb and the ratio used to decide edge patterns re.

Although boundary detection methods can be applied to other applications, the criteria may be unsuitable for evaluating the performance. Furthermore, there are no publicly accepted standards to evaluate the quality of boundary point detection methods in unlabeled datasets. Therefore, for 2-D synthetic datasets, the simple way to illustrate the performance of boundary point detection methods is to show relative locations of boundary points in visible space [1]. The visualization of detection results for Flame and Aggregation are respectively illustrated Figs. 14 and



(a) Heartdisease

(b) Inonsphere

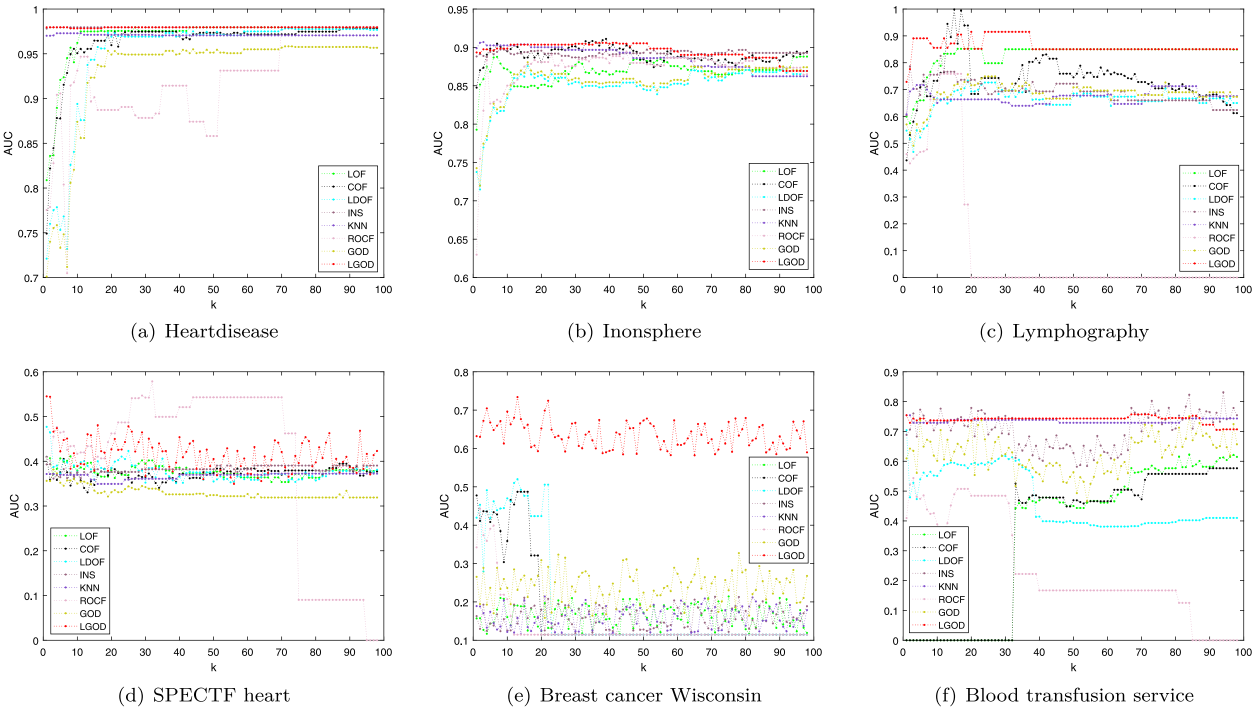
(c) Lymphography

(d) SPECTF heart

(e) Breast cancer Wisconsin

(f) Blood transfusion service

Fig. 10. Detection performance (F1) of 8 methods for real datasets.



(a) Heartdisease

(b) Inonsphere

(c) Lymphography

(d) SPECTF heart

(e) Breast cancer Wisconsin

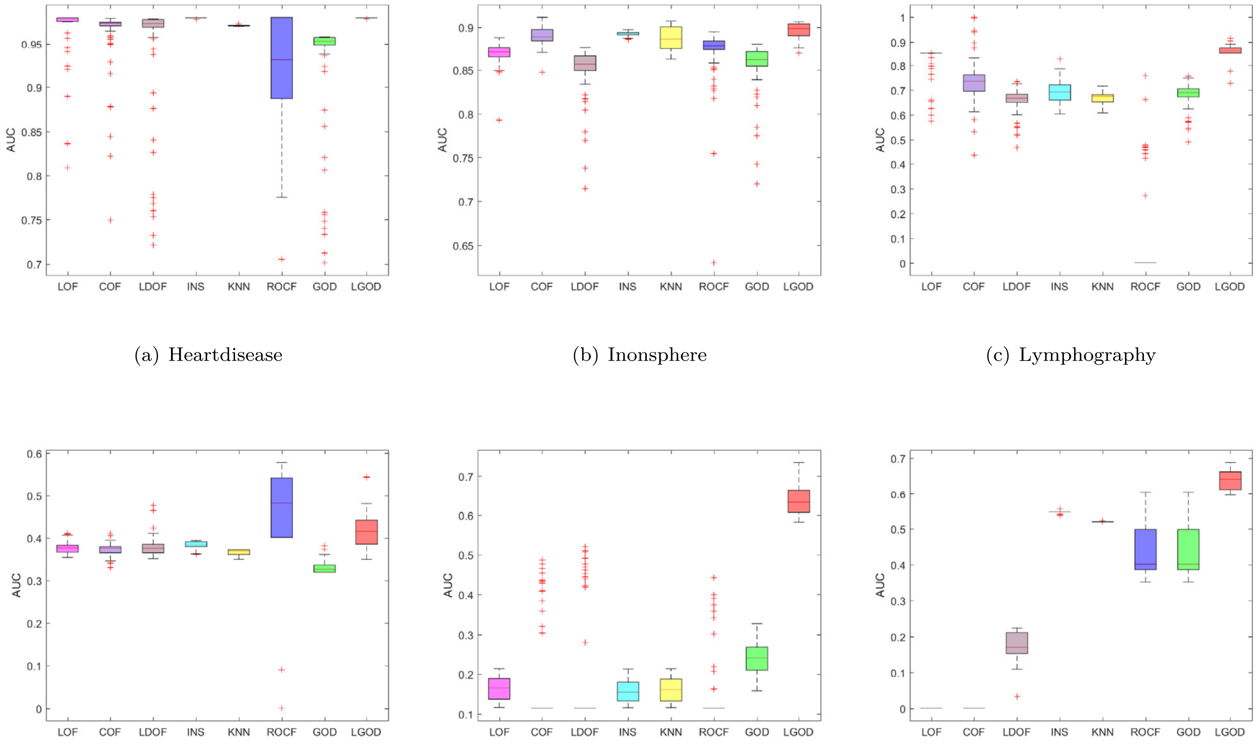
(f) Blood transfusion service

Fig. 11. Detection performance (AUC) of 8 methods for real datasets.

15. The blue points represent boundary points which are detected by different methods.

The effectiveness of boundary point detection methods is verified on the database shown in Fig. 13. Although most boundary points are detected by the ABOD method, some inner points are misidentified as boundary points. The boundary points detected by KNN is shown in Fig. 14(b), and about 20% of points in the Flame dataset are selected as boundary points. As shown in Fig. 14(c), BEPS can only identify part of boundary points. The detection results of Fig. 14(d) revel that boundary points are located near the margin of the Flame dataset.

In Fig. 15, detection methods on the Aggregation dataset are evaluated to demonstrate their effectiveness. It is clear that the blue points are located near the margin of the Aggregation dataset by LGBD. Similar to the previous dataset Flame, many inner points are misidentified as boundary points by ABOD and KNN. BEPS cannot find all boundary points. Compared with related methods, LGBD better reflects the characteristics of the data.



(a) Heartdisease

(b) Inonsphere

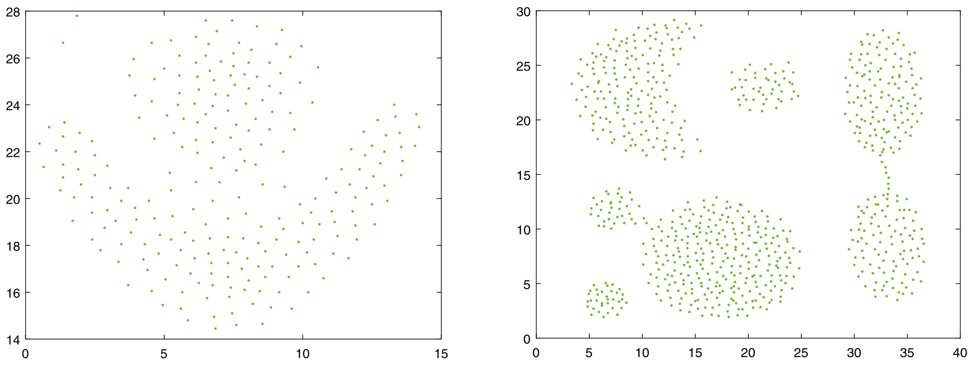
(c) Lymphography

(d) SPECTF heart

(e) Breast cancer Wisconsin

(f) Blood transfusion service

Fig. 12. AUC boxplots of 8 detection methods for real datasets.



(a)Flame

(b)Aggregation

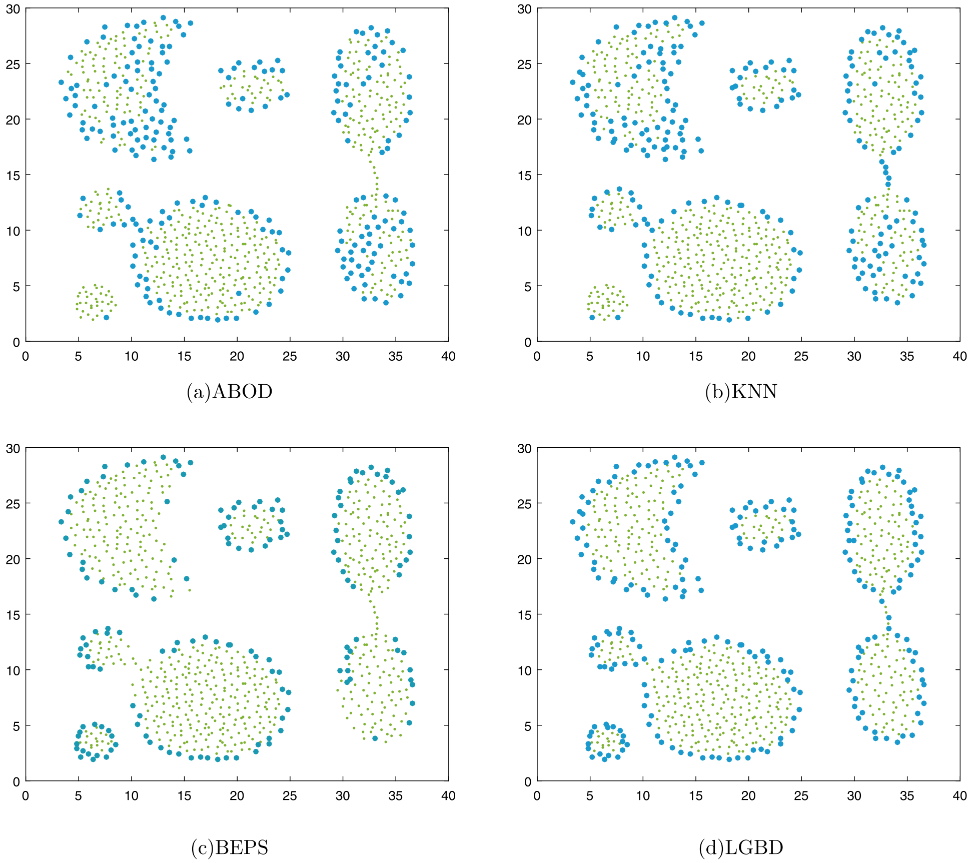
Fig. 13. 12 original synthetic datasets.

Table 4
The parameter setting of each clustering algorithm on synthetic datasets.

Synthetic data	KNN	ABOD	BEPS	LGBD
Flame	$\mu = 20\%, k = 9$	$\mu = 20\%$	$k = 13, ke = 29$ $rb = 70, ke = 90$	$\mu = 20\%$
Aggregation	$\mu = 25\%, k = 10$	$\mu = 25\%$	$k = 15, ke = 35$ $rb = 70, ke = 90$	$\mu = 25\%$

5. Conclusion

This paper presents a local-gravitation model that can be used to simultaneously identify both outliers and boundary points. The proposed model exploits the difference of the LRF change rate to distinguish outliers, boundary points and inner points in a dataset. This paper reveals that the LRF change rate to pairs of points remain rather high for an outlier whereas the LRF change rate is smaller for boundary points and extremely small for inner points. Note that the LRF change rate of an outlier is greater than that of a boundary point. Therefore, a novel local-gravitation-based outlier detection method (LGOD) is proposed. LGOD are not sensitive to the parameter k. Moreover, LGOD can automatically identify outliers by exploiting the level partitioning method. The greater the value of the LRF change rate, the most likely the point is a boundary point. In order to detect boundary points, a novel local-gravitation-based boundary point detection method (LGBD) is proposed. Compared with related methods, our proposed model better reflects the characteristics of the data. Furthermore, the proposed model can simultaneously identify outliers and boundary points. The experimental results in synthetic and real datasets show that the effectiveness and efficiency of our method.



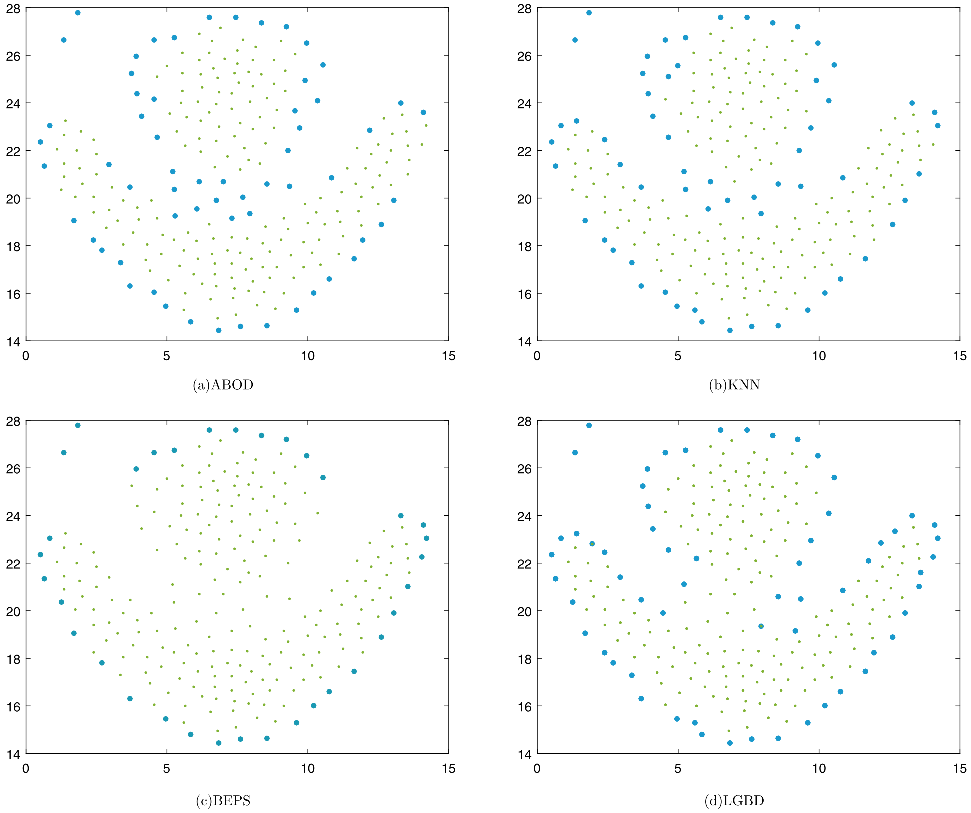
(a)ABOD

(b)KNN

(c)BEPS

(d)LGBD

Fig. 14. The relative locations of boundary points in visible space for Flame. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a)ABOD

(b)KNN

(c)BEPS

(d)LGBD

Fig. 15. The relative locations of boundary points in visible space for Aggregation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In fact, there also exist distinct differences among LRF's directions of the data points close to the cluster centers, outlier and boundary point. Therefore, the following research can consider the magnitudes and direction of LRF at the same time.

CRediT authorship contribution statement

Jiang Xie: Conceptualization, Writing - original draft, Writing - review & editing. **Zhongyang Xiong:** Data curation. **Qizhu Dai:** Software. **Xiaoxia Wang:** Validation. **Yufang Zhang:** Data curation.

Acknowledgments

We are obliged to Ming Ming Zhang for giving advice on our manuscript. This work is partly funded by the national nature science foundation of China (no. 61762025), the project of frontier and application foundation research program of CQ CSTC, China (no. cstc2017jcyjAX0340) and CERNET Next Generation Internet Technology Innovation Project, China (NGII 20150706).

References

[1] X. Li, J. Lv, Z. Yi, An efficient representation-based method for boundary point and outlier detection, IEEE Trans. Neural Netw. 29 (1) (2018) 51–62.
[2] R. Domingues, M. Filippone, P. Michiardi, J. Zouaoui, A comparative evaluation of outlier detection algorithms: experiments and analyses, Pattern Recognit. 74 (2018) 406–421.
[3] K.V.V. Chandola, A. Banerjee, Outlier detection using k-nearest neighbour graph, ACM Comput. Surv. 41 (3) (2009) 1–58.
[4] C.C. Aggarwal, Outlier Analysis, second ed., 2016.
[5] S. Bharti, K.K. Pattanaik, Gravitational outlier detection for wireless sensor networks: gravitational outlier detection for wireless sensor networks, Int. J. Commun. Syst. 29 (13) (2016) 2015–2027.
[6] Z. Wang, Z. Yu, C.L.P. Chen, J. You, T. Gu, H.S. Wong, J. Zhang, Clustering by local gravitation, IEEE Trans. Cybern. PP (99) (2017) 1–14.
[7] D.M. Hawkins, Identification of outliers, 1980.
[8] J. Ha, S. Seok, J.S. Lee, A precise ranking method for outlier detection, Inform. Sci. 324 (C) (2015) 88–107.
[9] C.C. Aggarwal, An introduction to outlier analysis, 2017.
[10] M.M. Breunig, H.P. Kriegel, R.T. Ng, J. Sander, Lof: Identifying density-based local outliers, in: Acm Sigmod International Conference on Management of Data, 2000, pp. 93–104.
[11] SimonByers, A. Raftery, Nearest-neighbor clutter removal for estimating features in spatial point processes, Publ. Amer. Statist. Assoc. 93 (442) (1998) 577–584.
[12] H.P. Kriegel, M. Schubert, A. Zimek, Angle-based outlier detection in high-dimensional data, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 444–452.
[13] L. Yuhua, M. Liam, Selecting Critical Patterns Based on Local Geometrical and Statistical Information, IEEE Computer Society, 2011, pp. 1189–1201.
[14] W.A. Shewhart, Economic quality control of manufactured product, Bell Syst. Tech. J. 9 (2) (1930) 364–389.
[15] G. Ratsch, S. Mika, B. Scholkopf, K.R. Muller, Constructing boosting algorithms from svms: an application to one-class classification, IEEE Trans. Pattern Anal. Mach. Intell. 24 (9) (2002) 1184–1199.
[16] Y. Liu, Y. Liu, Y. Chen, Fast support vector data descriptions for novelty detection, IEEE Trans. Neural Netw. 21 (8) (2010) 1296–1313.
[17] X. Peng, D. Xu, Efficient support vector data descriptions for novelty detection, Neural Comput. Appl. 21 (8) (2012) 2023–2032.
[18] R. Sadeghi, J. Hamidzadeh, Automatic support vector data description, Soft Comput. 22 (1) (2016) 1–12.
[19] J. Hamidzadeh, R. Sadeghi, N. Namaei, Weighted support vector data description based on chaotic bat algorithm, Appl. Soft Comput. (2017).
[20] T. Ide, H. Kashima, Eigenspace-based anomaly detection in computer systems, in: Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 440–449.
[21] Z. He, S. Deng, X. Xu, J.Z. Huang, A fast greedy algorithm for outlier mining, Lecture Notes in Comput. Sci. 3918 (2005) 567–576.
[22] J. Ha, S. Seok, J.S. Lee, Robust outlier detection using the instability factor, Knowl.-Based Syst. 63 (2) (2014) 15–23.
[23] S. Ramaswamy, R. Rastogi, K. Shim, T. Korea, Efficient algorithms for mining outliers from large data sets, in: ACM SIGMOD International Conference on Management of Data, 2000, pp. 427–438.
[24] K. Zhang, M. Hutter, H. Jin, A new local distance-based outlier detection approach for scattered real-world data, Knowl. Discov. Data Min. (2009) 813–822.
[25] J. Tang, Z. Chen, A.W. Fu, D.W. Cheung, Enhancing effectiveness of outlier detections for low density patterns, Knowl. Discov. Data Min. (2002) 535–548.
[26] C. Xia, W. Hsu, M.L. Lee, B.C. Ooi, Border: Efficient computation of boundary points, IEEE Trans. Knowl. Data Eng. 18 (3) (2006) 289–303.
[27] Y. Li, Selecting training points for one-class support vector machines, Pattern Recognit. Lett. 32 (11) (2011) 1517–1522.
[28] X. Ding, Y. Li, A. Belatreche, L.P. Maguire, Novelty detection using level set methods, IEEE Trans. Neural Netw. 26 (3) (2015) 576–588.
[29] W.E. Wright, Gravitational clustering, Pattern Recognit. 9 (3) (1977) 151–166.
[30] J. Gomez, D. Dasgupta, O. Nasraoui, A new gravitational clustering algorithm, 2003, pp. 83–94.
[31] J. Gomez, O. Nasraoui, E. Leon, Rain: data clustering using randomized interactions between data points, 2004, pp. 250–255.
[32] S. Kundu, Gravitational clustering: a new approach based on the spatial distribution of the points, Pattern Recognit. 32 (7) (1999) 1149–1160.
[33] H.Q.T. Zhang, An improved clustering algorithm, in: Proc. 3rd Int. Symp. Comput. Sci. Comput. Technol.(ISCSCT), Jiaozuo, China, 2010, pp. 112–115.
[34] M.A. Sanchez, O. Castillo, J.R. Castro, P. Melin, Fuzzy granular gravitational clustering algorithm for multivariate data, Inform. Sci. 279 (2014) 498–511.
[35] E. Rashedi, H. Nezamabadipour, S. Saryazdi, Gsa: A gravitational search algorithm, Inform. Sci. 179 (13) (2009) 2232–2248.
[36] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, Inform. Sci. 222 (2013) 175–184.
[37] S.H. Yue, L. Ping, J.D. Guo, S.G. Zhou, Using greedy algorithm: Dbscan revisited ii, J. Zhejiang Univ. Sci. 5 (11) (2004) 1405–1412.
[38] J. Huang, Q. Zhu, L. Yang, D.D. Cheng, Q. Wu, A novel outlier cluster detection algorithm without top-n parameter, Knowl.-Based Syst. 121 (2017) 32–40.
[39] R. Sadeghi, T. Banerjee, W. Romine, Early hospital mortality prediction using vital signals, Smart Health (2018) S2352648318300357.
[40] L.M. Bache K, Uci machine learning repository.
[41] Zabihimayvan, Mahdieh, S. Reza, R.H. Nathan, D. Derek, A soft computing approach for benign and malicious web robot detection, Expert Syst. Appl. 87 (2017) 129–140.
[42] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[43] H. He, Y. Bai, E.A. Garcia, S. Li, Adasyn: Adaptive synthetic sampling approach for imbalanced learning, in: IEEE International Joint Conference on Neural Networks, 2008, pp. 1322–1328.
[44] G.O. Campos, A. Zimek, J. Sander, R.J.G.B. Campello, B. Micenková, E. Schubert, I. Assent, M.E. Houle, On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study, Data Min. Knowl. Discov. 30 (4) (2016) 891–927.