

# A non-parameter outlier detection algorithm based on Natural Neighbor



Jinlong Huang, Qingsheng Zhu\*, Lijun Yang, Ji Feng

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

## ARTICLE INFO

### Article history:

Received 2 April 2015

Revised 8 October 2015

Accepted 9 October 2015

Available online 30 October 2015

### Keywords:

Outlier detection

Natural Neighbor

Natural Outlier Factor

## ABSTRACT

Outlier detection is an important task in data mining with numerous applications, including credit card fraud detection, video surveillance, etc. Although many Outlier detection algorithm have been proposed. However, for most of these algorithms faced a serious problem that it is very difficult to select an appropriate parameter when they run on a dataset. In this paper we use the method of Natural Neighbor to adaptively obtain the parameter, named Natural Value. We also propose a novel notion that Natural Outlier Factor (NOF) to measure the outliers and provide the algorithm based on Natural Neighbor (NaN) that does not require any parameters to compute the NOF of the objects in the database. The formal analysis and experiments show that this method can achieve good performance in outlier detection.

© 2015 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]. At present, the studies on outlier detection is very active. Many outlier detection algorithms have been proposed. Outlier detection algorithm can be roughly divided into distribution-based, depth-based, distance-based, clustering-based and density-based act.

In distribution-based methods, the observations that deviate from a standard distribution are considered as outliers [7]. But distribution-based methods not applicable to dataset that multi-dimensional or the distribution unknown. The depth-based [10,11] methods can improve 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 dataset. In clustering-based methods, the outliers are by-products of clustering, such as DBSCAN [12], CLARANS [13], CHAMELEON [14], BIRCH [15], and CURE [16]. But the target of clustering-based methods is finding clusters, not detecting outliers, so the efficiency of detecting outliers is low too.

The distance-based algorithms was widely used for the 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, since distance-based algorithms do not 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 local outliers have received much attention recently. The density-based methods can solve this problem well. And many density-based outlier detection algorithms have been proposed. In paper [17], authors define the concept of a local outlier factor (LOF) that a measure of outlier degree in density between an object and its neighborhood objects. The article [18] made an improved on LOF and proposed an outlier detection algorithm, which defined the influenced outlieriness (INFLO) computed by considering both neighbors and reverse neighbors as the outlier degree. This results in a meaningful outlier detection.

Given our motivation, through the above analysis, although the density-based methods can solve problem of local outliers well, density-based methods face the same problem that parameter selection as the first four methods. All of these algorithms almost cannot effectively detect the outliers without appropriate parameter. In other words, most of these algorithms have high dependency to the parameter. Once the parameter changed, the result of outlier detecting would have obvious difference. So the selection of parameter is very important for outlier detection algorithm. In fact, however, determination of parameter is dependent on the knowledge of researcher's experience and a lot of experiment. For example, it is difficult to select an appropriate parameter  $k$  that the number of neighbors when use LOF or INFLO to detect the outlier on database.

More detailed analysis of the problem with existing approaches can be available in paper [19]. Paper [19] also propose a new outliers detection algorithm (INS) using the instability factor. INS is

\* Corresponding author. Tel.: +86 2365105660; fax: +86 2365104570.

E-mail address: [qs Zhu@cqu.edu.cn](mailto:qs Zhu@cqu.edu.cn) (Q. Zhu).

insensitive to the parameter  $k$  when the value of  $k$  is large as shown in Fig. 7(c). However, the cost is that the accuracy is low when the accuracy stabilized. Moreover INS hardly find a properly parameter to detect the local outliers and global outliers simultaneously. In other words, when the value of  $k$  is well to detect the global outliers, the effect on local outliers detection is bad, and vice versa.

In this paper, in order to solve the above problem, we first introduce a novel concept of neighbor named Natural Neighbor (NaN) and its search algorithm (NaN-Searching). Then we obtain the number of neighbors, the value of parameter  $k$ , use the NaN-Searching algorithm. We also define a new concept of Natural Influence Space (NIS) and Natural Neighbor Graph (NNG), and compute the Natural Outlier Factor (NOF). The bigger the value of NOF is, the greater the possibility of object is outlier.

The paper is organized as follows. In Section 2, we present the existing definition and our motivation. In Section 3, properties of Natural Neighbor are introduced. In Section 4, we propose a outlier detection algorithm based on Natural Neighbor. In Section 5, a performance evaluation is made and the results are analyzed. Section 6 concludes the paper.

## 2. Related work

In this section, we will briefly introduce concept of LOF and INS. LOF is a famous density-based outlier detection algorithm. And INS is a novel outlier detection algorithm proposed in 2014. Interested readers are referred to papers [17] and [19].

Let  $D$  be a database,  $p, q$ , and  $o$  be some objects in  $D$ , and  $k$  be a positive integer. We use  $d(p, q)$  to denote the Euclidean distance between objects  $p$  and  $q$ .

**Definition 1** ( $k$ -distance and nearest neighborhood of  $p$ ). The  $k$ -distance of  $p$ , denoted as  $k_{dist}(p)$ , is the distance  $d(p, o)$  between  $p$  and  $o$  in  $D$ , such that:

- (1) For at least  $k$  objects  $o' \in D/\{p\}$  it holds that  $d(p, o') \leq d(p, o)$ , and
- (2) For at most  $(k - 1)$  objects  $o' \in D/\{p\}$  it holds that  $d(o, o') < d(p, o)$

The  $k_{dist}(p)$  can reflect the density of the object  $p$ . The smaller  $k_{dist}(p)$  is, the much denser the area around  $p$  is.

**Definition 2** ( $K$ -Nearest neighborhood). The  $k$ -nearest neighborhood of  $p$ ,  $NN_k(p)$  is a set of objects  $X$  in  $D$  with  $d(p, X) \leq k_{dist}(p)$ :  $NN_k(p) = \{X \in D/\{p\} | d(p, X) \leq k_{dist}(p)\}$ .

Note that the number of objects in  $k$ -nearest neighborhood of  $p$  may be more than  $k$ . In other words, there may be more than  $k$  objects within  $NN_k(p)$ .

**Definition 3** (Reachability distance of  $p$  w.r.t object  $o$ ). The reachability distance of object  $p$  with respect to object  $o$  is defined as follows:

$$reach - dist_k(p, o) = \max\{k - distance(o), d(p, o)\} \quad (1)$$

**Definition 4** (Local reachability density of  $p$ ). The local reachability density of an object  $p$  is the inverse of the average reachability distance from the  $k$ -nearest-neighbors of  $p$ . This is defined as follows:

$$lrd_k(p) = 1 / \frac{\sum_{o \in NN_k(p)} reach - dist_k(p, o)}{|NN_k(p)|} \quad (2)$$

Essentially, the local reachability density of an object  $p$  is the reciprocal of the average distance between  $p$  and the objects in its  $k$ -neighborhood. Based on local reachability density, paper [17] defines the local outlier factor as follows:

$$LOF_k(p) = \frac{\sum_{o \in NN_k(p)} \frac{lrd_k(o)}{lrd_k(p)}}{|NN_k(p)|} \quad (3)$$

Obviously, LOF is the average of the ratios of the local reachability density of  $p$  and  $p$ 's  $k$ -nearest-neighbors. We can think about it in this way that LOF is the ratios of the local reachability density of  $p$  and the average local reachability density of  $p$ 's  $k$ -nearest-neighbors. Intuitively,  $p$ 's local outlier factor will be very high if its local reachability density is much lower than those of its neighbors. In this way, the bigger  $p$ 's local outlier factor is, the more likely  $p$  is outlier.

Although LOF has been used widely, there are some problem existed in it. It is the main problem that LOF is sensitive to parameters. To solve this problem, paper [19] proposed a new algorithm (INS) using the instability factor. The follows are some briefly introduce to INS.

**Definition 5** (The  $k$  center of gravity). The  $k$  center of gravity of  $p$  is defined as a centroid of the objects in  $NN_k(p)$ , which is given by

$$m_k(p) = \frac{1}{k+1} \sum_{q \in NN_k(p)} X_q \quad (4)$$

where  $X_q = (x_{q1}, x_{q2}, \dots, x_{qd})$  is the coordinates of the object  $q$  observed in a  $d$ -dimensional space (under the assumption that the space is Euclidean).

Let  $d_i(p)$  denote the distance between  $m_i(p)$  and  $m_{(i+1)}(p)$ , which is defined by the following equation:

$$d_i(p) = d(m_i(p), m_{(i+1)}(p)), i = 1, 2, \dots, k-1 \quad (5)$$

**Definition 6** (Absolute difference). The absolute difference between  $\theta_i(p)$  and  $\theta_{(i+1)}(p)$ , denoted as  $\Delta\theta_i(p)$ , which is defined as:

$$\Delta\theta_i(p) = |\theta_i(p) - \theta_{(i+1)}(p)|, i = 1, 2, \dots, k-2 \quad (6)$$

**Definition 7** (Instability factor). The instability factor, and  $INS(p, k)$  are defined by the following equation:

$$INS(p, k) = \sum_{i=1}^{k-2} \Delta\theta_i(p) \quad (7)$$

INS improve the problem that are sensitive to parameter. The changes of accuracy, using INS, were minor when the parameter is changed. And INS can be flexibly used for both local and global detection of outliers by controlling its parameter. But the accuracy is not high when the accuracy stabilized. Moreover INS hardly find a properly parameter to detect the local outliers and global outliers simultaneously.

Though above analysis, we know that LOF and INS have their own advantages and disadvantages. However, no matter LOF or INS, there is a same problem that parameter selection. It is difficult to select an appropriate parameter  $k$  that the number of neighbors.

In order to solve the problem that parameter selection, we introduce the concept of Natural Neighbor. Natural Neighbor can adaptively obtain the appropriate value of  $k$  that the number of neighbors without any parameters. The outlier detection algorithm that we proposed in this paper use the  $k$  as parameter to detect the outliers. In the section that follows, a detailed introduction will be made to Natural Neighbor.

## 3. Natural Neighbor and NOF algorithm

### 3.1. NaN definition and algorithm

Natural Neighbor is a new concept of neighbor. The concept originates in the knowledge of the objective reality. The number of one's real friends should be the number of how many people are taken him or her as friends and he or she take them as friends at the same time. For data objects, object  $y$  is one of the Natural Neighbor of object  $x$  if object  $x$  considers  $y$  to be a neighbor and  $y$  considers  $x$  to be a neighbor at the same time. In particular, data points lying in sparse region should have small number of neighbors, whereas data points lying in

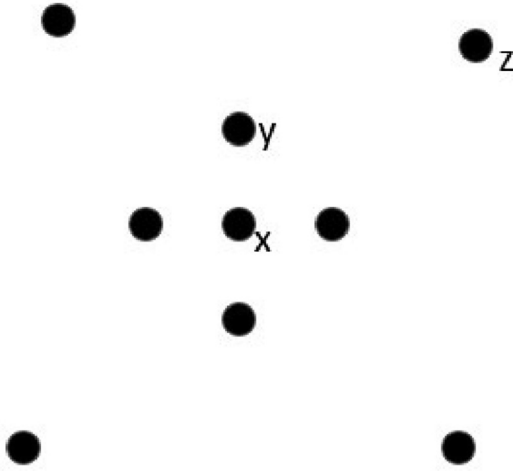


Fig. 1. Sparse and dense.

dense region should have large number of neighbors. Here note that “sparse” and “dense” is comparative. As shown in Fig. 1,  $y$  is sparser than  $x$ , but  $y$  lies in a dense region relative to  $z$ .

The key idea of Natural Neighbor is that the object lying in sparse region should possess little energy, whereas object lying in dense region should possess higher power. The whole computational procedure of Natural Neighbor can automatically fulfilled without any parameters. If the formation of  $K$ -Nearest Neighbor is regarded as an active neighbor searching procedure, then the forming of Natural Neighbor is completely passive.

The search cost of KNN and RNN for each object in the database is huge. So we introduce the KD-tree into the Natural Neighbor searching. Here we do not gave a detailed description of KD-tree. Detailed description can be found in paper [20]. The Natural Neighbor searching algorithm as the Algorithm 1.

**Algorithm 1** NaN-Searching(SetOfPoints) SetOfPoints is UNCLASSIFIED.

- (1) Initializing:  $r = 1, Rnb(i) = 0, NN_r(i) = \emptyset, RNN_r(i) = \emptyset, NaN(i) = \emptyset$
- (2)  $kdtree = creatKDTree(SetOfPoints)$  //create a KD-tree
- (3) Use kdtree to find the  $r$ th neighbor  $y$  for each data point  $x$ .
  - a.  $Rnb(y) = Rnb(y) + 1$
  - b.  $NN_r(x) = NN_r(x) \cup \{y\}$
  - c.  $RNN_r(y) = RNN_r(y) \cup \{x\}$
- (4) Compute the number of data point  $x$  that  $Rnb(x) = 0$ 
  - a. If the number does not changed for 3 times goto step 5
  - b. else
    - $r = r + 1$  and goto step 3
- (5)  $sup_k = r$  and output the max  $Rnb(i)$

$Rnb(i)$  is the times that point  $i$  is contained by the neighborhood of other points, which the number of  $i$ 's reverse neighbor.  $NN_r(x)$  is the  $r$ -neighborhood.  $RNN_r(y)$  is the  $r$ -reverse-neighborhood.  $sup_k$  is the average value of the number of each point's neighbors, called Natural Eigenvalue. The value size of  $sup_k$  to a certain stand the complexity of the distribution of data objects. If the distribution of data objects is regular or the size of dataset is small, the value of  $sup_k$  is small. On the contrary, the value of  $sup_k$  will be corresponding increased. However, no matter how increased the value of  $sup_k$  is, the value is still much smaller than the size of dataset. Since KD-tree is introduced into NaN-Searching, the time complexity of NaN-searching algorithm is  $O(N^* \log N)$ .  $N$  is the number of data in SetOfPoints.

**Definition 8** (Natural Neighbor – NaN). Based on the Natural Neighbor searching algorithm, the following will be shown, if point  $x$  be-

longs to the neighbors of point  $y$  and  $y$  belongs to the neighbors of point  $x$ , then  $x$  is called as  $y$ 's Natural Neighbor (NaN). In the same way,  $y$  is Natural Neighbor of  $x$ .

Compared with the published concept of neighbor that has been used widely [21],  $k$ -nearest neighbor proposed by Stevens [22], the main difference is that Natural Neighbor is a scale-free concept of neighbor, the great advantage is that the search method of Natural Neighbor is non-parameter. Scale-free means that the number of neighbors for each object are not necessarily identical. However, the number of neighbors for each object are equal, the number is  $k$ , in the concept of  $k$ -nearest neighbor.

### 3.2. Natural Neighborhood Graph

Through the above NaN-Searching algorithm, we can obtain two eigenvalue  $Rnb(i)$  and  $sup_k$ . Therefore, we can define different neighborhood graph by connected to each point in a different way, as the following definition.

**Definition 9** (NNG). Natural Neighborhood Graph (NNG) which can be comprised by connecting each point  $i$  to its natural neighbors.

**Definition 10** (MNG). Maximum Neighborhood Graph (MNG) which can be comprised by connecting each point to its  $\max\{Rnb(i)\}$  nearest neighbors.

It is possible that different point have different number of natural neighbors in NNG. It has to be noted that all points have the same number of neighbors similar to  $k$ -nn graph but  $k$  is the value of  $\max\{Rnb(i)\}$  in MNG. And the value of  $\max\{Rnb(i)\}$ , named Natural Value, is adaptively obtained by NaN-searching algorithm. Based on MNG, we propose a novel non-parameter outlier detection algorithm detailed introduced in the following section.

### 3.3. NOF related definition and algorithm

No matter density-based and distance-based approaches have their particular weakness, such as the low density patterns problem and the local density problem analyzed in paper [19]. In fact, many density-based approaches can solve the problem of the above with appropriate parameters. So in this paper committed to find the appropriate parameter  $k$  using Natural Neighborhood search method described in Section 3. In this section, we will introduce our new measure and related properties.

**Definition 11** (Natural Outlier). If the  $Rnb(i)$  remains zero after the end of Algorithm 1. Then, the point that remained  $Rnb(i) = 0$  is called Natural Outlier.

Through experiment we find that the number of points that  $Rnb(i) = 0$  do not decreased as Algorithm 1 repeats a certain stage. The point that remained  $Rnb(i) = 0$  is Natural Outlier. In fact, the Natural Outlier of dataset must be the outlier of the dataset.

**Definition 12** (Natural Influence Space). The Natural Influence Space is defined as:

$$NIS(p) = NN_k(p) \cup RNN_k(p), k = \max(Rnb(i)) \quad (8)$$

There must note that  $k$  is not a parameter that set artificially, but the Natural Value. Therefore, there is non-paramete when obtain the  $NIS(p)$ .

**Definition 13** (Natural Outlier Factor). The Natural Outlier Factor (NOF) is defined as:

$$NOF(p) = \frac{\sum_{q \in NIS(p)} lrd_k(q)}{|NIS(p)| * lrd_k(p)}, k = \max(Rnb(i)) \quad (9)$$

Based on the algorithm1, we propose a outlier detecting algorithm based on Natural Neighbor. The NOF algorithm will make use of the

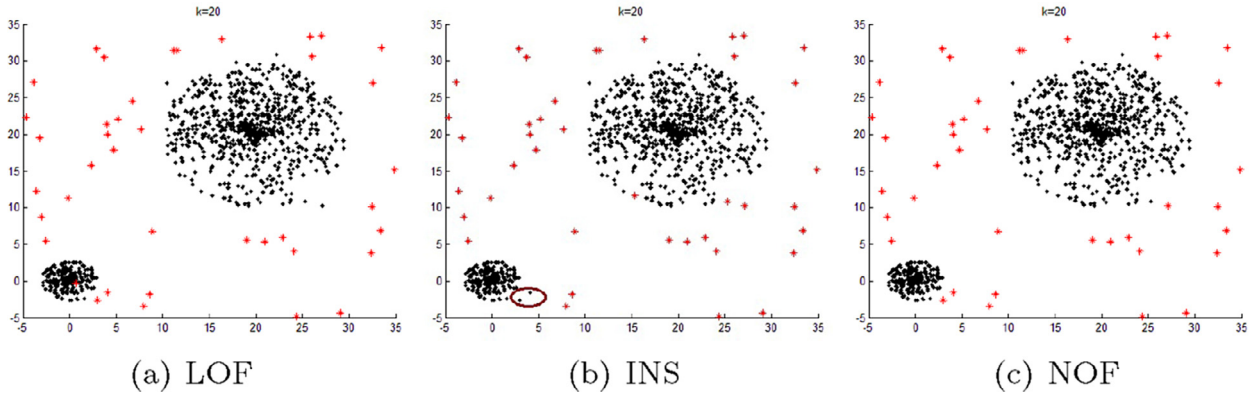


Fig. 2. Detection results of data1 by LOF, INS and NOF. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

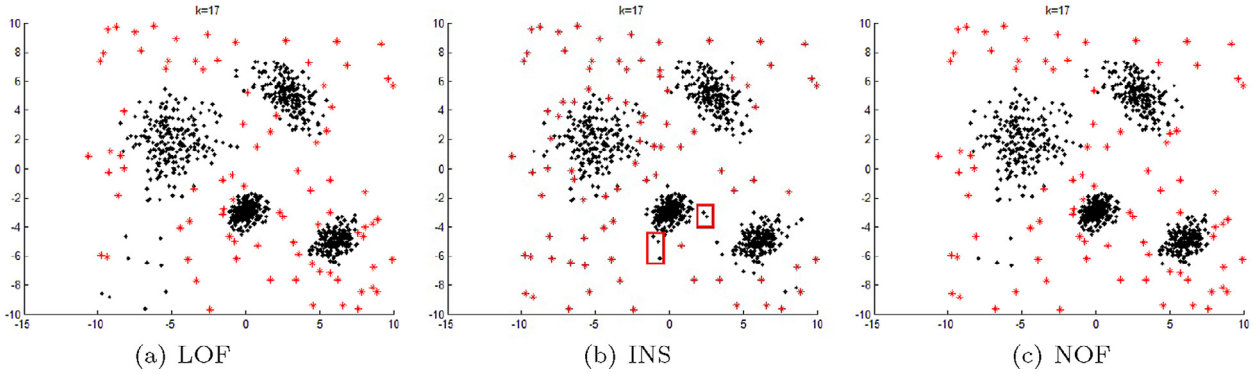


Fig. 3. Detection results of data2 by LOF, INS and NOF. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

MNG, explained in Definition 11, to detect outlier. The NOF algorithm as Algorithm 2.

**Algorithm 2** NOF-NAN(DataSet,  $n$ ) //  $n$  is the number of output outlier.

• **Output:** top- $n$  NOF of DataSet

- (1) Using the NAN-searching algorithm to construct the MNG, and obtain the  $k = \max(\text{Rnb}(i))$ .
- (2) Compute the  $k$ -distance for each point of DataSet.
- (3) compute the local reachability density for each point of DataSet.
- (4) compute the NOF for each point of DataSet.
- (5) Sort the NOF( $x$ ) and output the top- $n$  point.

As most recent studies of outlier detection have done, the proposed method ranks outliers by scoring the degree of outlierness (NOF). Therefore,  $n$  objects with the highest Natural Outlier Factor are declared as outliers at the final step of the NOF algorithm. Unlike existing outlier detect algorithm that the value of parameter  $k$  must be pre-determined, the value of parameter  $k$  of the proposed method is obtained adaptively using Algorithm 1, which is one of its strengths. The main cost of time is that obtaining the neighbors of each points in Algorithm 1. Hence, the same as Algorithm 1, the time complexity of NOF is  $O(N \log N)$ .

#### 4. Performance evaluation

##### 4.1. Metrics for measurement

For performance evaluation of the algorithms, we use two metrics, namely Accuracy and Rank-Power [23], to evaluate the detection results. Let  $N$  be the number of the true outliers that dataset  $D$  contains. And let  $M$  be the number of the true outliers that detected by an algorithm. In experiment, we detect out  $N$  most suspicious instances.

Then the Accuracy (Acc) is given by:

$$\text{Acc} = \frac{M}{N} \quad (10)$$

If using a given detection method, true outliers occupy top positions with respect to the non-outliers among  $N$  suspicious instances, then the Rank-Power (RP) of the method is said to be high [24]. If  $n$  denotes the number of outliers found within top  $N$  instances and  $R_i$  denote the rank of the  $i$ th true outlier, then the RP is given by:

$$\text{RP} = \frac{n(n+1)}{2 \sum_{i=1}^n R_i} \quad (11)$$

RP can obtain the maximum value 1 when all  $n$  true outliers are located in top  $n$  positions. And larger values of these metrics imply better performance.

##### 4.2. Synthetic examples

In order to show the effectiveness of the proposed method and the Natural Value obtained by the Natural Neighbor search method in density-based approaches, comparative study based on five synthetic datasets was conducted. These five synthetic datasets, taken from [19], contain various cluster patterns, different degrees of cluster density, and different cluster sizes in order to evaluate the NOF method in a harsh testing environment. We compared the performance of our method with those two existing approaches (the LOF and the INS). In agreement with the previous results, the objects with the highest outlierness scores by each method are colored red in follow experiments.

Figs. 2–6 show the outlier detection results of the three methods. There must be noted that the value of parameter  $K$  is Natural Value (NV), marked in red in Table 1, that obtained by Algorithm 1, not artificially set, in this experiment. Fig. 2 shows the results of the three



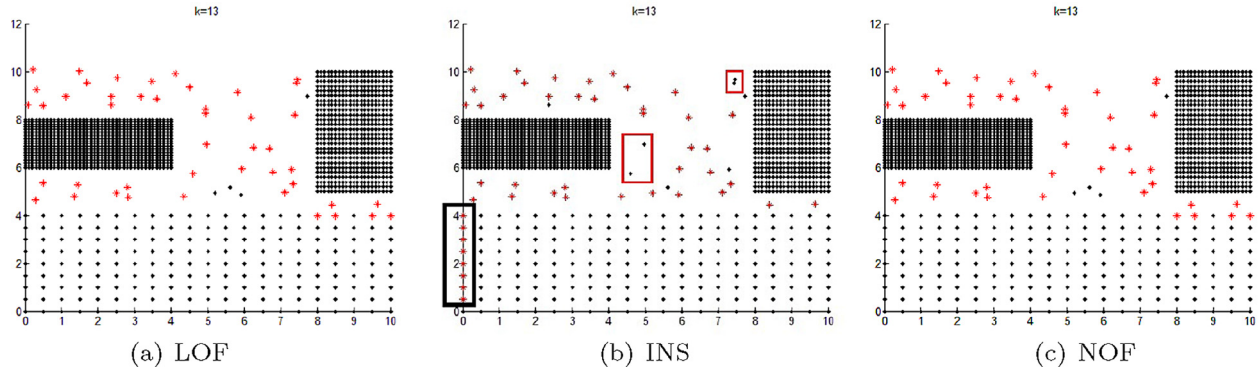


Fig. 4. Detection results of data3 by LOF, INS and NOF. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

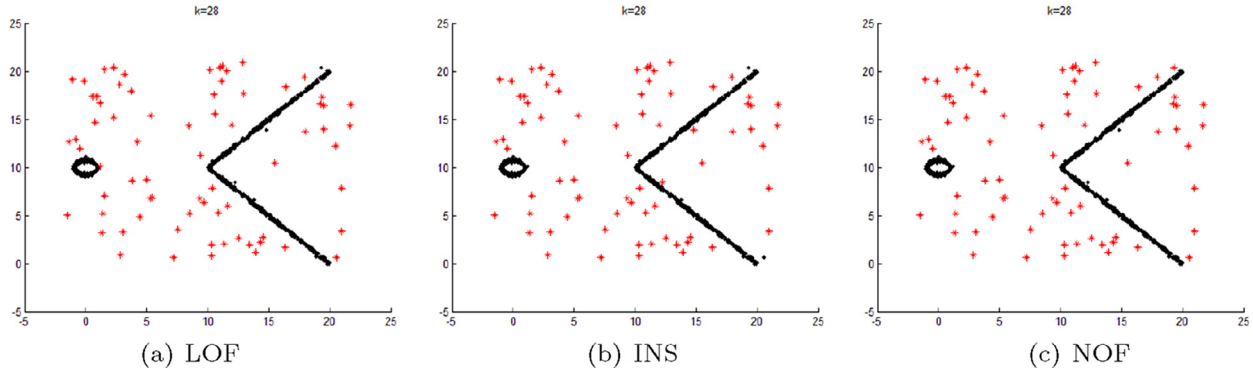


Fig. 5. Detection results of data4 by LOF, INS and NOF.

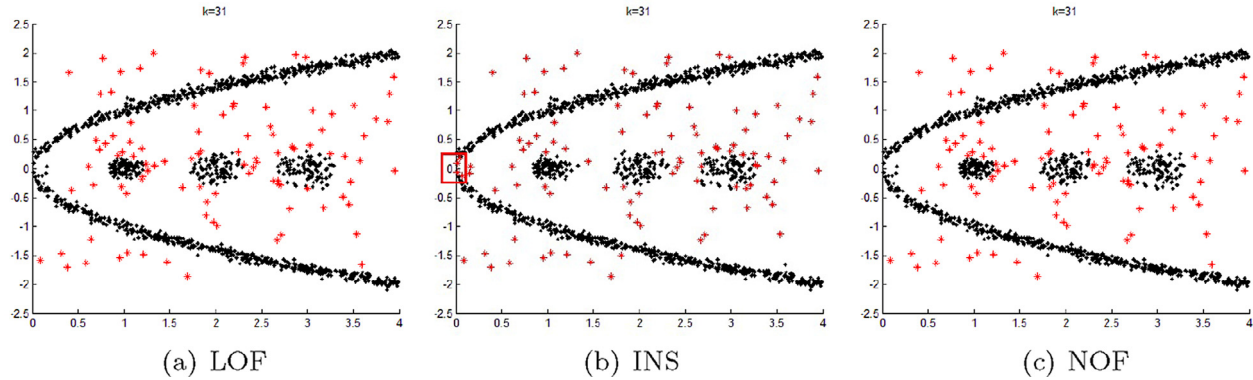


Fig. 6. Detection results of data5 by LOF, INS and NOF. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

approaches on data1. This dataset consists of two clusters, one is dense, another is sparse. A total of 1050 objects are included in data1, of which 41 objects are outliers. The results of LOF and NOF are similar, and outperforms the INS on local outlier detection on data1. INS is failed to detect the local outliers that located in the red circle. However, from the results of Fig. 7a, we can see that the accuracies of LOF and INS are very close to the highest that they can get, and NOF is the best, when the value of parameter  $k$  is assigned Natural Value ( $k = 20$ ).

Data2 consists of 1000 objects, four formal clusters and 85 outliers, as shown in Fig. 3. From the results, shown in Fig. 3, we can see that the global outliers detection effect of NOF is poor than INS, but better than LOF. However, the effect of LOF and NOF is better than INS in local outliers detection (some local outlier was marked in the red square). So the accuracy of NOF is highest among the three methods when the value of parameter  $k$  is assigned Natural Value ( $k = 17$ ). But, as shown in Fig. 7b, the accuracy of LOF and INS is close to the

highest too, when the value of parameter  $k$  is assigned Natural Value ( $k = 17$ ).

Data3 shown in Fig. 4 involves a local density problem. A total of 1641 objects are included in this dataset, and 45 objects are supposed to be outliers. The result of INS is the worst of the three methods. INS erroneously detect the normal points that are located in black square as outliers, and some real outliers that are located in red square cannot be detected. Although some real outliers cannot be detected by LOF and NOF too, the effect of LOF and NOF is obviously better than INS. As shown in Fig. 7c, we can see that the accuracy of NOF and LOF is higher than INS. Yet despite all, the accuracy of LOF and INS is close to the highest too, when the value of parameter  $k$  is assigned Natural Value ( $k = 13$ ).

Data4 shown in Fig. 5 involves a low density patterns problem. A total of 880 objects are included in this dataset and 72 objects are outliers. From the results, shown in Fig. 5, we can see that the results for the three methods are almost identical. This further proved the

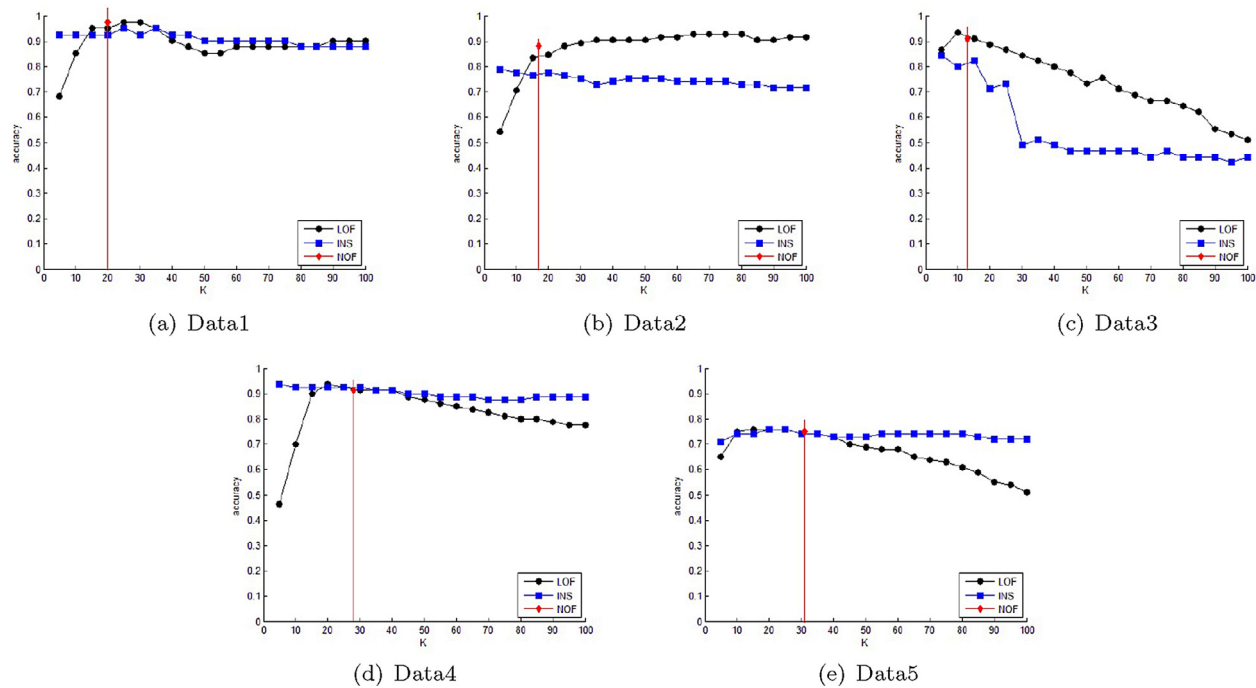


Fig. 7. Detection accuracies of the three detection methods over a range of  $k$  values.

**Table 1**  
Performance of the synthetic dataset.

Database	N	K	LOF		INS		NOF		
			Acc	RP	Acc	RP	NV	Acc	RP
Data1	41	10	0.85	0.97	0.93	0.99	20		
			0.95	1	0.93	0.99			
		50	0.85	0.98	0.90	0.98			
Data2	85	10	0.70	0.90	0.77	0.91	17	0.88	0.96
			0.82	0.96	0.75	0.93			
		50	0.90		0.75	0.92			
Data3	45	7	0.97	0.98	0.80	0.97	13		
				0.99	0.82	0.95			
		25	0.87	0.98	0.73	0.96			
Data4	80	10	0.70	0.95	0.92	0.99	28		
				0.99	0.92	0.99			
		50	0.87	0.99	0.90	0.99			
Data5	100	10		0.95	0.74	0.93	31		
			0.74	0.96	0.74	0.92			
		50	0.69	0.96	0.73	0.92			
		100	0.51	0.70	0.72	0.91			

effectiveness of Natural Value. As shown in Fig. 7d, the accuracy of these three methods are almost identical too, when the value of parameter  $k$  is assigned Natural Value ( $k = 28$ ). This result indicate that Natural Value not only is available to NOF, but also is available to LOF and NOF. It is possible that Natural Valuer is available to other outlier detection algorithms.

As shown in Fig. 6, Data5 have low density patterns and different degrees of clusters with manifolds. Data 5 consists of 1400 objects, of which 100 objects are outliers. The result of INS is the worst one. INS erroneously detect the points, located in the red square, that belong to the cluster with manifold as outliers. The results of LOF and NOF is almost identical. Furthermore, from the result of Fig. 7e, we can see that the accuracy of LOF and INS is very close to highest, and the ac-

curacy of NOF is highest among these three methods when the value of parameter  $k$  is assigned Natural Value ( $k = 31$ ).

In order to strengthen the effectiveness of Natural Value. We also make an experiment on above five datasets, and plot the situation that accuracy changed with the value of  $k$ , shown in Fig. 7 and Table 1. The results show that NOF has a good effect, shown in Figs. 2–6, on all of the five datasets. As shown in Table 1, the Acc and RP of NOF is the highest, marked in red, for data1 and data3–5. Though the Acc and RP of NOF is not the highest for Data2, NOF does not need artificial parameter and the Acc and RP of NOF are close to the best result, marked in red, that is obtained by LOF when  $k = 50, 100$  that hard to know in practical application. Moreover, no matter LOF or INS, use Natural Value as the value of parameter  $k$ , the experimental effect is the best or close to the best of all the results. In other words, these results prove that Natural Value not only appropriate to NOF, but also can be applied to LOF and INS.

4.3. Real data examples

We also applied the proposed method to two real-world-datasets. The wine dataset contains 113 objects that was group into 3 clusters. The iris dataset contains 106 objects that were grouped into 3 clusters too. Both datasets are obtained from the University of California, Irvine (UCI) machine learning repository. In this paper, we select two clusters among the three clusters as normal objects and select 6 objects from the remaining cluster as outliers. Since both of the datasets are multi-dimensional, we employed principal component analysis for the purpose of visualization. Each dataset was plotted on the space of two principal components. After computing the outlier-ness scores using the three methods, we find out 6 suspicious objects and compute the ACC and RP. The results are shown in Table 2. From the results of Table 2, we can see that the Natural Value is 20 and 15, and the Acc = 1 and RP = 1 of NOF is the highest for Wine and Iris datasets. For Wine dataset, LOF can get the best result too, Acc = 1 and RP = 1, when  $k = 50$ . But it must use more time than  $k = 20$  (Natural Value). For Iris dataset, when the value of  $k$  is small the effect of LOF is bad, but the effect of INS is bad when the value of  $k$  is large.

**Table 2**  
Performance of the synthetic dataset.

Database	N	K	LOF		INS		NOF		
			Acc	RP	Acc	RP	NV	Acc	RP
Wine	6	10	0.67	0.56	0.67		20		
		50			0.50	0.85			
Iris	6	7	0.33	0.60			15		
		50			0.67	0.67			

No matter LOF or INS get the best results that  $\text{Acc} = 1$  and  $\text{RP} = 1$  when the value of parameter( $k$ ) is Natural Value.

## 5. Conclusions and further study

In this study, we propose a new density-based algorithm for outlier detection. The proposed method combine the concept of the Natural Neighbor and previous density-based methods. As the most of the previous outlier detection methods, an object with a high outlierness scores is a promising candidate for an outlier. But unlike the most of the previous outlier detection approaches, our method is non-parametric. We use Algorithm 1 to adaptively obtain the value of  $k$ , named Natural Value. Experimental results clearly indicate that the effectiveness of the proposed method for both synthetic as well as the real datasets (Wine and Iris). Moreover, In addition to the NOF, Natural Value can be used in other outlier detection algorithm such as LOF and INS, and get nice results. Through the above analysis, we confirmed that the proposed method can accurately detects outliers from most patterns, and the proposed approach is non-parametric, and the Natural Value is applicable to other outlier detection methods. In order to further prove the effectiveness of Natural value, we will apply the Natural Value to more outliers detection and clustering algorithms in further studies.

## Acknowledgment

This research was supported by the National Natural Science Foundation of China (Nos. 61272194 and 61073058).

## References

- [1] W. Jin, A.K. Tung, J. Han, Mining top- $n$  local outliers in large databases, *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: Concepts and Techniques*, Elsevier, 2011.
- [3] T. Pang-Ning, M. Steinbach, V. Kumar, *Introduction to Data Mining*, Library of Congress, 2006.
- [4] E.M. Knox, R.T. Ng, Algorithms for mining distance-based outliers in large datasets, *Proceedings of the International Conference on Very Large Data Bases*, Citeseer, 1998.
- [5] E.M. Knorr, R.T. Ng, A unified notion of outliers: properties and computation, in: *Proceedings of International Conference on Knowledge Discovery and Data Mining*, KDD, 1997.
- [6] 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.
- [7] V. Barnett, T. Lewis, *Outliers in Statistical Data*, vol. 3, Wiley, New York, 1994.
- [8] D.M. Hawkins, *Identification of Outliers*, vol. 11, Springer, 1980.
- [9] S. Shekhar, S. Chawla, *A Tour of Spatial Databases*, Prentice Hall, Upper Saddle River, New Jersey, 2002.
- [10] I. Ruts, P.J. Rousseeuw, Computing depth contours of bivariate point clouds, *Comput. Stat. Data Anal.* 23 (1) (1996) 153–168.
- [11] T. Johnson, I. Kwok, R.T. Ng, Fast computation of 2-dimensional depth contours, *Proceedings of International Conference on Knowledge Discovery and Data Mining*, KDD, Citeseer, 1998.
- [12] M. Ester, A density-based algorithm for discovering clusters in large spatial databases with noise, in: *Proceedings of International Conference on Knowledge Discovery and Data Mining*, KDD, 1996.
- [13] T.N. Raymond, J. Han, Efficient and effective clustering methods for spatial data mining, in: *Proceedings of the 20th International Conference on Very Large Data Bases*, 1994.
- [14] G. Karypis, E.-H. Han, V. Kumar, Chameleon: hierarchical clustering using dynamic modeling, *Computer* 32 (8) (1999) 68–75.
- [15] T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: an efficient data clustering method for very large databases, *ACM SIGMOD Record*, ACM, 1996.
- [16] S. Guha, R. Rastogi, K. Shim, CURE: an efficient clustering algorithm for large databases, *ACM SIGMOD Record*, ACM, 1998.
- [17] M.M. Breunig, LOF: identifying density-based local outliers, *ACM Sigmod Record*, ACM, 2000.
- [18] W. Jin, Ranking outliers using symmetric neighborhood relationship, *Advances in Knowledge Discovery and Data Mining*, Springer, 2006, pp. 577–593.
- [19] J. Ha, S. Seok, J.-S. Lee, Robust outlier detection using the instability factor, *Knowledge-Based Syst.* 63 (2014) 15–23.
- [20] J.L. Bentley, Multidimensional binary search trees used for associative searching, *Commun. ACM* 18 (9) (1975) 509–517.
- [21] X. Luo, Boosting the  $K$ -nearest-neighborhood based incremental collaborative filtering, *Knowledge-Based Syst.* 53 (2013) 90–99.
- [22] S.S. Stevens, *Mathematics, Measurement, and Psychophysics*, American Psychological Association, 1951.
- [23] C. Lijun, A data stream outlier detection algorithm based on reverse  $k$  nearest neighbors, *Proceedings of the 2010 International Symposium on Computational Intelligence and Design (ISCID)*, IEEE, 2010.
- [24] J. Tang, Enhancing effectiveness of outlier detections for low density patterns, *Advances in Knowledge Discovery and Data Mining*, Springer, 2002, pp. 535–548.