Ann Oper Res (2009) 168: 151–168
DOI 10.1007/s10479-008-0371-9

Cluster-based outlier detection

Lian Duan · Lida Xu · Ying Liu · Jun Lee

Published online: 12 June 2008

Abstract Outlier detection has important applications in the field of data mining, such as fraud detection, customer behavior analysis, and intrusion detection. Outlier detection is the process of detecting the data objects which are grossly different from or inconsistent with the remaining set of data. Outliers are traditionally considered as single points; however, there is a key observation that many abnormal events have both temporal and spatial locality, which might form small clusters that also need to be deemed as outliers. In other words, not only a single point but also a small cluster can probably be an outlier. In this paper, we present a new definition for outliers: cluster-based outlier, which is meaningful and provides importance to the local data behavior, and how to detect outliers by the clustering algorithm LDBSCAN (Duan et al. in Inf. Syst. 32(7):978–986, 2007) which is capable of finding clusters and assigning LOF (Breunig et al. in Proceedings of the 2000 ACM SIG MOD International Conference on Manegement of Data, ACM Press, pp. 93–104, 2000) to single points.

Keywords Outlier detection · Cluster-based outlier · LDBSCAN · Local outlier factor

L. Duan (✉)
Management Sciences Department, University of Iowa, Iowa City, IA, USA
e-mail: lian-duan@uiowa.edu

L. Xu
College of Economics and Management, Beijing Jiaotong University, Beijing 100044, China
e-mail: odubus@gmail.com

L. Xu
Department of Information Technology & Decision Science, Old Dominion University, Norfolk, VA 23529, USA

Y. Liu
Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing, China
e-mail: yingliu@gucas.ac.cn

J. Lee
China Science and Technology Network, Chinese Academy of Sciences, Beijing, China
e-mail: jlee@cstnet.cn

1 Introduction

For many KDD applications, such as detecting criminal activities in e-business environment, finding the rare instances or the outliers can be more interesting than finding the common patterns. Finding such exceptions and outliers has received as much attention in the KDD community as some other topics have. An outlier in a dataset is defined informally as an observation that is considerably different from the remainders as if it is generated by a different mechanism. Many current studies have focused on outlier detection; however, almost all of them only regard the outlier as a single point that deviates from a certain cluster. This paper presents a new definition for outliers, namely cluster-based outlier, which is intuitive and meaningful. Our approach is based on a key observation that many abnormal events have both temporal and spatial locality, so a small cluster might also be an outlier. Given a patient who has a fever, his body temperature will increase and be different from the normal value. However, when the body temperature increases to a certain point, the temperature will fluctuate around this point instead of drastically deviating from this point. If we sample the patient's body temperature every hour, the sampled data during the fever will form a small cluster deviating from the big cluster formed by the sampled data when he is healthy. This small cluster is a cluster-based outlier and indicates the anomalous period when he has the fever. In addition, many literatures on outlier detection regard being an outlier as a binary property; however, for many scenarios, it is more meaningful to assign to each outlier a degree of being an outlier and the deviation degree depends on how isolated the outlier is with respect to the surrounding neighborhood. Therefore, an algorithm is desired which can detect both single point outliers and cluster-based outliers, and can assign each outlier a degree of being an outlier.

In this paper, a new approach to detect both single point outliers and cluster-based outliers is proposed. The paper is organized as follows. Related work on outlier detection is briefly introduced in Sect. 2. In Sect. 3, the different notions of an outlier are presented, and their advantages and disadvantages are carefully discussed. Section 4 briefly introduces the algorithms LDBSCAN which is used to detect outliers. In Sect. 5, how to detect the outliers based on the clustering result of LDBSCAN, some theorems about cluster-based outliers, and the parameter which indicates the degree of deviation are presented. Then, a thorough experimental evaluation of the effectiveness and efficiency of our approach is carried out in Sect. 6. Finally Sect. 7 concludes the paper with a summary and some directions for future research.

2 Related work

Many data mining algorithms in the literature find outliers as a side-product of clustering algorithms (Ester et al. 1996; Zhang et al. 1996; Wang et al. 1997; Agrawal et al. 1998; Hinneburg and Keim 1998; Guha et al. 1998; Sheikholeslami et al. 1998; Ankerst et al. 1999; Li and Xu 2001; Ng and Han 2002; Li et al. 2003; Qiu et al. 2003; Zhang et al. 2003; Xu 2006; Carvalho and Costa 2007; Hsu and Wallace 2007; Luo et al. 2007; Shi et al. 2007; Xu et al. 2008). These techniques define outliers as points which do not lie in clusters. Thus, the techniques implicitly define the outlier as the background noise in which the clusters are embedded. The noise is typically tolerated or ignored when these algorithms produce the clustering result. Even if the outliers are not ignored, the notions of an outlier are essentially binary, which cannot enjoy many desirable properties of assigning to each outlier a degree of being an outlier (Breunig et al. 2000).

The statistics community has studied the concept of outliers quite extensively (Barnett and Lewis 1994). The first category is distribution-based, where a standard distribution is used to fit the data. Outliers are defined based on the probability distribution. A key drawback of this category is that most of the distributions used are univariate while some tests are multivariate. In addition, for many KDD applications the underlying distribution is unknown, so fitting the data with standard distributions is costly and may not produce satisfactory results. Another type of outlier studies in statistics is depth-based. Each data object is represented in a k-dimension space, and is assigned a depth (Johnson et al. 1998; Preparata and Shamos 1988; Tukey 1977). However, depth-based approaches rely on the computation of k–d convex hulls, so they become inefficient when $k \geq 4$.

The notion of distance-based outliers is proposed by Knorr and Ng (1998, 1999). Their notion generalizes many notions from the distribution-based approaches, and enjoys better computational complexity than the depth-based approaches for larger values of k. In Ramaswamy et al. (2000) the notion of the distance-based outlier is extended by using the distance of the k-nearest neighbor to rank the outlier, but its notion of an outlier is still distance-based. Nevertheless, this measure is naturally susceptible to the curse of high dimensionality, especially when very different local-density clusters exist in different regions of data space.

A more effective technique (Breunig et al. 2000) finds outliers based on the density of local neighborhood. This technique has some advantages in accounting for local levels of skews and anomalies in data collection. It assigns to each object a degree of being an outlier, which captures the spirit of local outliers, rather than a binary value to indicate whether or not the object is an outlier. However, this approach only finds single point outliers while ignoring cluster-based outliers.

There are some preliminary ideas about cluster-based outliers (He et al. 2003; Jiang et al. 2001; Knorr and Ng 1999); however, these methods have two major problems. First, they try to evaluate each point in a small cluster instead of evaluating the small cluster as a whole. Second, the clustering algorithms they use are not suitable to find a small cluster. Distance-based outlier method also can find cluster-based outliers in some cases; however, the objects it finds are each isolated point and it doesn't know which points belong to the same cluster.

3 Definition of outlier

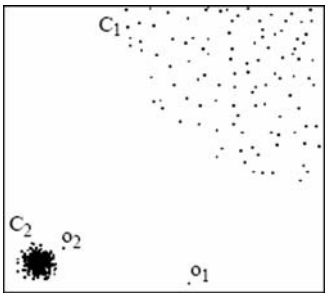
Some of existing work in outlier detection lies in the field of statistics. Intuitively, outlier can be defined as given by Hawkins (1980).

Definition 1 (Hawkins-Outlier) An outlier is an observation that deviates so much from other observations as to arouse suspicion that it is generated by a different mechanism.

Throughout this paper, we use o, p, q to denote objects in a dataset and use the notation $d(p, q)$ to denote the distance between object p and q. Also C is used for a set of objects and $d(p, C)$ denotes the minimum distance between p and object q in C, i.e. $d(p, C) = min\{d(p, q) \mid q \in C\}$.

Definition 2 ($DB(pct, dmin)$-Outlier) An object p in a dataset D is a $DB(pct, dmin)$-outlier if at least percentage pct of the objects in D lies greater than distance $dmin$ from p, i.e., the cardinality of the set $\{q \in D \mid d(p, q) <= dimin\}$ is less than or equal to (100-pct)% of the size of D.

Fig. 1 Dataset DS1



The above definition captures only certain kinds of outliers. Because the definition takes a global view of the dataset, these outliers can be viewed as "global" outliers. However, for many interesting real-world datasets which exhibit a more complex structure, there is another kind of outlier. There can be objects that are outlying relative to their local neighborhoods, particularly with respect to the densities of the neighborhoods.

Consider the example given in Fig. 1. This is a simple 2-dimensional dataset containing 502 objects. There are 400 objects in the first cluster C_1, 100 objects in the cluster C_2, and two additional objects o_1 and o_2. According to Hawkins' definition, both o_1 and o_2 should be called outliers, while objects in C_1 and C_2 should not be. However, within the framework of distance-based outlier, only o_1 is a reasonable $DB(pct, dmin)$-outlier. If for every object q in C_1, the distance between q and its nearest neighbor is greater than the distance between o_2 and C_2, there is no appropriate value of pct and dmin such that o_2 is a $DB(pct, dmin)$-outlier but the objects in C_1 are not. In order to solve this problem, a formal definition of density-based local outlier is developed, which avoids the shortcomings presented above. The key difference between density-based local outlier and $DB(pct, dmin)$-outlier is that being outlier is no longer a binary property. Instead, each object is assigned a local outlier factor, which indicates the degree of the outlying object.

Definition 3 (Density-based local outlier) Given the lowest acceptable bound of LOF (Breunig et al. 2000), an object p in a dataset D is a density-based local outlier if $LOF(p) > LOFLB$.

The core idea of LOF is comparing its local density with the local densities of its neighbors. The local density of an object p is the inverse of the average distance to its k-nearest neighbors. The concept of the local density is very intuitive. The farther away p is from its k-nearest neighbors, the sparser its local density is. The LOF of an object p is the ratio of the local density of p and those of p's k-nearest neighbors. It is easy to see that the lower p's local density is, and the higher the local densities of p's k-nearest neighbors are, the higher the LOF of p is. The following figure gives us an intuitive impression of LOF. Figure 2(a) shows a 2-dimensional dataset containing one low density Gaussian cluster of 200 objects and three large clusters of 500 objects each. Among these three, one is a dense Gaussian cluster and the other two are uniform cluster of different densities. Besides, it contains a few outliers. Figure 2(b) plots the LOF of all the objects as a third dimension. The objects in any cluster have their LOF close to 1. Slightly outside of the Gaussian clusters, there are several weak outliers whose LOFs are relatively low but larger than 1. The remaining objects have significantly larger LOF. From Fig. 2, it is clear that the LOF of each object depends on the density of the cluster relative to it and the distance between it and the cluster.

Fig. 2 LOFs for points

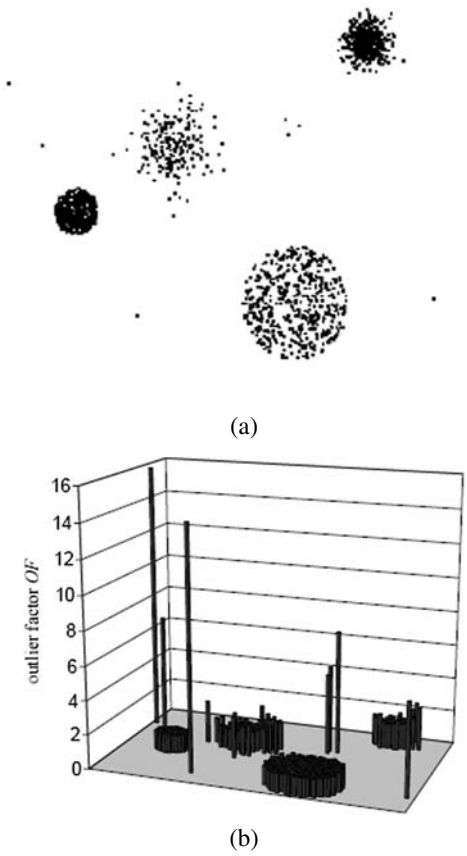
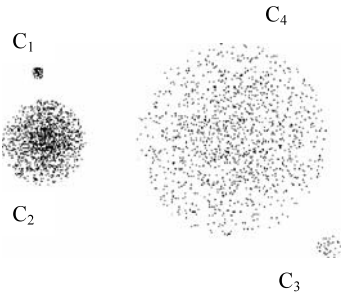


Fig. 3 Cluster-based outliers



Consider the example given in Fig. 3. The LOF of each object contained in cluster C_1 and C_3 is approximate to 1. According to definition of density-based local outlier, they are not outliers. However, it is more reasonable to consider C_1 and C_3 as outliers according to Hawkins' definition. In order to solve this problem, a formal definition of cluster-based outliers is developed. The density-based local outlier works well in finding single point outliers while ignoring the existence of cluster-based outliers. In order to find out cluster-based outliers as well as single point outliers, first we present the algorithm LDBSCAN

which is used to calculate the LOF of each object and discover all the clusters in a dataset. Then a cluster-based outlier factor, which is called CBOF, is assigned to each discovered cluster. Finally outliers in the dataset are located.

4 Brief introduction of LDBSCAN

The formal definitions and pseudo-code for the procedure of LDBSCAN are shortly introduced in the following. More details are provided in Duan et al. (2007).

Definition 1 (k-distance of an object p) For any positive integer k, the k-distance of object p, denoted as $k\text{-}distance(p)$, is defined as the distance $d(p,o)$ between p and an object $o \in D$ such that:

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

Definition 2 (k-distance neighborhood of an object p) Given the k-distance of p, the *k-distance neighborhood of p* contains every object whose distance from p is not greater than the k-distance, i.e. $N_{k\text{-}distance(p)}(p) = \{q \in D\backslash\{p\} | d(p,q) \leq k\text{-}distance(p)\}$. These objects q are called the k-nearest neighbors of p.

As no confusion arises, the notation can be simplified to use $N_k(p)$ as a shorthand for $N_{k\text{-}distance(p)}(p)$.

Definition 3 (Reachability distance of an object p w.r.t. object o) Let k be a natural number. The *reachability distance* of object p with respect to object o is defined as $reach\text{-}dist_k(p,o) = max\{k\text{-}distance(o), d(p,o)\}$.

Definition 4 (Local reachability density of an object p) The *local reachability density* of p is defined as

$$LRD_{MinPts}(p) = 1 \Big/ \left(\frac{\sum_{o \in N_{MinPts}(p)} reach\text{-}dist_{MinPts}(p,o)}{|N_{MinPts}(p)|} \right).$$

Intuitively, the *local reachability density* of an object p is the inverse of the average *reachability distance* based on the *MinPts*-nearest neighbors of p.

Definition 5 (Local outlier factor of an object p) The *local outlier factor* of p is defined as

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{LRD_{MinPts}(o)}{LRD_{MinPts}(p)}}{|N_{MinPts}(p)|}.$$

The *local outlier factor* of object p is the average of the ratio of the *local reachability density* of p and those of p's *MinPts*-nearest neighbors.

Definition 6 (Core point) A point p is a *core point* w.r.t. $LOFUB$ if $LOF(p) \leq LOFUB$.

If $LOF(p)$ is small enough, it means that point p is not an outlier and must belong to some clusters. Therefore it can be regarded as a core point.

Definition 7 (Directly local-density-reachable) A point p is *directly local-density-reachable* from a point q w.r.t. *pct* and *MinPts* if

(1) $p \in N_{MinPts}(q)$ and
(2) $LRD(q)/(1+pct) < LRD(p) < LRD(q) * (1+pct)$.

Definition 8 (Local-density-reachable) A point p is *local-density-reachable* from the point q w.r.t. *pct* and *MinPts* if there is a chain of points $p_1, p_2, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly local-density-reachable from p_i.

Definition 9 (Local-density-connected) A point p is *local-density-connected* to a point q from o w.r.t. *pct* and *MinPts* if there is a point o such that both p and q are local-density-reachable from o w.r.t. *pct* and *MinPts*.

Definition 10 (Cluster) Let D be a database of points, and point o is a selected core point of C, i.e. $o \in C$ and $LOF(o) \leq LOFUB$. A *cluster* C w.r.t. *LOFUB*, *pct* and *MinPts* is a non-empty subset of D satisfying the following conditions:

(1) $\vee p : p$ is local-density-reachable from o w.r.t. *pct* and *MinPts*, then $p \in C$. (Maximality)
(2) $\vee p, q \in C : p$ is local-density-connected q by o w.r.t. *LOFUB*, *pct* and *MinPts*. (Connectivity)

Definition 11 (Noise) Let $C_1, \ldots, C_k$ be the clusters of the database D w.r.t. parameters *LOFUB*, *pct* and *MinPts*. Then we define the *noise* as the set of points in the database D not belonging to any cluster C_i, i.e. noise $= \{p \in D \mid \vee i : p \notin C_i\}$.

Given the parameters *LOFUB*, *pct* and *MinPts*, clusters can be found with a two-step approach. First, an arbitrary point p from the database satisfying the core point condition $LOF(p) \leq LOFUB$ as a seed is chosen. Second, all points that are local-density-reachable from the seed obtaining the cluster which contains the seed are retrieved.

In the following, we present a basic version of LDBSCAN without details of data types and generation of additional information about clusters:

```
LDBSCAN (SetOfPoints, LOFUB, pct, MinPts)
  // SetOfPoints is UNCLASSIFIED
  InitSet (SetOfPoints); // calculate LRD and LOF of each point
  ClusterID := 0;
  FOR i FROM 1 TO SetOfPoints.size DO
    Point := SetOfPoints.get(i);
    IF Point.ClId = UNCLASSIFIED THEN
      IF LOF(Point) ≤ LOFUB THEN // core point
        ClusterID := ClusterID + 1;
        ExpandCluster(SetOfPoints, Point, ClusterID, pct, MinPts);
      ELSE // no core point
        SetOfPoint.changeClId(Point, NOISE);
      END IF
    END IF
  END FOR
END; //LDBSCAN
```

SetOfPoints is the set of the whole database. *LOFUB*, *pct* and *MinPts* are the carefully chosen parameters. The function SetOfPoints.get(i) returns the i-th element of SetOfPoints. Points which have been marked to be NOISE may be changed later if they are local-density-reachable from some core points of the database. The most important function used by LDBSCAN is ExpandCluster which is presented in the following:

```
ExpandCluster(SetOfPoints, Point, ClusterID, pct, MinPts)
  SetOfPoint.changeClId(Point, ClusterID);
  FOR i FROM 1 TO MinPts DO
    currentP := Point.Neighbor(i);
    IF currentP.ClId IN {UNCLASSIFIED, NOISE} and
    DirectReachability(currentP, Point)
THEN
      TempVector.add(currentP);
      SetOfPoint.changeClId(currentP, ClusterID);
    END IF
  END FOR
  WHILE TempVector <> Empty DO
    Point := TempVector.firstElement();
    TempVector.remove(Point);
    FOR i FROM 1 TO MinPts DO
      currentP := Point.Neighbor(i);
      IF currentP.ClId IN {UNCLASSIFIED, NOISE} and
DirectReachability(currentP, Point) THEN
        TempVector.add(currentP);
        SetOfPoint.changeClId(currentP, ClusterID);
      END IF
    END FOR
  END WHILE
END; //ExpandCluster
```

The function DirectReachability(currentP, Point) is presented in the following:

```
DirectReachability(currentP, Point) : Boolean
  IF LRD(currentP) > LRD(Point)/(1 + pct) and
  LRD(currentP) < LRD(Point) * (1 + pct) THEN
    RETURN True;
  ELSE
    RETURN False;
END; //DirectReachability
```

The LDBSCAN algorithm randomly selects one core point which has not been clustered, and then retrieves all points that are local-density-reachable from the chosen core point to form a cluster. It won't stop until there is no unclustered core point.

5 Cluster-based outliers

In this section, we give the definition of cluster-based outliers and conduct a detailed analysis on the properties of cluster-based outliers. The goal is to show how to discover cluster-based

outliers and how the definition of the cluster-based outlier factor (CBOF) captures the spirit of cluster-based outliers. The higher the CBOF is, the more abnormal the cluster-based outliers are.

5.1 Definition of cluster-based outliers

Intuitively, most data points in the data set should not be outliers; therefore, only the clusters that hold a small portion of data points are candidates for cluster-based outliers. Considering the different and complicated situations, it is impossible to provide a definite number as the upper bound of the number of the objects contained in a cluster-based outlier (UBCBO). Here, only a guideline is provided to find the reasonable upper bound.

Definition 12 (Upper bound of the cluster-based outlier) Let $C_1, \ldots, C_k$ be the clusters of the database D discovered by LDBSCAN in the sequence that $|C_1| \geq |C_2| \geq \cdots \geq |C_k|$. Given parameters α, the number of the objects in the cluster C_i is the UBCBO if $(|C_1| + |C_2| + \cdots + |C_{i-1}|) \geq |D| * \alpha$ and $(|C_1| + |C_2| + \cdots + |C_{i-2}|) < |D| * \alpha$.

Definition 12 gives quantitative measure to UBCBO. Consider that most data points in the dataset are not outliers; therefore, clusters that hold a large portion of data points should not be considered as outliers. For example, if α is set to 90%, we intend to regard clusters which contain 90% of data points as normal clusters.

Definition 13 (Cluster-based outlier) Let $C_1, \ldots, C_k$ 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 UBCBO.

Note that this guideline is not always appropriate. For example, in some cases the abnormal cluster deviated from a large cluster might contain more points than a certain small normal cluster. In fact, due to spatial and temporal locality, it would be more proper to choose the clusters which have small spatial or temporal span as cluster-based outliers than the clusters which contain few objects. The notion of cluster-based outliers depends on situations.

5.2 The lower bound of the number of the objects contained in a cluster

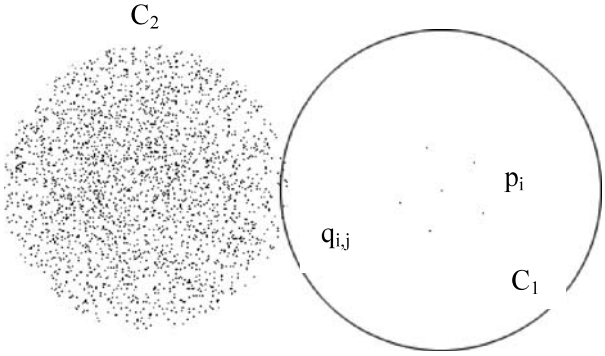
Comparing with single point outliers, cluster-based outliers are more interesting. Many single point outliers are related to occasional trivial events, while cluster-based outliers concern some important lasting abnormal events. Generally speaking, it is reckless to form a cluster with only 2 or 3 objects, so the lower bound of the number of the objects contained in a cluster generated by LDBSCAN will be discussed in the following.

Definition 14 (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

$$dist(C_1, C_2) = min\{dist(p, q) \mid p \in C_1, q \in C_2\}.$$

Theorem 1 *Let C_1 be the smallest cluster discovered by LDBSCAN w.r.t. appropriate parameters LOFUB, pct and MinPts, and C_2 which is large enough be the closest normal cluster to C_1. Let $LRD(C_1)$ denote the minimum LRD of all the objects in C_1, i.e.,*

Fig. 4 2-d Datase DS2



$LRD(C_1) = min\{LRD(p) \mid p \in C_1\}$. Similarly, let $LRD(C_2)$ denote the minimum LRD of all the objects in C_2.

Then for LBC, the lower bound of the number of the objects contained in a cluster, such that:

$$LBC = \left\lceil \frac{(MinPts+1)LRD(q) - (LOFUB * MinPts + 1)LRD(p)}{LRD(q) - LRD(p)} \right\rceil + 1.$$

Proof (Sketch): Let p_i denote the i-th object in C_1 and $q_{i,j}$ be the j-th close object to p_i in C_2. And let k be the number of the objects in C_1. To simplify our proof, we only consider the situation that each point only has k-nearest neighbors and the density within a cluster fluctuates slightly.

(a) If $k \geq MinPts + 1$, according to the definition of LOF, the LOF of any object in C_1 is approximately equal to 1. That is, $LOF(p_i) < LOFUB$ and each object in C_1 is a core point. In addition, each object in C_1 has the similar LRD to its neighbors which belong to the same cluster with it. According to the definition of the cluster, the cluster C_1 would be discovered by LDBSCAN. Thus, LBC is no more than $MinPts + 1$.

(b) In the following, the situation when $k \leq MinPts$ is discussed. Since $k \leq MinPts$, the $MinPts$-distance neighbors of p_i contain the $k-1$ rest objects in C_1 and the other $MinPts - k + 1$ neighbors in C_2. Obviously, the MinPts-distance of each fixed object p_j in C_1 is greater than the distance between any object p_i in C_1 and p_j, so $reach\text{-}dist(p_i, p_j) = MinPts\text{-}distance(p_j)$. Furthermore, the $MinPts\text{-}distance(q_{i,j}) \ll dist(C_1, C_2) \leq d(p_i, q_{i,j})$.

$$\Rightarrow \quad LRD_{MinPts}(p_i) = MinPts \Big/ \left(\sum_{a=1}^{k} MinPts\text{-}dist(p_a) - MinPts\text{-}dist(p_i) + \sum_{a=1}^{MinPts-k+1} d(p_i, q_{i,a}) \right)$$

$$LRD_{MinPts}(q_i) = MinPts \Big/ \sum_{o \in N_{MinPts}(q_i)} reach\text{-}dist_{MinPts}(q_i, o) \quad \text{and}$$

$$\forall p_i \in C_1$$

Let $MinPts\text{-}dist(p) = min\{MinPts\text{-}dist(p_i) \mid p_i \in C_1\}$, and then $MinPts\text{-}dist(p_i) = MinPts\text{-}dist(p) + \varepsilon_i$. Similarly, let $d(p, q) = min\{d(p_i, q_{i,j}) \mid p_i \in C_1, q_{i,j} \in C_2$ and $q_{i,j}$

is the *MinPts*-neighbor of $p_i\}$ and $d(p_i, q_{i,j}) = d(p,q) + \varepsilon_{i,j}$. Because we assume that the density within a cluster fluctuates slightly, $\mathit{MinPts\text{-}dist}(p) \gg \varepsilon_i$ and $d(p,q) \gg \varepsilon_{i,j}$.

Compare the LRD of object p_i with that of its neighbor p_j in C_1.

$$
\begin{aligned}
&\frac{\mathit{LRD}_{\mathit{MinPts}}(p_i)}{\mathit{LRD}_{\mathit{MinPts}}(p_j)} \\
&= \frac{\sum_{a=1}^{k} \mathit{MinPts\text{-}dist}(p_a) - \mathit{MinPts\text{-}dist}(p_j) + \sum_{a=1}^{\mathit{MinPts}-k+1} d(p_j, q_{j,a})}{\sum_{a=1}^{k} \mathit{MinPts\text{-}dist}(p_a) - \mathit{MinPts\text{-}dist}(p_i) + \sum_{a=1}^{\mathit{MinPts}-k+1} d(p_i, q_{i,a})} \\
&= \frac{\sum_{a=1}^{k} \mathit{MinPts\text{-}dist}(p_a) - \mathit{MinPts\text{-}dist}(p) - \varepsilon_j + \sum_{a=1}^{\mathit{MinPts}-k+1} (d(p,q) + \varepsilon_{j,a})}{\sum_{a=1}^{k} \mathit{MinPts\text{-}dist}(p_a) - \mathit{MinPts\text{-}dist}(p) - \varepsilon_i + \sum_{a=1}^{\mathit{MinPts}-k+1} (d(p,q) + \varepsilon_{i,a})} \\
&= \frac{\sum_{a=1}^{k} \mathit{MinPts\text{-}dist}(p_a) - \mathit{MinPts\text{-}dist}(p) + (\mathit{MinPts} - k + 1) * d(p,q) + \sum_{a=1}^{\mathit{MinPts}-k+1} \varepsilon_{j,a} - \varepsilon_j}{\sum_{a=1}^{k} \mathit{MinPts\text{-}dist}(p_a) - \mathit{MinPts\text{-}dist}(p) + (\mathit{MinPts} - k + 1) * d(p,q) + \sum_{a=1}^{\mathit{MinPts}-k+1} \varepsilon_{i,a} - \varepsilon_i} \\
&\approx 1
\end{aligned}
$$

Thus, the objects in C_1 have the similar LRD.

Now consider the ratio of the LRD of the object p_i to that of its neighbor q_j in C_2. Let reach-dist-max be the maximum reachability distance of the object q_j which is the object in C_2.

$$
\begin{aligned}
&\because \mathit{MinPts\text{-}dist}(p_i) > \mathit{dist}(C_1, C_2) \quad \text{and} \quad d(p_i, q_{i,j}) > \mathit{dist}(C_1, C_2) \\
&\therefore \frac{\mathit{LRD}_{\mathit{MinPts}}(q_j)}{\mathit{LRD}_{\mathit{MinPts}}(p_i)} = \frac{\sum_{a=1}^{k} \mathit{MinPts\text{-}dist}(p_a) - \mathit{MinPts\text{-}dist}(p_i) + \sum_{a=1}^{\mathit{MinPts}-k+1} d(p_i, q_{i,a})}{\sum_{o \in N_{\mathit{MinPts}}(q_i)} \mathit{reach\text{-}dist}_{\mathit{MinPts}}(q_i, o)} \\
&\quad > \frac{\mathit{MinPts} * \mathit{dist}(C_1, C_2)}{\mathit{MinPts} * \mathit{reach\text{-}dist\text{-}max}} = \frac{\mathit{dist}(C_1, C_2)}{\mathit{reach\text{-}dist\text{-}max}} \\
&\therefore \mathit{dist}(C_1, C_2) \gg \mathit{reach\text{-}dist\text{-}max}
\end{aligned}
$$

and the appropriate $pct < 1$ (Duan et al. 2007).

$\therefore \frac{\mathit{LRD}(q_j)}{\mathit{LRD}(p_i)} \gg 2 > 1 + pct$. That is, objects in C_2 will not be assigned to cluster C_1.

Then, if objects in C_1 form a cluster which can be discovered by LDBSCAN, the inequality, $\mathit{Min}(\mathit{LOF}_{\mathit{MinPts}}(p_i)) < \mathit{LOFUB}$, must be satisfied.

$$
\begin{aligned}
\Rightarrow \quad & \mathit{Min}(\mathit{LOF}_{\mathit{MinPts}}(p_i)) \\
& = \mathit{Min}\left(\frac{\sum_{a=1}^{k} \mathit{LRD}_{\mathit{MinPts}}(p_a) - \mathit{LRD}_{\mathit{MinPts}}(p_i) + \sum_{a=1}^{\mathit{MinPts}-k+1} \mathit{LRD}(q_{i,a})}{\mathit{MinPts} * \mathit{LRD}_{\mathit{MinPts}}(p_i)} \right) \\
& \geq \frac{(k-1)\mathit{LRD}(p) + (\mathit{MinPts} - k + 1)\mathit{LRD}(q)}{\mathit{MinPts} * (\mathit{LRD}(p) + \varepsilon_i)} \\
\Rightarrow \quad & \frac{(k-1)\mathit{LRD}(p) + (\mathit{MinPts} - k + 1)\mathit{LRD}(q)}{\mathit{MinPts} * (\mathit{LRD}(p) + \varepsilon_i)} \leq \mathit{LOFUB} \\
\Rightarrow \quad & (\mathit{MinPts} + 1)\mathit{LRD}(q) - \mathit{LRD}(p) \\
& \leq \mathit{LOFUB} * \mathit{MinPts}(\mathit{LRD}(p) + \varepsilon_i) + k * (\mathit{LRD}(q) - \mathit{LRD}(p)) \\
\Rightarrow \quad & k \geq \frac{(\mathit{MinPts} + 1)\mathit{LRD}(q) - (\mathit{LOFUB} * \mathit{MinPts} + 1)\mathit{LRD}(p) - \mathit{LOFUB} * \mathit{MinPts} * \varepsilon_i}{\mathit{LRD}(q) - \mathit{LRD}(p)}
\end{aligned}
$$

$$\therefore LBC = \left[\frac{(MinPts+1)LRD(q) - (LOFUB * MinPts + 1)LRD(p)}{LRD(q) - LRD(p)} \right] + 1$$

Since the LOF of objects deep in a cluster is approximately equal to 1, the LOFUB must be greater than 1. Then

$$\begin{aligned} LBC &= \left[\frac{(MinPts+1)LRD(q) - (LOFUB * MinPts + 1)LRD(p)}{LRD(q) - LRD(p)} \right] + 1 \\ &< \left[\frac{(MinPts+1)LRD(q) - (MinPts + 1)LRD(p)}{LRD(q) - LRD(p)} \right] + 1 = MinPts + 2. \end{aligned}$$

In other words, LBC satisfies the inequality, $LBC \leq MinPts + 1$, discussed in part (a). Let's consider another extreme situation. The LOFUB is so big that $(LOFUB * MinPts + 1) * LRD(p)$ is bigger than $(MinPts + 1) * LRD(q)$, and in this case LBC is less than 1. As a matter of fact, it is impossible for LBC to be less than 1. When LOFUB is big enough, the object p which is a single point outlier still satisfies the core point condition, $LOF(p) \leq LOFUB$; therefore, the object p is deemed as a core point that should belong to a certain cluster. In this case, it forms a cluster which contains only one object by itself. □

5.3 The cluster-based outlier factor

Since outliers are far more than a binary property (Breunig et al. 2000), a cluster-based outlier also needs a value to demonstrate its degree of being an outlier. In the following we give the definition of the cluster-based outlier factor.

Definition 15 (Cluster-based outlier factor) Let C_1 be a cluster-based outlier and C_2 be the nearest non-outlier cluster of C_1. The cluster-based outlier factor of C_1 is defined as

$$CBOF(C_1) = |C_1| * dist(C_1, C_2) * \sum_{p_i \in C_2} lrd(p_i)/|C_2|.$$

The cluster-based outlier factor of the cluster C_1 is the result of multiplying the number of the objects in C_1 by the product of the distance between C_1 and its nearest normal cluster C_2 and the average local reachability density of C_2. The outlier factor of cluster C_1 captures the degree to which we call C_1 an outlier. Assume that C_1 as a cluster-based outlier is deviated from its nearest normal cluster C_2. It is easy to see that the more objects C_1 contains, and the farther away C_1 is from C_2, and the more dense C_2 is, the higher the CBOF of C_1 is and the more abnormal C_1 is.

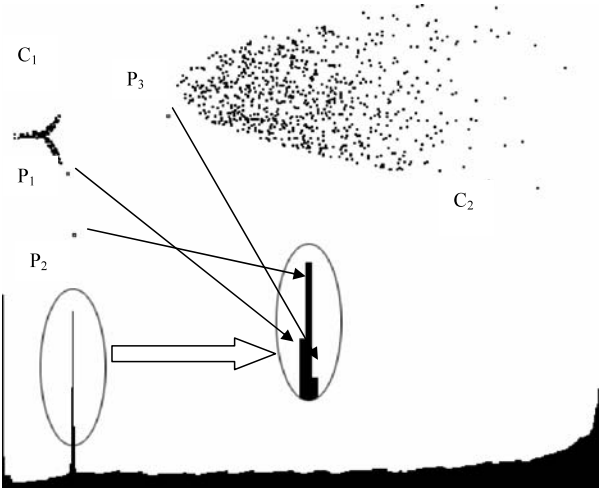
6 Experiments

A comprehensive performance study has been conducted to evaluate our algorithm. In this section, we describe those experiments and their results. The algorithm was run on both real-life datasets obtained from the UCI Machine Learning Repository and synthetic datasets.

6.1 Comet-like data

In order to demonstrate the accuracy of the clustering results of LDBSCAN, both LDBSCAN and OPTICS are applied to a 2-dimension dataset shown in the following Fig. 5.

Fig. 5 Comet-like clusters



LDBSCAN discovers the cluster C_1 consisting of small rectangle points, the cluster C_2 consisting of small circle points, and the outlier P_1, P_2, P_3 denoted by hollow rectangle points. OPTICS discovers the clusters whose reachability-distance falls into the dents and assigns the point to a cluster according to its reachability-distance, regardless its neighborhood density. Because the reachability-distance of the point P_3 is similar to that of the points in the right side of the cluster C_2, the side whose density is relatively low, OPTICS would assign the point P_3 to the cluster C_2, while LDBSCAN discovers the point P_3 as an outlier due to its different local density from its neighbors. Although both OPTICS and LDBSCAN can discover the points P_1, P_2 as outliers, the clustering result of OPTICS is not accurate especially when the border density of a cluster varies, such as the comet-like cluster.

6.2 Wisconsin breast cancer data

The second used dataset is the Wisconsin breast cancer data set, which has 699 instances with nine attributes, and each record is labeled as benign (458 or 65.5%) or malignant (241 or 34.5%). In order to avoid the situation in which the local density can be ∞ if there are more than MinPts objects, different from each other, but sharing the same spatial coordinates, only 3 duplicates of certain spatial coordinates are reserved and the rest are removed. In addition, the 16 records with missing values are also removed. Therefore, the resultant dataset has 327 (57.8%) benign records and 239 (42.2%) malignant records.

The algorithm processed the dataset when $pct = 0.5$, $LOFUB = 3$, $MinPts = 10$, and $\alpha = 0.95$. Both LOF and our algorithm find the 4 following noise records which are sing point outliers shown in Table 1. Understandably, our algorithm processes based on the result of LOF, and thus both can find the same single point outliers.

Besides the single point outliers, our algorithm discovers 3 clusters shown in Table 2, among which there are 2 big clusters and 1 small cluster. One big cluster A contains 296 benign records and 6 malignant records, and the other one B contains 26 benign records and 233 malignant records. The small cluster C contains only 1 record p. Among all the MinPts-nearest neighbors of the only one record in C, six neighbors belong to the cluster A and the other four belong to the cluster B. The record p is in the middle of cluster A and B, and $LOF(p) = 1.795$. It is closer to A than B, but has the similar local reachability density

Table 1 Single point outliers in Wisconsin breast cancer dataset

Sample code number	Value	Type	LOF
1033078	2, 1, 1, 1, 2, 1, 1, 1, 5	Benign	3.142
1177512	1, 1, 1, 1, 10, 1, 1, 1, 1	Benign	4.047
1197440	1, 1, 1, 2, 1, 3, 1, 1, 7	Benign	3.024
654546	1, 1, 1, 1, 2, 1, 1, 1, 8	Benign	4.655

Table 2 Clusters in Wisconsin breast cancer dataset

Cluster name	Number of benign records	Number of malignant records	Average local reachability density
A	296	6	0.743
B	26	233	0.167
C	1	0	0.170

to B rather than A. Thus, it forms a cluster by itself. This kind of special record cannot be easily discovered by LOF when its MinPts-nearest neighborhood overlaps with more than one cluster.

6.3 Boston housing data

The Boston housing dataset, which is taken from the StatLib library, concerns housing values in suburbs of Boston. It contains 506 instances with 14 attributes. Before clustering, data need to be standardized in order to assign each variable an equal weight. Here the z-score process is used because using mean absolute deviation is more robust than using standard deviation (Han and Kamber 2006). The mean absolute deviation is calculated: $s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \cdots + |x_{nf} - m_f|)$ where $m_f = \frac{1}{n}(x_{1f} + x_{2f} + \cdots + x_{nf})$. And then the standardized measurement (z-score) is retrieved: $z_{if} = \frac{x_{if} - m_f}{s_f}$. The algorithm processed the dataset when $pct = 0.5$, $LOFUB = 2$, $MinPts = 10$, and $\alpha = 0.9$. One single point outlier, 3 normal clusters and 6 cluster-based outliers are discovered. There are few single point outliers in this dataset. The maximum LOF, the value of the 381st record, is 2.624 which indicates that there is not a significant deviation. In addition, the 381st record is assigned to the 9^{th} cluster which is a cluster-based outlier. Its LOF exceeds LOFUB due to the small number of the objects contained in the 9^{th} cluster to which it belongs. The small number, which is less than MinPts, would affect the accuracy of LOF. Eight of all the nine records whose LOF exceeds LOFUB are assigned to a certain cluster and the LOF of the only single point outlier, the 215^{th} record, is 2.116. The 215^{th} record has a smaller proportion of owner-occupied units built prior to 1940, the 7^{th} attribute, than its neighbors.

However, the 6 cluster-based outliers are more interesting than the only single point outlier. Table 3 demonstrates the information of all the 9 clusters, and the additional information of the cluster-based outliers is shown in Table 4. The 3^{rd} cluster, which is a cluster-based outlier and has the maximum CBOF, deviates from the 1^{st} cluster. Its 12^{th} attribute, $1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town, is much lower than that of the 1^{st} cluster. Both the 9^{th} cluster and the 6^{th} cluster deviate from the 1^{st} cluster. Although the 6^{th} cluster contains more object than the 9^{th} cluster, the CBOF of the 6^{th} cluster is less than that of the 9^{th} cluster because the 9^{th} cluster is farther away from the 1^{st} cluster than the 6^{th} cluster. The records in the 9^{th} cluster have significantly big per capita crime rate by town,

Table 3 Clusters in Boston housing dataset

Cluster Id	Number of records	Average local reachability density
1	82	0.556
2	345	0.528
3	26	0.477
4	34	0.266
5	1	0.303
6	9	0.228
7	1	0.228
8	1	0.155
9	6	0.127

Table 4 Cluster-based outliers in Boston housing dataset

Cluster Id	CBOF	Nearest cluster	$dist(C_1, C_2)$	The nearest object pair	The contained records
3	54.094	1	3.744	436–445	412, 416, 417, 420, 424, 425, 426, 427, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 446, 451, 455, 456, 457, 458, 467
9	24.514	1	7.353	415–385	381, 406, 411, 415, 419, 428
6	20.005	1	4.000	399–401	366, 368, 369, 372, 399, 405, 413, 414, 418
7	2.452	2	4.648	103–35	103
5	2.269	1	4.084	410–461	410
8	1.468	4	5.522	284–283	284

comparing with those of the 1st cluster. However, it is not easy to do not differentiate the records in the 6th cluster from those of the 1st cluster. Moreover, the relationship between the 4th cluster and the 8th cluster is also impressive. There are 35 records which show that its tract bounds the Charles River, demonstrated by the 4th attribute, in the whole dataset, and 34 of them is discovered in the 4th cluster. The only exceptional record, the 284th record, has a slightly high proportion of residential land zoned for lots over 25,000 square feet, the 2nd attribute, and a relatively low proportion of non-retail business acres per town, the 3rd attribute. The area denoted by the 284th record is more like a residential area than the other areas along the Charles River.

6.4 Abnormal network throughput detection

The cluster-based outlier detection has been applied to the Backbone Anomaly Detection System for CSTNET, an Internet Service Provider for all the institutes of Chinese Academy of Sciences. The Backbone Anomaly Detection System continuously monitors the input and output throughput of about 300 network nodes of CSTNET. Each node generates its average throughput record every five minutes, so the Backbone Anomaly Detection System checks the node state $300 * 12 = 3600$ times each hour. Network throughput has the characteristic that are consistent with self-similarity (Crovella and Bestavros 1997). Under normal circumstances, the throughput of a certain node forms a cluster. When abnormal events happen, the throughput might drastically change. But during the period of an abnormal event,

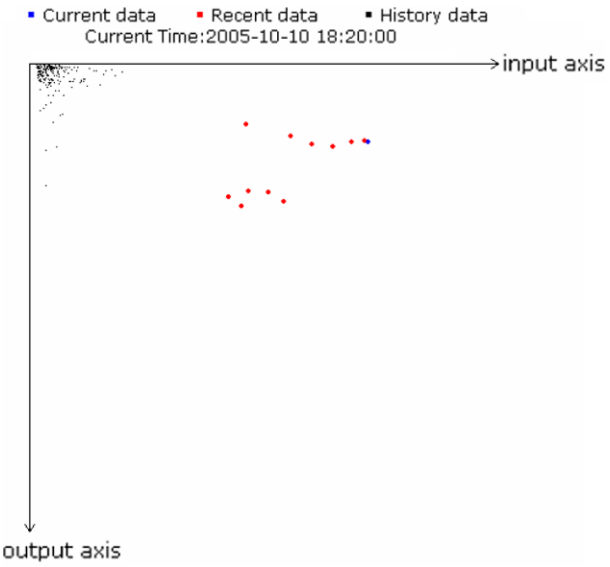


Fig. 6 An example of an abnormal event

the throughput exhibits temporal locality, i.e., it forms a new cluster which is different from history. The following Fig. 6 is an example of an abnormal event. If applying LOF, the best algorithm for single point outlier detection, the Backbone Anomaly Detection System generates 50 alerts per hour. And many of the alerts are related to normal occasional fluctuations. If applying cluster-based outlier detection, the system generates 10 alerts per hour. Some alerts of cluster-based outlier detection might not relate to real abnormal events, but according to the feedback of the network administrators in CSTNET, cluster-based outlier detection generates more accurate and reasonable alerts than single point outlier detection.

7 Conclusion

Outlier detection is attractive for the task of detecting unusual patterns in spatial database; however, previous researches fail to realize the importance of discovering cluster-based outliers. In this paper, an outlier detection algorithm which relies on the clustering result of LDBSCAN is proposed. Experiments have been carried out on both real data and synthetic data. The experiments show that the proposed algorithm provides more accurate clusters than OPTICS and outperforms LOF on identifying meaningful and interesting outliers.

There are several opportunities for future research. The guideline to discover the cluster-based outliers needs to be improved. In addition, the effectiveness of our algorithm is diminishing due to the distance function with the increasing dimension of data space (Beyer et al. 1999; Hinneburg et al. 2000). An effective distance function for high-dimensional data is desired. With it, our algorithm can yield a reasonable result in high-dimensional spaces as well.

References

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Record*, *27*(2), 94–105. doi:10.1145/276305.276314.

Ankerst, M., Breunig, M. M., Kriegel, H., & Sander, J. (1999). OPTICS: ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD international conference on management of data* (pp. 49–60). SIGMOD'99, Philadelphia, Pennsylvania, United States, May 31–June 03, 1999. New York: ACM Press.

Barnett, V., & Lewis, T. (1994). *Outliers in statistical data*. New York: Wiley.

Beyer, K. S., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is "nearest neighbor" meaningful? In C. Beeri & P. Buneman (Eds.), *Lecture notes in computer science: Vol. 1540. Proceeding of the 7th international conference on database theory* (pp. 217–235). January 10–12, 1999. London: Springer.

Breunig, M. M., Kriegel, H., Ng, R. T., & Sander, J. (2000). LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on management of data* (pp. 93–104). SIGMOD'00, Dallas, Texas, United States, May 15–18, 2000. New York: ACM Press.

Carvalho, R., & Costa, H. (2007). Application of an integrated decision support process for supplier selection. *Enterprise Information Systems*, *1*(2), 197–216. doi:10.1080/17517570701356208.

Crovella, M. E., & Bestavros, A. (1997). Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, *5*(6), 835–846.

Duan, L., Xu, L., Guo, F., Lee, J., & Yan, B. (2007). A local-density based spatial clustering algorithm with noise. *Information Systems*, *32*(7), 978–986. doi:10.1016/j.is.2006.10.006.

Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noises. In *Proc. 2nd int. conf. on knowledge discovery and data mining* (pp. 226–231). AAAI Press: Portland.

Guha, S., Rastogi, R., & Shim, K. (1998). CURE: an efficient clustering algorithm for large databases. In A. Tiwary & M. Franklin (Eds.), *Proceedings of the 1998 ACM SIGMOD international conference on management of data* (pp. 73–84). SIGMOD'98 Seattle, Washington, United States, June 01–04, 1998. New York: ACM Press.

Han, J., & Kamber, M. (2006). *Data mining: concepts and techniques*. Amsterdam: Elsevier.

Hawkins, D. (1980). *Identification of outliers*. London: Chapman and Hall.

He, Z., Xu, X., & Deng, S. (2003). Discovering cluster-based local outliers. *Pattern Recognition Letters*, *24*(9–10), 1641–1650. doi:10.1016/S0167-8655(02)00160-5.

Hinneburg, A., & Keim, D. 1998. An efficient approach to clustering in large multimedia databases with noise. In *Proc. 4th int. conf. on knowledge discovery and data mining* (pp. 58–65). New York.

Hinneburg, A., Aggarwal, C. C., & Keim, D. A. (2000). What is the nearest neighbor in high dimensional spaces? In A. E. Abbadi, M. L. Brodie, S. Chakravarthy, U. Dayal, N. Kamel, G. Schlageter, & K. Whang (Eds.), *Proceedings of the 26th international conference on very large data bases* (pp. 506–515). Very large data bases, September 10–14, 2000. San Francisco: Morgan Kaufmann Publishers.

Hsu, C., & Wallace, W. A. (2007). An industrial network flow information integration model for supply chain management and intelligent transportation. *Enterprise Information Systems*, *1*(3), 327–351. doi:10.1080/17517570701504633.

Jiang, M. F., Tseng, S. S., & Su, C. M. (2001). Two-phase clustering process for outliers detection. *Pattern Recognition Letters*, *22*(6–7), 691–700.

Johnson, T., Kwok, I., & Ng, R. (1998). Fast computation of 2-dimensional depth contours. In *Proc. 4th int. conf. on knowledge discovery and data mining* (pp. 224–228). New York: AAAI Press.

Knorr, E. M., & Ng, R. T. (1998). Algorithms for mining distance-based outliers in large datasets. In A. Gupta, O. Shmueli, & J. Widom (Eds.), *Proceedings of the 24rd international conference on very large data bases* (pp. 392–403). Very large data bases, August 24–27, 1998. San Francisco: Morgan Kaufmann Publishers.

Knorr, E. M., & Ng, R. T. (1999). Finding intensional knowledge of distance-based outliers. In M. P. Atkinson, M. E. Orlowska, P. Valduriez, S. B. Zdonik, & M. L. Brodie (Eds.), *Proceedings of the 25th international conference on very large data bases* (pp. 211–222). Very large data bases, September 07–10, 1999. San Francisco: Morgan Kaufmann Publishers.

Li, H., & Xu, L. (2001). Feature space theory—a mathematical foundation for data mining. *Knowledge-Based Systems*, *14*(5–6), 253–257. doi:10.1016/S0950-7051(01)00103-4.

Li, H., Xu, L., Wang, J., & Mo, Z. (2003). Feature space theory in data mining: transformations between extensions and intensions in knowledge representation. *Expert Systems*, *20*(2), 60–71. doi:10.1111/1468-0394.00226.

Luo, J., Xu, L., Jamont, J., Zeng, L., & Shi, Z. (2007). Flood decision support system on agent grid: method and implementation. *Enterprise Information Systems*, *1*(1), 49–68. doi:10.1080/17517570601092184.

Ng, R., & Han, J. (2002). CLARANS: a method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, *14*(5), 1003–1016.

Preparata, F., & Shamos, M. (1988). *Computational geometry: an introduction*. Berlin: Springer.

Qiu, G., Li, H., Xu, L., & Zhang, W. (2003). A knowledge processing method for intelligent systems based on inclusion degree. *Expert Systems*, *20*(4), 187–195. doi:10.1111/1468-0394.00243.

Ramaswamy, S., Rastogi, R., & Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on management of data* (pp. 427–438). SIGMOD'00, Dallas, Texas, United States, May 15–18, 2000. New York: ACM Press.

Sheikholeslami, G., Chatterjee, S., & Zhang, A. (1998). WaveCluster: a multi-resolution clustering approach for very large spatial databases. In A. Gupta, O. Shmueli, & J. Widom (Eds.), *Proceedings of the 24rd international conference on very large data bases* (pp. 428–439). Very large data bases, August 24–27, 1998. San Francisco: Morgan Kaufmann Publishers.

Shi, Z., Huang, Y., He, Q., Xu, L., Liu, S., Qin, L., Jia, Z., Li, J., Huang, H., & Zhao, L. (2007). MSMiner-a developing platform for OLAP. *Decision Support Systems*, *42*(4), 2016–2028. doi:10.1016/j.dss.2004.11.006.

Tukey, J. W. (1977). *Exploratory data analysis*. Reading: Addison–Wesley.

Wang, W., Yang, J., & Muntz, R. R. (1997). STING: a statistical information grid approach to spatial data mining. In M. Jarke, M. J. Carey, K. R. Dittrich, F. H. Lochovsky, P. Loucopoulos, & M. A. Jeusfeld (Eds.), *Proceedings of the 23rd international conference on very large data bases* (pp. 186–195). Very large data bases, August 25–29, 1997. San Francisco: Morgan Kaufmann Publishers.

Xu, L. (2006). Advances in intelligent information processing. *Expert Systems*, *23*(5), 249–250. doi:10.1111/j.1468-0394.2006.00405.x.

Xu, L., Liang, N., & Gao, Q. (2008). An integrated approach for agricultural ecosystem management, *IEEE Transactions on Systems Man and Cybernetics, Part C*, *38*(3).

Zhang, M., Xu, L., Zhang, W., & Li, H. (2003). A rough set approach to knowledge reduction based on inclusion degree and evidence reasoning theory. *Expert Systems*, *20*(5), 298–304. doi:10.1111/1468-0394.00254.

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. In J. Widom (Ed.), *Proceedings of the 1996 ACM SIGMOD international conference on management of data* (pp. 103–114). SIGMOD'96 Montreal, Quebec, Canada, June 04–06, 1996. New York: ACM Press.