Knowl Inf Syst (2005) 00: 1–41
DOI 10.1007/s10115-005-0233-6

Knowledge and
Information Systems

REGULAR PAPER

Jian Tang · Zhixiang Chen ·
Ada Waichee Fu · David W. Cheung

Capabilities of outlier detection schemes in large datasets, framework and methodologies

Received: 6 May 2004 / Revised: 30 November 2004 / Accepted: 5 July 2005

Abstract Outlier detection is concerned with discovering exceptional behaviors of objects. Its theoretical principle and practical implementation lay a foundation for some important applications such as credit card fraud detection, discovering criminal behaviors in e-commerce, discovering computer intrusion, etc. In this paper, we first present a unified model for several existing outlier detection schemes, and propose a compatibility theory, which establishes a framework for describing the capabilities for various outlier formulation schemes in terms of matching users' intuitions. Under this framework, we show that the density-based scheme is more powerful than the distance-based scheme when a dataset contains patterns with diverse characteristics. The density-based scheme, however, is less effective when the patterns are of comparable densities with the outliers. We then introduce a connectivity-based scheme that improves the effectiveness of the density-based scheme when a pattern itself is of similar density as an outlier. We compare density-based and connectivity-based schemes in terms of their strengths and weaknesses, and demonstrate applications with different features

J. Tang (✉)
Department of Computer Science, Memorial University of Newfoundland, St. John's,
Newfoundland, Canada
E-mail: jian@cs.mun.ca

Z. Chen
Department of Computer Science, University of Texas-Pan American Edinburgh, Texas, USA
E-mail: chen@cs.panam.edu

A. W. Fu
Department of Computer Science and Engineering, Chinese University of Hong Kong
Shatin, Hong Kong
E-mail: adafu@cse.cuhk.edu.hk

D. W. Cheung
Department of Computer Science and Information Systems, University of Hong Kong,
Pokfulam, Hong Kong
E-mail: dcheung@csis.hku.hk

where each of them is more effective than the other. Finally, connectivity-based and density-based schemes are comparatively evaluated on both real-life and synthetic datasets in terms of recall, precision, rank power and implementation-free metrics.

Keywords Outlier detection · Scheme capability · Distance-based outliers · Density-based outliers · Connectivity-based outliers · Performance metrics

1 Introduction

Outlier detection is concerned with discovering exceptional behaviors of certain objects. Revealing these behaviors is important since it signifies that something out of the ordinary has happened and shall deserve people's attention. In many cases, such exceptional behaviors will cause damages to the users and must be stopped. Therefore, in some sense detecting outliers is at least as significant as discovering general patterns. Outlier detection schemes lay a foundation in many applications, for instances, calling card fraud in telecommunications, credit card fraud in banking and finance, computer intrusion in information systems [13, 15, 25, 32], to name a few.

Harkins [18] characterizes an outlier in a quite intuitive way as follows: "An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism." Following the spirit of this definition, researchers have proposed various schemes for outlier detection. A large amount of the work was done under the general topic of clustering [14, 16, 27, 31, 34], where clustering algorithms are used to detect outliers as by-products of the clustering processes. The rational of using clustering algorithms to detect outliers is based on the understanding that outliers and cluster objects are mutually complemental. That is, an outlier shall not be in any cluster, and a cluster object shall not be an outlier. For example, the DBSCAN clustering algorithm in [14] explicitly defines data objects outside any density-based clusters as outliers (or noises). However, the outliers discovered this way are highly dependent on the clustering algorithms used and hence subject to the clusters generated. In [9], we presented some formal study about complementarity of outliers and cluster objects, focusing on the well-established density-based clustering method DBSCAN. Interestingly, we showed that in some restricted cases outliers and cluster objects are indeed complemental, however, in general, they are not regardless of parameter settings used in the clustering algorithm.

Most methods in the early work that detect outliers independently have been developed in the field of statistics [6, 18]. These methods normally assume some knowledge about the underlying distribution of a dataset. In reality, however, prior knowledge about the distribution of a dataset is not always obtainable. Besides, these methods do not scale well for even modest number of dimensions as the size of the dataset increases.

In the current literature, some outlier detection schemes have been proposed that are not subject to any clustering algorithms, and do not require any prior knowledge about the underlying distributions of the dataset. These can be basically categorized into *distance-based* schemes [2, 5, 23, 24, 28] and *density-based* schemes [8, 21].[1] Distance-based schemes are originated from the proposal in [23], called $DB(n, v)$-outliers, where n and v are parameters. Let $\mathcal{D}$ be the dataset. For any $p \in \mathcal{D}$ and any positive value v, define $N_v(p) = \{o : dist(p, o) \leq v \;\&\; o \neq p \;\&\; o \in \mathcal{D}\}$ (called the v-neighborhood of p). If $|N_v(p)| < n$, then p is called an outlier with respect to n and v, otherwise it is not. A prominent variation of the distance-based scheme is proposed in [28], called (t, k)-nearest neighbor scheme. For each object, its k-distance is defined as the distance to its k-nearest neighbor(s). Among all the objects, the top t with the maximum k-distances are outliers. As will be noted in later sections, this scheme is actually a special case of $DB(n, v)$-outliers.

The work in [8] gives the framework for density-based schemes. Let $p, o \in \mathcal{D}$. The reachability distance of p with respect to o for k is defined as:

$$reach\text{-}dist_k(p, o) = max\{k\text{-}distance(o), dist(p, o)\}.$$

The reachability distance smooths the fluctuation of the distances between p and its "close" neighbors. The local reachability density of p for k is defined as:

$$lrd_k(p) = \left(\frac{\sum_{o \in N_{k\text{-}distance(p)}(p)} reach\text{-}dist_k(p, o)}{|N_{k\text{-}distance(p)}(p)|} \right)^{-1}.$$

That is, $lrd_k(p)$ is the inverse of the average reachability distance from p to the objects in its k-distance neighborhood. The local outlier factor (LOF) of p is defined as

$$LOF_k(p) = \frac{\sum_{o \in N_{k\text{-}distance(p)}(p)} \frac{lrd_k(o)}{lrd_k(p)}}{|N_{k\text{-}distance(p)}(p)|}.$$

The value on the right side is the average fraction of the reachability densities of p's *k-distance* neighbors and that of p. Thus, as pointed out in [8], the lower the density of p, or the higher the densities of p's neighbors, the larger the value of $LOF_k(p)$, which indicates that p has a higher degree of being an outlier.

Note that the original version of the density-based scheme does not explicitly categorize the objects into either outliers or non-outliers. The LOF value of an object measures how strong it can be an outlier. However, when explicit classification is desirable, we can choose a *cut-off threshold* to determine whether or not an object is an outlier, depending on whether its LOF value is less than the cut-off threshold or not. In the following, therefore, we assume that the cut-off threshold is also a parameter.

In this paper, we study the modeling power of outlier detection schemes. We first use a generic model to represent distance-based and density-based schemes, and then propose a compatibility theory, which is a list of gradually relaxing criteria for describing the capabilities of a scheme to match users' intuitions. We then introduce a new scheme, called the connectivity-based outlier factor (COF)

[1] These schemes can be generally viewed as unsupervised learning. There are also much work done on outlier detections for specific domains under the framework of supervised learning. Supervised outlier detections are not the concern of this paper.

scheme, for outlier formulation. We compare COF and LOF schemes with the criteria developed in the compatibility theory, and use empirical analysis to demonstrate applications with different features where one scheme is more effective than the other.

The rest of this paper is organized as follows. In Sect. 2, we give a general model for outlier detection and introduce the compatibility theory. In Sect. 3, we use the compatibility theory to evaluate the capabilities of the distance-based and the density-based schemes. In Sect. 4, we introduce the connectivity-based COF scheme, and compare it with the density-based LOF scheme. In Sect. 5, we introduce recall, precision, rank power and implementation-free metrics for evaluating the performance of an outlier detection scheme. We report evaluation results of the COF scheme in comparison with the LOF scheme, based on experiments on both real-life and synthetic datasets. Execution time and scalability results are also presented. Finally, in Sect. 6, we conclude the paper by summarizing the main results and introducing problems for future research.

2 A framework for the capabilities of outlier detection schemes

2.1 A generic model for outlier detection

All the outlier detection schemes mentioned in the previous section contain some parameterized conditions applied to individual objects in a dataset. Let $\mathcal{D}$ be a multi-dimensional dataset. In general, we define an outlier detection scheme on $\mathcal{D}$ as a condition *Cond(o; P)*, where $o \in \mathcal{D}$ and P is a parameter set. A given assignment of values to the parameters in P is called a *parameter setting*. In the subsequent discussions, we use sets of values to denote parameter settings. (For example, for the parameter set $\{x, y\}$, the set $\{1, 2\}$ is the parameter setting: $x = 1$ and $y = 2$.) For a given parameter setting P', an object $o \in \mathcal{D}$ is an outlier if $Cond(o; P') = true$, or is a non-outlier if $Cond(o; P') = false$. We call *Cond(o; P)* an *outlier detection scheme* (or simply a *scheme*) for $\mathcal{D}$.

Under this framework, we have the following:

- $DB(n, v)$-outlier scheme: $Cond(o; \{n, v\}) \equiv |\ N_v(o)\ | < n$.
- (t, k)-nearest neighbor scheme: $Cond(o; \{t, k\}) \equiv |\ N_v(o)\ | < k$, where $v = (k\text{-}distance_t + k\text{-}distance_{t+1})/2$ and $k\text{-}distance_t$ is the $t$sth largest $k\text{-}distance$ in the dataset.[2]
- $LOF_k(o, u)$ scheme:

$$Cond(o; \{k, u\}) \equiv \frac{\sum_{p \in N_{k\text{-}distance(o)}(o)} \frac{lrd_k(p)}{lrd_k(o)}}{|N_{k\text{-}distance(o)}(o)|} \geq u.$$

Note that the second formulation mentioned earlier implies that the (t, k)-nearest neighbor scheme is a special case of the $DB(n, v)$-outlier scheme, thus we will not discuss it separately.

[2] For simplicity, we assume that different objects have different k-distances here.

2.2 Two levels of outlying properties

Outliers are identified by their distinct properties. These properties may be *conceptual*, e.g., unauthorized usage of credit cards, or *physical*, e.g., the amount involved per transaction. Conceptual properties exist in users' minds, while physical properties are described by input data. A programmable scheme can use physical properties for outlier detection. These are included as the operands in the conditions in the generic model mentioned earlier. It is worth noting here that a user having an expectation of outliers/non-outliers does not mean he/she knows where they are in the dataset. It merely implies that according to the conceptual properties he/she has about the outliers, each object has a unique interpretation, either an outlier or a non-outlier.

Definition 2.1 *Let $\mathcal{D}$ be a finite dataset. An expectation of $\mathcal{D}$ is a partition $\mathcal{D} = \mathcal{D}_o \cup \mathcal{D}_n$, where $\mathcal{D}_o$ and $\mathcal{D}_n$ denote the outlier set and the non-outlier set, respectively.*

In reality, the conceptual properties of outliers are not always described by their physical properties. For example, whether the credit card usage is legal may not be corresponding to any combination of the amount involved, the timing for the transaction, the user account, etc. Thus, the outliers detected by any specific scheme may not always match what a user expects.[3]

The purpose of using parameters is to adapt a scheme to the environment in which a user invokes that scheme. In the general case, different parameter settings identify different sets of outliers in the dataset. Given a user's expectation, if the sets of outliers and non-outliers can be identified by some parameter setting of a scheme, then we say that the scheme *has the capability* of detecting outliers as expected by the users, otherwise, it does not. Consider a dataset that contains two patterns, C_1 and C_2, and an outlier o from the user's expectation. (Thus, o does not belong to either pattern.) Suppose o has a set of physical properties P_1 that distinguishes it from C_1, and another set of physical properties P_2 that distinguishes it from C_2. Ideally, a scheme has the ability to detect o if and only if it can use a single parameter setting to describe both properties. This may be not possible, however, if the scheme envisions some conflict between P_1 and P_2, which cannot be resolved by any parameter setting. But in some cases, the conflict may be resolved separately by two parameter settings. This implies that the scheme has been weakened. This motivates the following framework.

2.3 Compatibility requirements

Again, we let $\mathcal{D}$ be a dataset, and $\mathcal{D}_o$ and $\mathcal{D}_n$ be an expectation for $\mathcal{D}$.

Definition 2.2 *An outlier detection scheme $Cond(o; P)$ for $\mathcal{D}$ is ON-compatible with the expectation $\mathcal{D}_o$ and $\mathcal{D}_n$ if there exists a parameter setting S, such that*

(1) $\forall p \in \mathcal{D}_o \;\; [Cond(p; S) = true]$, *and*
(2) $\forall p \in \mathcal{D}_n \;\; [Cond(p; S) = false]$.

[3] It may also occur that a user does not have any conceptual property in mind, implying that the result returned by any scheme is acceptable. In this paper, we do not consider this case.

ON-compatibility imposes the strongest requirement, and therefore characterizes the ideal capability that any scheme can achieve, in terms of matching the user's expectation.

Definition 2.3 *An outlier detection scheme $Cond(o; P)$ for $\mathcal{D}$ is O-compatible with the expectation $\mathcal{D}_o$ and $\mathcal{D}_n$ if there exist a sequence of parameter settings $\langle S_1, \ldots, S_m \rangle$ where $m \geq 1$, such that*

(1) $\forall p \in \mathcal{D}_o \ \forall i \in \{1, \ldots, m\} \ [Cond(p; S_i) = true]$*, and*
(2) $\forall p \in \mathcal{D}_n \ \exists j \in \{1, \ldots, m\} \ [Cond(p; S_j) = false]$.

O-compatibility uses a sequence of parameter settings. Each outlier is identified by all the settings, and each non-outlier is identified by at least one setting. Thus, different settings can be used to accommodate patterns with diverse characteristics. O-compatibility has a dual, called N-compatibility, defined as follows.

Definition 2.4 *An outlier detection scheme $Cond(o; P)$ for $\mathcal{D}$ is N-compatible with the expectation $\mathcal{D}_o$ and $\mathcal{D}_n$ if there exist a sequence of parameter settings $\langle S_1, \ldots, S_m \rangle$ where $m \geq 1$, such that*

(1) $\forall p \in \mathcal{D}_n \ \forall i \in \{1, \ldots, m\} \ [Cond(p; S_i) = false]$*, and*
(2) $\forall p \in \mathcal{D}_o \ \exists j \in \{1, \ldots, m\} \ [Cond(p; S_j) = true]$.

That is, N-compatibility requires that each outlier is identified by at least one parameter setting, and each non-outlier is identified by all the settings. In a sense O- and N-compatibilities are symmetric. With O-compatibility, all the outliers have compatible properties with respect to each pattern, whereas with N-compatibility, each outlier has compatible properties with respect to all the patterns. The results derived for one can be reformulated for the other. In the following discussions, therefore, we will mainly consider O-compatibility.

Definition 2.5 *An outlier detection scheme $Cond(o; P)$ for $\mathcal{D}$ is an O-cover for the expectation $\mathcal{D}_o$ and $\mathcal{D}_n$ if there exists a parameter setting S such that $\forall p \in \mathcal{D}_o \ [Cond(p; S) = true]$. It is an N-cover if there exists a parameter setting S' such that $\forall p \in \mathcal{D}_n \ [Cond(p; S') = false]$.*

The O-cover implies that a scheme will detect all outliers for a specific expectation, but may have "false positives." That is, the scheme may misclassify some non-outliers as outliers. Similarly, the N-cover implies that a scheme will detect all non-outliers for a specific expectation, but may have "false negatives." That is, the scheme may misclassify some outliers as non-outliers. "False positives" or "false negatives" may occur when the conceptual properties of the outliers grossly mismatch their physical properties. For example, if the credit card transaction made by an illegal user turns out to be quite normal when the involved amount, timing, frequency, etc. are viewed separately, then a scheme that examines these properties in an isolated fashion may not detect the transaction made by the illegal user.

The following lemma draws a three-layer hierarchy among the compatibilities defined earlier.

Lemma 2.1 *If an outlier detect scheme is ON-compatible, then it is both O-compatible and N-compatible. If it is O-compatible (N-compatible), then it is an O-cover (N-cover).*

The proof of the lemma mentioned earlier follows directly from Definitions 2.2–2.5.

The compatibilities proposed earlier are based on *perfect matching* of a user's expectation, rather than *error rate*. Our intention is to untie the effects of any parameter selection from the capability of a scheme. In the general case, error rates depend on specific parameter values used by a scheme, and a high error rate may simply mean that the specific parameter values are not right, not necessarily an indication of the quality of the underlying outlier detection scheme.[4]

In the following, we present two theorems that will be useful in proving that a scheme is not compatible.

Theorem 2.1 *An outlier detection scheme $Cond(o; P)$ for $\mathcal{D}$ is not ON-compatible with any given expectation $\mathcal{D}_o$ and $\mathcal{D}_n$ if for any parameter setting S, there exist $a \in \mathcal{D}_o$ and $b \in \mathcal{D}_n$ such that $Cond(a; S) = true \Rightarrow Cond(b; S) = true$.*

Proof. Assume the contrary. Then there is a parameter setting S that satisfies the two properties in Definition 2.2. For the setting S, the condition of the theorem implied that there are $a \in \mathcal{D}_o$ and $b \in \mathcal{D}_n$ such that $Cond(a; S) = true \Rightarrow Cond(b; S) = true$. Property (1) in Definition 2.2 indicates that $Cond(a; S) = true$, hence $Cond(b; S) = true$, a contradiction to property (2) in Definition 2.2. □

Theorem 2.2 *Let $\mathcal{S}$ be the set of all the parameter settings for a given outlier detection scheme $Cond(o; P)$ for $\mathcal{D}$. Then the following hold:*

(1) $Cond(o; P)$ is not O-compatible with an expectation $\mathcal{D}_o$ and $\mathcal{D}_n$ if
$\exists\, a \in \mathcal{D}_n\ \forall S \in \mathcal{S}\ \exists\, b \in \mathcal{D}_o\ [Cond(b; S) = true \Rightarrow Cond(a; S) = true]$.

(2) It is not N-compatible with the expectation $\mathcal{D}_o$ and $\mathcal{D}_n$ if
$\exists\, a \in \mathcal{D}_o\ \forall S \in \mathcal{S}\ \exists\, b \in \mathcal{D}_n\ [Cond(a; S) = true \Rightarrow Cond(b; S) = true]$.

Proof. We prove part (1) only, since the proof for part (2) is similar. Assume the contrary. Then $Cond(o; P)$ is O-compatible with $\mathcal{D}_o$ and $\mathcal{D}_n$. Let $\langle S_1, \ldots, S_m \rangle$ be the sequence of settings mentioned in Definition 2.3. Thus, by property (1) of Definition 2.3, for any i with $1 \leq i \leq m$ and any $p \in \mathcal{D}_o$, we have $Cond(p; S_i) = true$. For the given object $a \in D_n$ in the first condition of the theorem, by property (2) of Definition 2.3, there is a j with $1 \leq j \leq m$ such that $Cond(a; S_j) = false$. Since we have $Cond(p; S_j) = true$ for every object $p \in \mathcal{D}_o$, we have $Cond(b; S_j) = true$ for the particular object $b \in D_o$ given in the first condition of the theorem. Hence, by this condition, we have $Cond(a; S_j) = true$, a contradiction to the fact $Cond(a; S_j) = false$ devised earlier. □

In the following two sections, we show how to use the techniques mentioned earlier to evaluate the capabilities of various outlier detection schemes.

3 Evaluating capabilities of outlier detection schemes

We will use the following notations throughout the rest of the paper. For any dataset C and any object o, $dist_{max}(o, C)$ and $dist_{min}(o, C)$ denote the largest and

[4] We will not address issues of possible parameter-free outlier detection schemes. To our best knowledge, there is no parameter-free outlier detection scheme reported in literature.

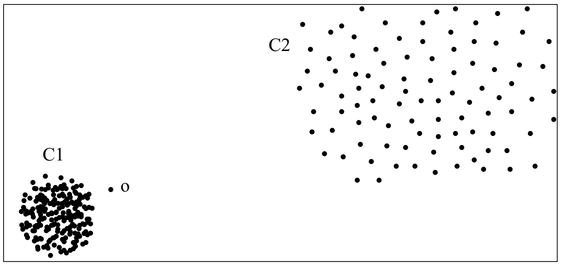


Fig. 1 DD-skewed dataset 1

smallest distances, respectively, from o to the points in C. (We will simply write $dist_{min}(o, C)$ as $dist(o, C)$.) For any dataset C, $diam(C)$ denotes the diameter of C, i.e, $diam(C) = max\{dist(p, q) : p, q \in C\}$. For any k, $k\text{-}distance_{max}(C) = max\{k\text{-}distance(r) : r \in C\}$, and $k\text{-}distance_{min}(C) = min\{k\text{-}distance(r) : r \in C\}$. For any two datasets C_1 and C_2, $dist(C_1, C_2) = min\{dist(p, q) : p \in C_1 \;\&\; q \in C_2\}$.

3.1 Distance and density skewed datasets

As discussed earlier, the effectiveness of an outlier detection scheme depends on the characteristics of datasets, the extent to which the physical properties of data can match the conceptual properties, and the way the physical properties are used by the scheme. In most practical datasets, the distances between objects, and the densities in the vicinities of objects are the two commonly available physical properties. For a given dataset, if at the conceptual level, both distances and densities are used to feature an outlier, but at different times (i.e., for different patterns) they may not match the physical properties of the data, then such a dataset is termed a *DD-skewed* dataset. (*DD* stands for *D*ensity and *D*istance).

Consider the dataset shown in Fig. 1.[5] This dataset contains an outlier o, a nearby dense pattern C_1, and a more distant sparse pattern C_2. Apparently, what distinguishes o from C_1 is the density, whereas what discriminates o from C_2 is the large distance between the two. This is because the local density of o is not lower than C_2, and hence is not a factor to disqualify o from being a member of C_2. DD-skewed datasets arise often in practices when data are generated from populations with a mixture of distributions.

3.2 The capability of the $DB(n, v)$-outlier scheme on DD-skewed datasets

This scheme essentially uses a parameter setting to define a size for the neighboring area for any object o, and an upper bound on the (average) density of the area. Intuitively, it works best for outliers possessing low neighborhood densities, but may be awkward for outliers with other properties. Thus, it is no surprising that

[5] This is a slightly modified version of the dataset used in [8].

its capability is limited on DD-skewed datasets. To make the formal derivation possible, we quantify some values in the dataset in Fig. 1, as described in Dataset Example 1.

3.2.1 Dataset Example 1

We assume that the dataset in Fig. 1 satisfies the following three conditions.

1. $\textit{1-distance}_{max}(C_1) < \frac{1}{2}dist(o, C_1)$
2. $\forall d, 0 < \mathrm{d} \leq dist_{max}(o, C_1), \forall r \in C_2, \mid N_d(r) \mid \leq \mid N_d(o) \mid$
3. $\forall d, d > dist_{max}(o, C_1), \exists t \in C_1, \mid N_d(t) \mid \leq \mid N_d(o) \mid$

Condition 1 means that the distance between any two objects in C_1 is less than half of the distance between o and C_1. Condition 2 specifies a range for the radius for which o has a higher density than that of C_2's members. Condition 3 describes what happens in large neighboring areas. That is, the neighborhood density of o will exceed that for some members of C_1. This is because the distance between o and C_2 is smaller than that between C_1 and C_2. Thus, for areas with the radius in the specified range in condition 3, o will always get at least as many neighbors as the points in C_1 can get.

Let $\mathcal{D}_o = \{o\}$ and $\mathcal{D}_n = C_1 \cup C_2$ be the interpretation for $\mathcal{D}$. We have the following results.

Result 3.1 *For the DD-skewed dataset described in Dataset Example 1, the* $DB(n, v)$*-outlier scheme is not ON-compatible with the earlier interpretation* $\mathcal{D}_o$ *and* $\mathcal{D}_n$*.*[6]

Proof. Let (n, v) be an arbitrary parameter setting for the $DB(n, v)$-outlier scheme. First consider the case of $0 < v \leq dist_{max}(o, C_1)$. By condition 2 of Dataset Example 1, for any $r \in C_2$, it is true that $\mid N_v(o) \mid < n \Rightarrow \mid N_v(r) \mid < n$. Now consider the case of $v > dist_{max}(o, C_1)$. By condition 3 of Dataset Example 1, there is $t \in C_1$, such that $\mid N_v(o) \mid < n \Rightarrow \mid N_v(t) \mid < n$. It follows from the two cases mentioned earlier that for any parameter setting (n, v), there are $o \in \mathcal{D}_o$ and b (r or t) $\in \mathcal{D}_n$ such that $\mid N_v(o) \mid < n$ is true $\Rightarrow \mid N_v(b) \mid < n$ is true. Thus, Result 3.1 follows from Theorem 2.1. □

For the dataset mentioned earlier, the $DB(n, v)$-outlier scheme can be O-compatible if, compared with the dense pattern, the sparse pattern is sufficiently large and far away from o, as described in the following result.

Result 3.2 *For the DD-skewed dataset in Dataset Example 1, if in addition* $\mid C_2 \mid \geq \mid C_1 \mid +2$ *and* $dist(o, C_2) > diam(C_2) > dist_{max}(o, C_1)$*, then the* $DB(n, v)$*-outlier scheme is O-compatible with the interpretation* $\mathcal{D}_o$ *and* $\mathcal{D}_n$*.*

Proof. Let $v_1 = \frac{1}{2}dist(o, C_1)$ and $n_1 = 1$. By condition 1 in Dataset Example 1, $\mid N_{v_1}(o) \mid = 0 < n_1$, and for all $r \in C_1$, $\mid N_{v_1}(r) \mid \geq n_1 = 1$. Since $dist(o, C_2) > diam(C_2) > dist_{max}(o, C_1)$, we can choose v_2 such that $dist(o, C_2) > v_2 > diam(C_2) > dist_{max}(o, C_1)$. We let $n_2 = \mid C_1 \mid +1$. By the given condition, $\mid N_{v_2}(o) \mid = \mid C_1 \mid < n_2 \leq \mid C_2 \mid$, and for all $r \in C_2$, $\mid N_{v_2}(r) \mid$

[6] Since there is only one outlier in the interpretation, ON-compatibility is the same as N-compatibility. Thus, $DB(n, v)$-outlier scheme is not N-compatible either.

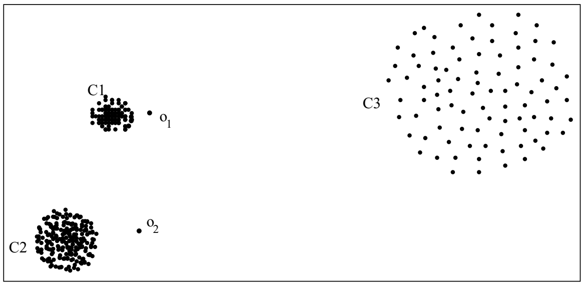


Fig. 2 DD-skewed dataset 2

$\geq | C_2 | -1 \geq n_2$. Thus, the settings (v_1, n_1) and (v_2, n_2) constitute the required sequence in Definition 2.3, hence Result 3.2 follows. □

Intuitively, the $DB(n, v)$-outlier scheme is O-compatible in a DD-skewed dataset if, with respect to each non-outlier pattern, all the outliers possess lower densities for at least one radius. In Fig. 1, for example, if there is another outlier near C_1, and far away from C_2, then O-compatibility would still hold, since they both would have lower densities with respect to C_1 in small radius, and also have lower densities with respect to C_2 in a larger distance (i.e., v_2 in the proof mentioned earlier). In cases where this condition is not true, the results become unpredictable. In the following, we present a dataset example where the dataset does not meet this criterion, and the $DB(n, v)$-outlier scheme is not O-compatible. Consider the dataset in Fig. 2. Its description is given in Dataset Example 2.

3.2.2 Dataset Example 2

The dataset in Fig. 2 contains two dense patterns C_1 and C_2, and a sparse pattern C_3, all being uniformly distributed, and two outliers o_1 and o_2. The dataset meets the following conditions.

1. $\frac{3}{4} \mid C_3 \mid \leq \mid C_1 \mid < \mid C_3 \mid < \mid C_2 \mid$
2. $4\times$ *1-distance*$_{max}(C_1) < dist(o_1, C_1)$
3. $\forall d, 0 < \text{d} \leq dist_{max}(o_1, C_1), \forall o \in C_3, \mid N_d(o) \mid \leq \mid N_d(o_1) \mid$
4. $4 \times$ *1-distance*$_{max}(C_2) < dist(o_2, C_2)$
5. $diam(\{o_1\} \cup C_1) < diam(\{o_2\} \cup C_2) < \frac{1}{2} diam(C_3)$
6. $\forall o \in C_3, \mid N_{\frac{1}{2}diam(C_3)}(o) \mid < \frac{3}{4} \mid C_3 \mid$
7. $\forall o \in C_3, dist(o, o_2) < dist(o, C_2)$
8. $max\{diam(\{o_1\} \cup C_1), diam(\{o_2\} \cup C_2)\} < dist(\{o_1\} \cup C_1, \{o_2\} \cup C_2)$
9. $diam(\{o_1\} \cup C_1 \cup \{o_2\} \cup C_2) < dist(\{o_1\} \cup C_1 \cup \{o_2\} \cup C_2, C_3)$

The conditions can essentially be categorized into two groups. The first group, conditions 2–4, 6, and 7, constrain the relative densities of the neighboring areas of the specified objects. For instance, condition 3 states that with the specified range of radius, the neighboring area of o_1 is denser than that of any point in C_3. Condition 6 requires that the sphere with a radius of $\frac{1}{2}diam(C_3)$ centered at any point in C_3 may not contain three quarter or more of the total points in C_3. The second group, conditions 1, 5, 8, and 9, exert group-based constraints. For example, condition 1 describes the relative sizes of groups, while the others constrain the diameters and inter-group distances. These conditions reflect the geometric structures depicted in Fig. 2.

Let $\mathcal{D}_o = \{o_1\} \cup \{o_2\}$ and $\mathcal{D}_n = C_1 \cup C_2 \cup C_3$ be the interpretation for the dataset $\mathcal{D}$ in Dataset Example 2. We have the following assertion.

Result 3.3 *The $DB(n, v)$-outlier scheme for the dataset described in Dataset Example 2 is not O-compatible.*

Informally, although o_1 and o_2 both have lower densities than C_1 and C_2 in a small radius (i.e., the smaller of $dist(o_1, C_1)$ and $dist(o_2, C_2)$), they do not show lower densities *simultaneously* than C_3 for any single radius. This is because for small to medium radius, the neighborhood of o_1 is denser than C_3, and for large radius the neighborhood of o_1 may be likely sparser than C_3, and all the points in C_2 will fall into the neighborhood of o_2, making it denser than C_3.

Proof. See Appendix A.

3.3 The capability of density-based schemes on DD-skewed datasets

The cardinality of the neighboring area of an object, as used by the $DB(n, v)$-outlier scheme, can be thought as the 'absolute' density of the object. Density-based schemes, on the other hand, use the relative density of an object, i.e., the comparison of its own density with the densities of its neighbors in a locality. The properties of any points or a group of points beyond this locality will not affect its relative density. This makes it more resilient to the skewness than the $DB(n, v)$-outlier scheme. We shall use Dataset Example 2 to exhibit our evaluation of the LOF scheme.

In order to simplify the formal analysis, we give two more conditions as stated in the following result. The first condition is just a quantification of the assumption of the uniform distribution for C_i for $i = 1, 2, 3$. The second is reasonable with respect to the geometric structure of the dataset in Fig. 2. (Note that these two additional conditions do not conflict with those specified in Dataset Example 2.)

Result 3.4 *Assume that the dataset in Dataset Example 2 satisfies two additional conditions:*

(a) $\frac{\textit{1-distance}_{min}(C_i)}{\textit{1-distance}_{max}(C_i)} \geq \frac{4}{5}$, *where* $1 \leq i \leq 3$*; and*

(b) $|N_1(o_j)| = 1$ *for* $j = 1, 2$.

Then, the LOF scheme is ON-compatible.

To simplify the presentation, we can let $k = 1$. Choosing a larger value for k will not change the proof in a fundamental way, but will nonetheless make the analysis more tedious. The idea is that, first, the nearest neighbor of o_1 must belong to C_1. Then, since the distance between o_1 and C_1 is much larger than the *1-distance* of any point in C_1, the value of $LOF_1(o_1)$ will be larger. On the other hand, since the *1-distances* of points in C_i, for $i = 1, 2, 3$, are roughly the same, the LOF values of these points will be small. Based on this observation, we can set a lower bound for $LOF_1(o_1)$ and an upper bound for the LOF value for any

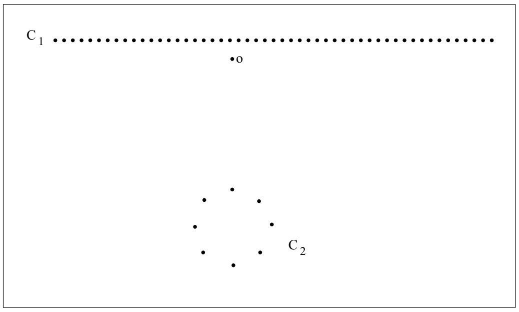


Fig. 3 A low-density pattern

point in C_1 (C_2 or C_3) and show that the former is larger than the latter. Similar arguments can be applied to o_2.

Proof. See Appendix B.

3.4 Limitation of the density-based scheme

One weakness of the density-based scheme is that it may rule out outliers that are shifting from a low-density pattern. To understand the problem, let us first take a closer look at the concept of pattern. According to the *Concise Oxford Dictionary*, a pattern is "a regular or logical form, order or arrangement of parts . . ." We observe that although a high density can reflect such a logical form, order or arrangement, it nonetheless is not a necessary condition, at least in the form defined in the current literature. As a result, an outlier does not always have to be of a lower density than the pattern it deviates.

For example, consider the dataset $\mathcal{D} = C_1 \cup C_2 \cup \{o\}$ of two-dimensional points as shown in Fig. 3.4, where $\mathcal{D}_o = \{o\} \cup C_2$. The pattern, C_1, is a straight line, which is of low density in a two-dimensional space. Since the outlier o shifts away from a low-density pattern, the density-based scheme will not be very effective to identify it, unless we use a small k. On the other hand, using too small a k will rule out the outliers in C_2, which can only be identified using a value for k larger than its cardinality. In Sect. 4.3.2, we will show the ineffectiveness of the LOF scheme in handling a similar case.

In the next section, we will introduce a new scheme that can handle low-density patterns such as the line of points in Fig. 3.4, while at the same time does not compromise detecting a group of staying-together outliers like those in C_2 in Fig. 3.4.

4 Connectivity-based outlier detection

4.1 Motivation

To cope with outliers with respect to low-density patterns, we differentiate "low density" from "isolativity." While low density normally refers to the fact that the number of points in the "close" neighborhood of an object is (relatively) small, isolativity refers to the degree that an object is "connected" to other objects. As a result, isolation can imply low density, but the other direction is not always true. For example, in Fig. 3.4 point o is isolated, while any point p in C_1 is not. But both of them are of roughly equally low density. In the general case, a low-density outlier results from deviating from a high-density pattern, and an isolated outlier results from deviating from a connected pattern.

We observe that patterns that possess low densities usually exhibit low dimensional structures. For example, a pattern shown in Fig. 3.4 is a line in the two-dimensional space. The isolativity of an object, on the other hand, can be described by the distance to its nearest neighbor. In the general case, we can also talk about the isolativity of a group of objects, which is the distance from the group to its nearest neighbor.

4.2 Concepts and definitions

In the following definitions, the function *dist*() has the same meaning as that defined in the previous sections, and $G = \{p_1, p_2, \ldots, p_r\}$ is a subset of dataset $\mathcal{D}$.

Definition 4.1 *Let $P, Q \subseteq \mathcal{D}$, $P \cap Q = \emptyset$ and $P, Q \neq \emptyset$. For any given $q \in Q$, we say that q is the nearest neighbor of P in Q if $dist(q, P) = dist(Q, P)$.*

Definition 4.2 *A set-based nearest path, or SBN-path, from p_1 on G is a sequence $\langle p_1, p_2, \ldots, p_r \rangle$ of all the elements in G such that for all $1 \leq i \leq r-1$, p_{i+1} is a nearest neighbor of set $\{p_1, \ldots, p_i\}$ in $\{p_{i+1}, \ldots, p_r\}$.*

Imagine that a set initially contains object p_1 only. Then it goes into an iterative expansion process. In each iteration, it picks up its nearest neighbor among the remaining objects. If its nearest neighbor is not unique, we can impose a predefined order among its neighbors to break the tie. Thus, an *SBN*-path is uniquely determined. An *SBN*-path indicates the order in which the nearest objects are presented.

Definition 4.3 *Let $s = \langle p_1, p_2, \ldots, p_r \rangle$ be an SBN-path from p_1 on G. A set-based nearest trail, or SBN-trail, with respect to s is a sequence $\langle e_1, \ldots, e_{r-1} \rangle$ such that for all $1 \leq i \leq r-1$, $e_i = (o_i, p_{i+1})$ where $o_i \in \{p_1, \ldots, p_i\}$, and $dist(e_i) = dist(o_i, p_{i+1}) = dist(\{p_1, \ldots, p_i\}, \{p_{i+1}, \ldots, p_r\})$. We call each e_i an edge and the sequence $\langle dist(e_1), \ldots, dist(e_{r-1}) \rangle$ the cost description of $\langle e_1, \ldots, e_{r-1} \rangle$.*

Again, if o_i is not unique, we should break the tie by a pre-defined order. Thus, the *SBN*-trail is unique for any *SBN*-path.

Definition 4.4 *Let $s = \langle p_1, p_2, \ldots, p_r \rangle$ be an SBN-path from p_1 on G, and $e = \langle e_1, \ldots, e_{r-1} \rangle$ be the SBN-trail with respect to s. The average chaining distance from p_1 on G, denoted by $ac\text{-}dist_G(p_1)$, is defined as*

$$ac\text{-}dist_G(p_1) = \sum_{i=1}^{r-1} \frac{2(r-i)}{r(r-1)} dist(e_i).$$

The average chaining distance from p_1 on G is the weighted sum of the cost description of the *SBN*-trail for the *SBN*-path from p_1 on G. Since this cost description is unique for p_1, our definition is well defined. Rewriting

$$ac\text{-}dist_G(p_1) = \frac{1}{r-1} \sum_{i=1}^{r-1} \frac{2(r-i)}{r}\, dist(e_i),$$

and viewing the fraction following the summation sign as the weight, this value can then be viewed as the average of the weighted distances in the cost description of the *SBN*-trail. Note that larger weights are assigned to the earlier terms in the *SBN*-trail. Thus, the edges closer to p_1 contribute more to $ac\text{-}dist_G(p_1)$ than the ones farther away. As a result, a point shifting away more from a pattern is likely to have a greater *ac-dist*. It is easy to see that, when all the $dist(e_i)$ are equal, $ac\text{-}dist_G(p) = dist(e_i)$ for all $p \in G$.

Note that by definition, for any object p, the k-nearest neighborhood $N_k(p)$ does not contain the object itself. The average chaining distance from p to its k-nearest neighbors is $ac\text{-}dist_{N_k(p)\cup\{p\}}(p)$. In the following, in order to simply notations, we let $ac\text{-}dist_k(p) = ac\text{-}dist_{N_k(p)\cup\{p\}}(p)$.

Definition 4.5 *Let $p \in \mathcal{D}$ and k be a positive integer. The connectivity-based outlier factor (COF) at p with respect to its k-nearest neighborhood is defined as*

$$COF_k(p) = \frac{|N_k(p)|\, ac\text{-}dist_k(p)}{\sum_{o\in N_k(p)} ac\text{-}dist_k(o)}.$$

We define the connectivity-based outlier detection scheme (or simply the COF scheme) under the generic outlier detection model as $Cond(p; \{k, u\}) \equiv COF_k(p) \geq u$. That is, p is an outlier with respect to parameters k and u if and only if $Cond(p; \{k, u\})$ is true.

The connectivity-based outlier factor at p is the ratio of the average chaining distance from p on $N_k(p)$ and the average of the average chaining distances from p's k-distance neighbors to their own k-distance neighbors. It indicates how far away a point shifts from a pattern. We now use an example to highlight the motivation behind it.

4.2.1 Dataset Example 3

Consider the dataset in Fig. 4.2.1. The pattern is a single line and two points shift away from it. Suppose $dist(1, 2) = 5$, $dist(2, 7) = 3$, and the distance between any two adjacent points in the line is 1. Let $k = 10$. We now calculate the average chaining distances for three sample points to show how the COF values of those sample points reflect "shifting from pattern" in an appropriate way.

For point 1: $N_k(1) = \{2, 9, 10, 8, 11, 7, 12, 6, 13, 5\}$. The *SBN*-path from 1 on $N_k(1)$ is

$$s_1 = \langle 1, 2, 7, 6, 5, 8, 9, 10, 11, 12, 13\rangle.$$

The *SBN*-trail for s_1 is

$$tr_1 = \langle (1, 2), (2, 7), (7, 6), (6, 5), (7, 8), (8, 9), (9, 10), (10, 11), (11, 12), (12, 13)\rangle.$$

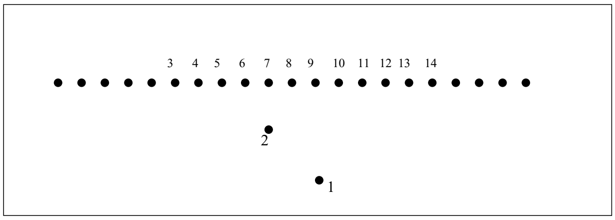


Fig. 4 Calculating COF

The cost description of tr_1 is $c_1 = \langle 5, 3, 1, 1, 1, 1, 1, 1, 1, 1 \rangle$, and $ac\text{-}dist_k(1) = 2.05$.

For point 2: $N_k(2) = \{7, 6, 8, 5, 9, 4, 10, 3, 11, 1\}$. The *SBN*-path from 2 on $N_k(2)$ is

$$s_2 = \langle 2, 7, 6, 5, 4, 3, 8, 9, 10, 11, 1 \rangle.$$

The *SBN*-trail for s_2 is

$$tr_2 = \langle (2, 7), (7, 6), (6, 5), (5, 4), (4, 3), (7, 8), (8, 9), (9, 10), (10, 11), (2, 1) \rangle.$$

The cost description of tr_2 is $c_2 = \langle 3, 1, 1, 1, 1, 1, 1, 1, 1, 5 \rangle$, and $ac\text{-}dist_k(2) = 1.46$.

For point 7: $N_k(7) = \{6, 8, 5, 9, 4, 10, 2, 3, 11, 12\}$. The *SBN*-path from 7 on $N_k(7)$ is

$$s_3 = \langle 7, 6, 5, 4, 3, 8, 9, 10, 11, 12, 2 \rangle.$$

The *SBN*-trail for s_3 is

$$tr_3 = \langle (7, 6), (6, 5), (5, 4), (4, 3), (7, 8), (8, 9), (9, 10), (10, 11), (11, 12), (7, 2) \rangle.$$

The cost description of tr_3 is $c_3 = \langle 1, 1, 1, 1, 1, 1, 1, 1, 1, 3 \rangle$, and $ac\text{-}dist_k(7) = 0.98$.

The average chaining distances for the other points on the line can be calculated similarly. The previous results show that for points that shift more from the pattern, such as points 1 and 2, the first few items in their cost description lists (or SBN-trails) tend to be larger than those for points that shift less, such as point 7. Since earlier items in a cost description list are assigned larger weights, they contribute more to the corresponding average chaining distance, which is the weighted sum of the values in the cost description. Thus, strongly shifted points will have larger average chaining distances than weakly shifted ones. In the general case, most points in the k-nearest neighborhood of a strongly shifted point should have small average chaining distances. This results in a larger connectivity-based outlier factor for such a strongly shifted point. On the other hand, for a weakly shifted point, most points in its k-nearest neighborhood should have comparable average chaining distance values, resulting in a smaller connectivity-based outlier factor for such a point. The weakest shifted points are those that belong to the pattern itself. Their connectivity-based outlier factors should be close to 1. For the three sample points in the previous example, for $k = 10$, we have the following:

$$COF_k(1) = 2.1, \;\; COF_k(2) = 1.35 \;\; \text{and} \;\; COF_k(7) = 0.96.$$

4.3 Capabilities of the COF scheme vs. the LOF scheme

We first show that, like the *LOF* scheme, the *COF* scheme is resilient to DD-skewed datasets. We then show that the *COF* scheme is more robust than the *LOF* scheme in detecting outliers with respect to low-density patterns. Finally, we use experimental results to show the strengths and weaknesses of both COF and LOF schemes.

4.3.1 Detecting outliers in DD-skewed datasets

Again, we use the dataset in Fig. 2 to illustrate an example of DD-skewed datasets. Our result is given in the following assertion.

Result 4.1 *Under the same conditions given in Result 3.4, the COF scheme is ON-compatible.*

The basic idea of the proof is similar to that for the LOF scheme in Result 3.4. It differs only in the details of calculation.

Proof. See Appendix C.

4.3.2 Detecting outliers with respect to low-density patterns

We have used the dataset in Fig. 3 as a typical example of outliers shifting away from low-density patterns. In order to simplify the analysis, we reshape the dataset but still retain its basic characteristics. The resulting dataset is shown in Fig. 5.

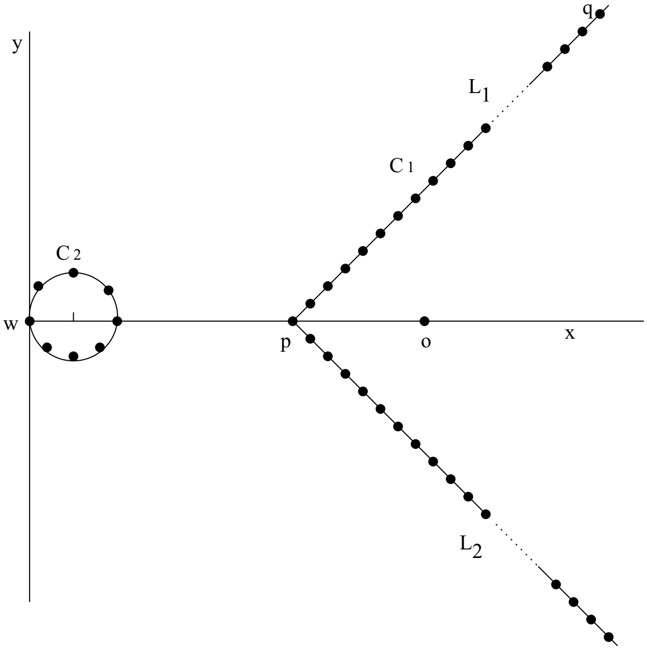


Fig. 5 Low-density patterns

4.3.3 Dataset Example 4

In Fig. 5, C_2 contains eight points lying on the circle with its center at $(1, 0)$ and a radius of 1. Distances between any adjacent points on the circle are the same. C_1 contains 91 points lying on two straight lines l_1 and l_2. The two lines meet at the point $p = (20, 0)$. Line l_1 and the x-axis form an angle of $\frac{\pi}{2}$, and so do line l_2 and the x-axis. C_1 contains p and 45 points on each of the lines l_1 and l_2. Moreover, the distance between any two adjacent points on each line is $\sqrt{2}$. Finally, $o = (23, 0)$. According to Hawkins' definition, it is easy to understand that point o and the points in C_2 are outliers while others are not. Thus, the expectation is $\mathcal{D} = \mathcal{D}_o \cup \mathcal{D}_n$, where $\mathcal{D}_o = \{o\} \cup C_2$ and $\mathcal{D}_n = C_1$.

Result 4.2 *For the dataset given in Dataset Example 4, the LOF scheme is not ON-compatible for the expectation $\mathcal{D}_o$ and $\mathcal{D}_n$.*

We support the previous assertion by the experimental data. We chose two non-outlier points $p = (20, 0)$ and $q = (65, 45)$ from $\mathcal{D}_n$ and two outlier points $w = (0, 0)$ and $o = (23, 0)$ from $\mathcal{D}_o$. The four points are illustrated in Fig. 5. Note that q is the end point of C_1 on line l_1. Note also that the total number of points in the dataset is 100. We calculated the LOF values for all those four points for $k = 1, 2, \ldots, 99$. The calculation was done by a C++ program with a precision of 10 decimal digits. The computing environment is a Dell Inspiron 8100 Pentium 1GHz laptop with 512 MB RAM and 20 GB HD. The LOF values of the four points are shown in Fig. 6. For $1 \leq k \leq 7$, we have $LOF_k(q) > LOF_k(w)$. Thus, for any value u, $LOF_k(w) \geq u \Rightarrow LOF_k(q) \geq u$. For $8 \leq k \leq 98$, we have $LOF_k(q) > LOF_k(o)$. This means $LOF_k(o) \geq u \Rightarrow LOF_k(q) \geq u$. For

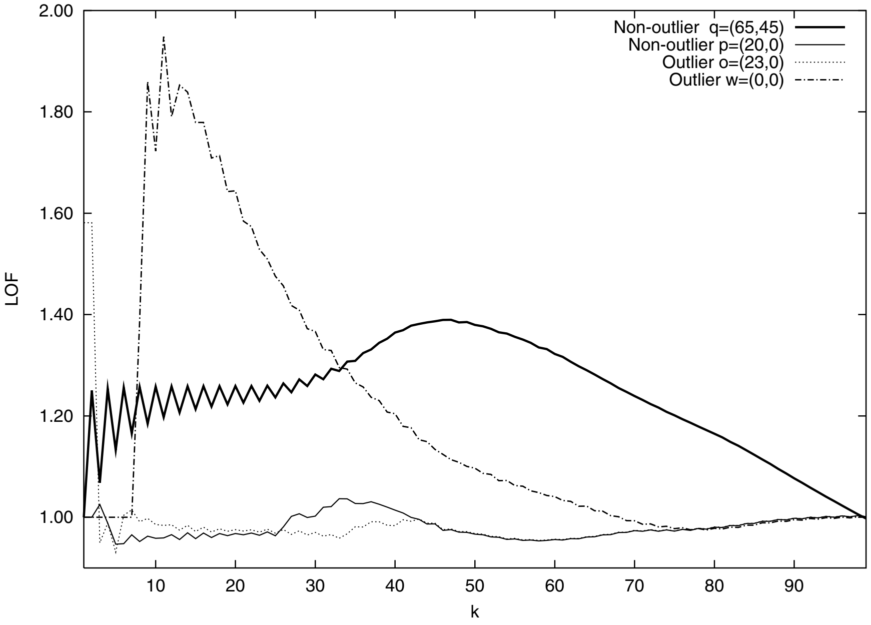


Fig. 6 LOF values of the four points

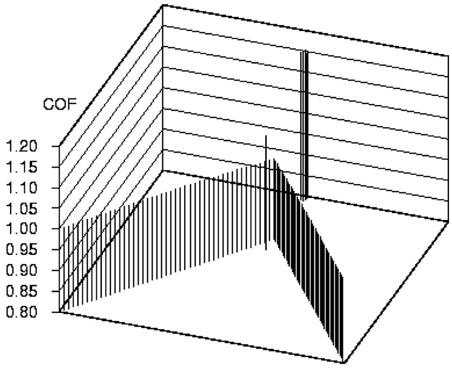


Fig. 7 COF values of all points

$k = 99$, we have $LOF_k(p) = 1.0013753983 > LOF_k(w) = 0.9992365171$, meaning $LOF_k(w) \geq u \Rightarrow LOF_k(p) \geq u$. Because p and q are non-outliers and o and w are outliers, by Theorem 2.1, the LOF scheme is not ON-compatible with the given expectation.

Result 4.3 *For the dataset in Dataset Example 4, the COF scheme is ON-compatible with the expectation $\mathcal{D}_o$ and $\mathcal{D}_n$ as defined in the example.*

This assertion is supported by the experimental result shown in Fig. 7. We chose $k = 13$ and calculate COF values for all points in the dataset. The calculation was done by a C++ program with a precision of 10 decimal digits. The computing environment is the same as that for Result 4.2. All the eight outliers in C_2 have the same COF value 1.1518705044 and the other outlier o has a COF value 1.0761474038. On the other hand, the first 15 points, starting from p on each of the two lines ℓ_1 and ℓ_2, have COF values between 0.9941766178 and 0.9995440551; and the rest of the points in C_1 have COF value of 1. Thus, we can set a threshold of 1.076 to distinguish the outliers from the non-outliers. Hence, by Definition 2.2, COF is ON-compatible with the expectation as defined in Dataset Example 4.

At this point, we would like to know if the LOF scheme is O- and/or N-compatible for the given dataset. The following assertion answers this question.

Result 4.4 *For the dataset and the expectation defined in Dataset Example 4, the LOF scheme is both O-compatible and N-compatible.*

We support this assertion by experiments. For each k starting from 1, we find the *floor of outliers*, which is the minimum of the LOF values for all the outliers. We then identify the maximal set of non-outliers, called *covered set of non-outliers*, whose LOF values are below the floor of outliers. If before k reaches the maximum (i.e., the size of the dataset $\mathcal{D}$), the union of all the covered sets identified so far is the entire set of non-outliers, then the LOF scheme is O-compatible, otherwise it is not. A similar procedure is used to determine if the LOF scheme is N-compatible, where we consider the *ceiling of non-outliers* and *covered set of outliers*.

Plotted in Fig. 8 is the experimental result. Each curve shows how the size of the union of the covered outliers/non-outliers increases as k increases. The figure

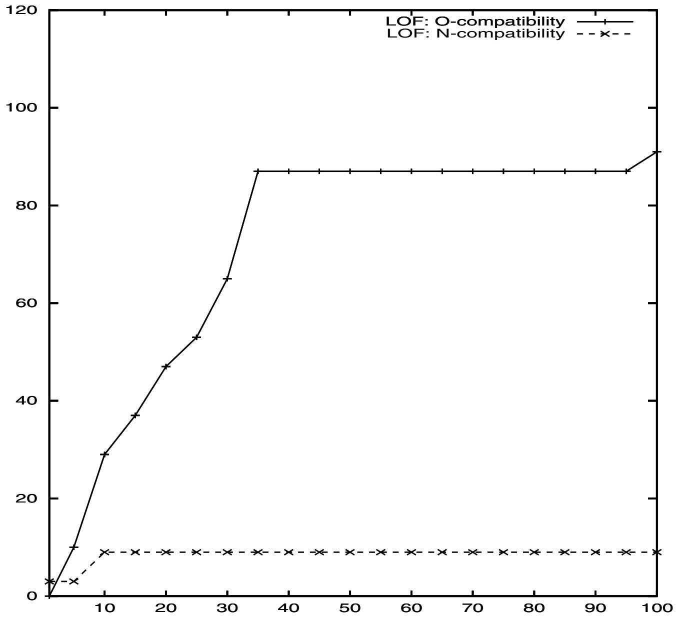


Fig. 8 LOF is O- and N-compatible

Table 1 *k* values for LOF O-compatibility

Sequence 1			Sequence 2		
k values	Floor of outliers	Non-outliers covered	*k* values	Floor of outliers	Non-outliers covered
9	0.9978	19	16	0.9800	19
18	0.9774	16	18	0.9774	6
26	0.9758	14	26	0.9758	14
45	0.9887	28	45	0.9887	28
99	0.9992	14	99	0.9992	24

Table 2 *k* values for LOF N-compatibility

Sequence 1			Sequence 2			Sequence 3		
k values	Ceiling of non-outliers	Outliers covered	*k* values	Ceiling of non-outliers	Outliers covered	*k* values	Ceiling of non-outliers	Outliers covered
1	1.0000	3	30	1.2818	8	9	1.1847	8
8	1.2572	6	99	1.0014	1	99	1.0014	1

shows that the entire set of 91 non-outliers is covered completely when *k* reaches the maximum, making the LOF scheme O-compatible, while the set of 9 outliers is covered when *k* reaches ten, implying N-compatibility of the LOF scheme.

In Tables 1 and 2 we give some sequences of *k* values for the O- and N-compatibilities for the LOF scheme. Please note that each number on the bottom row in each table indicates the number of outliers (non-outliers) that are covered

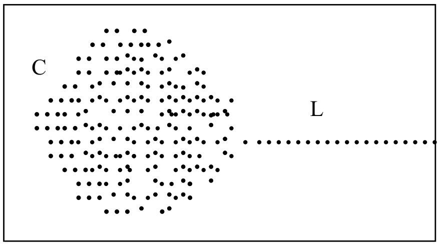


Fig. 9 Connected outliers

by the current k value but not by any of the preceding ones. Thus, the numbers of the covered outliers (non-outliers) in the same sequence sum up to the total number of outliers (non-outliers) in the dataset.

4.3.4 Detecting connected outliers

Although the connectivity-based COF scheme is more effective than the density-based scheme in detecting isolated outliers, it is not as effective in detecting "connected" outliers as the LOF scheme. Consider the dataset described in Dataset Example 5.

4.3.5 Dataset Example 5

The dataset shown in Fig. 9 contains a disk-shaped pattern C and a line pattern L, where $\mid C \mid = 180$ and $\mid L \mid = 20$. Assume the points in C are uniformly distributed, and so are the points in L. In addition, the average of the distances between adjacent points in L is roughly the same as the average of the $1\text{-}distance(p)$ for all $p \in C$. Let the expectation be $\mathcal{D} = \mathcal{D}_o \cup \mathcal{D}_n$, where $\mathcal{D}_o = L$ and $\mathcal{D}_n = C$. We have the following result.

Result 4.5 *For the dataset described in Dataset Example 5, the LOF scheme is not ON-compatible, but is both O-compatible and N-compatible with the expectation $\mathcal{D}_o$ and $\mathcal{D}_n$. However, the COF scheme is neither O-compatible nor N-compatible with the expectation.*

We again use the experimental result to justify the previous assertion. The result is shown in Fig. 10. The x-axis indicates k values, and the y-axis denotes the number of non-outliers that are not covered (i.e., their LOF values are not lower than the floor of the outliers for the corresponding k). It can be seen from Fig. 10 that for each k, the curve never touches zero. Thus, there is not a single value for k such that all the non-outliers are covered. This means that the LOF scheme is not ON-compatible. (The minimum number of non-outliers that escapes from being covered is 10. This occurs when $k = 183$.) The LOF scheme is, however, both O- and N-compatible. This is shown in Fig. 11. Some sequences of k values for O/N-compatibility are given in Tables 3 and 4.

For the COF scheme, Fig. 12 shows that it is neither O-compatible nor N-compatible. From the figure, when k reaches the maximum (the size of the

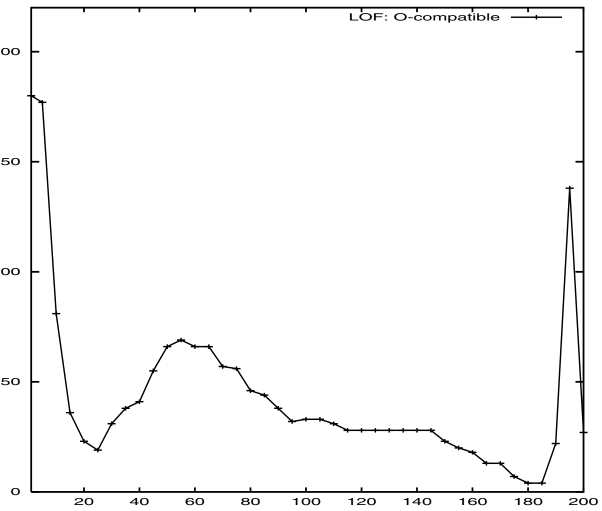


Fig. 10 LOF is not ON-compatible

dataset), 157 out of 180 non-outliers are covered, making COF not O-compatible, while 11 out 20 outliers are covered, making it not N-compatible.

The previous results are consistent with intuition. Since all the points in the dataset are evenly apart from their neighbors, they have the same connectivity. So the COF scheme cannot distinguish the outliers from non-outliers, but the LOF scheme can do better since the line pattern has a lower density than the disk-shaped pattern.

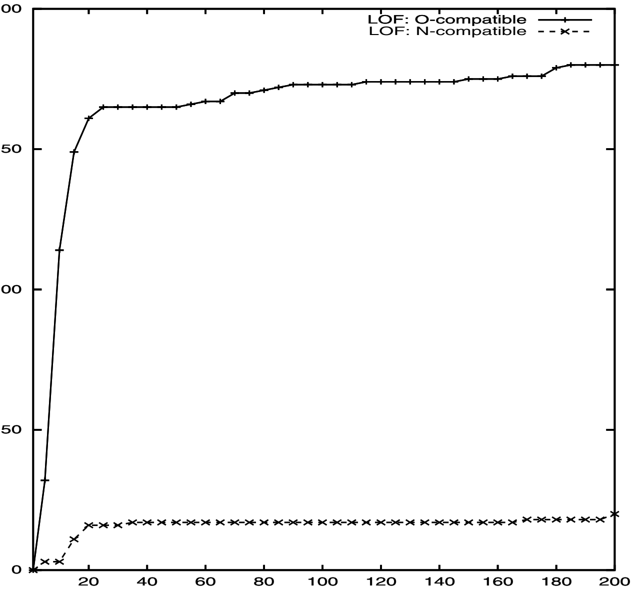


Fig. 11 LOF is O- and N-compatible

Table 3 *k* values for LOF O-compatibility

Sequence 1			Sequence 2		
k values	Floor of outliers	No. of non-outliers covered	*k* values	Floor of outliers	No. of non-outliers covered
2	0.9287	23	184	1.0056	178
22	1.1020	142	188	1.0040	1
181	1.010	15	198	0.9982	1

4.3.6 Time complexity

Although algorithmic aspects for finding the COF values is not the main concern of the present paper, we include a brief discussion about the complexity for completeness.

Suppose that the database $\mathcal{D}$ has n d-dimensional objects. Like the LOF algorithm in [8], we can compute COF values for objects in $\mathcal{D}$ in two major steps. The first step is preprocessing. In this step, we find all k-nearest neighborhoods and all average chain distances. Precisely, we find, for any object $p \in \mathcal{D}$, the k-nearest neighbors and then use the Prim's algorithm [12] to find the average chain distance of p. The result of this step is saved in an intermediate dataset $\mathcal{M}$. We know that the number of k-nearest neighbors of an object may be more than k, and in the worst case may be as large as $n-1$. But on average it is reasonable to assume that the number of k-nearest neighbors of an object is $O(k)$. Hence, the number of objects in the intermediate dataset $\mathcal{M}$ is nk. As in the LOF algorithm, the number of objects in $\mathcal{M}$ is independent of the dimensionality of the original dataset $\mathcal{D}$. Since each object have d many components, the size of $\mathcal{M}$ is $O(nkd)$. The time complexity for this step is $O(n^2d + nk^2d)$ because we need to consider every object in $\mathcal{D}$, and for each object we first need to search $\mathcal{D}$ to find its k-nearest neighbors, and then use the Prim's algorithm to find its average chain distance from those k-nearest neighbors.

In the second step, we compute, for any object $p \in \mathcal{D}$, the value of $COF_k(p)$ with the help of the intermediate dataset $\mathcal{M}$. The original dataset is not needed at this step, since $\mathcal{M}$ contains sufficient information. We scan the dataset $\mathcal{M}$ twice. In the first scan, we find the average chain distance $\textit{ac-dist}_k(p)$ and the k-nearest neighbors of p. In the second scan, we search for the average chain distances of those k-nearest neighbors. The time complexity is $O(|\mathcal{M}| + k|\mathcal{M}| + k) = O(nk^2d)$.

Table 4 *k* values for LOF N-compatibility

Sequence 1			Sequence 2		
k values	Ceiling of non-outliers	No. of outliers covered	*k* values	Ceiling of non-outliers	No. of outliers covered
1	1.0359	18	30	1.0165	18
8	1.0015	2	99	1.0015	2

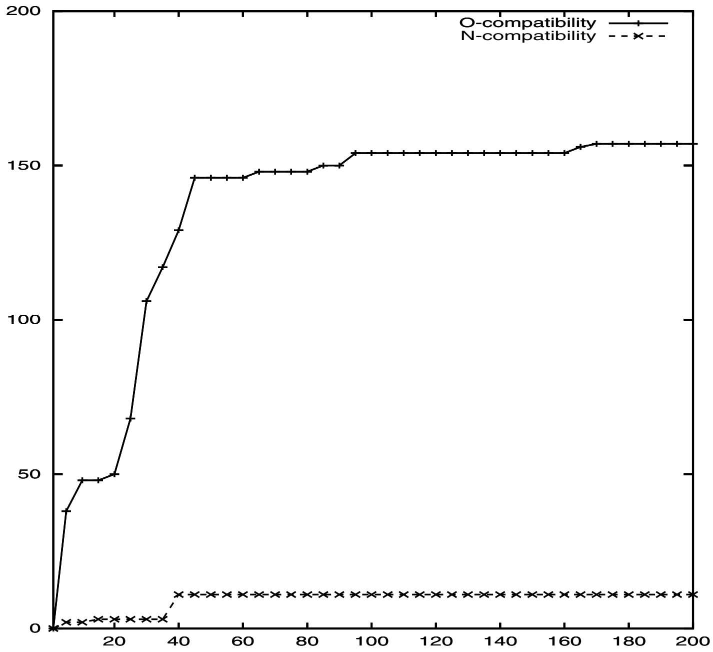


Fig. 12 COF is neither O-compatible nor N-compatible

Combining the two steps mentioned earlier, the time complexity of computing COF values of all objects in the dataset $\mathcal{D}$ is $O(n^2d+nk^2d)$. This time complexity is more efficient than that of the LOF algorithm in [8], because one can show that the LOF algorithm has $O(n^2k^3d)$ complexity.

5 Experimental results

We conducted performance experiments on both real-life and synthetic datasets to evaluate our proposed COF scheme. Since it is shown in [8] that the LOF scheme is more effective than the distance-based schemes, we will focus on comparing the COF scheme with the LOF scheme. The real-life datasets were obtained from the UCI Machine Learning Repository [7]. Algorithms were implemented in C++. The computing environment is a Dell Inspiron 8100 Pentium 1GHz laptop with 512 MB RAM and 20 GB HD.

As suggested in [1], one way to evaluate the performance of an outlier detection scheme is to test the scheme on datasets to discover rare classes. The performance of the scheme is then measured by the percentage of data, which are from the rare classes, discovered by the scheme. This approach was adopted by [17, 19, 20], and was used in our evaluation experiments.

In the following, we will introduce four metrics, namely, precision, recall, rank power and implementation-free metric, for measuring the performance of an outlier scheme.

5.1 Performance metrics

5.1.1 Precision and recall

These are the two traditional performance metrics of the quality of an information system [4, 30], and can be tailored to measure the performance of an outlier detection scheme. Assume that a dataset $\mathcal{D} = \mathcal{D}_o \cup \mathcal{D}_n$ with $\mathcal{D}_o$ being the set of all outliers and $\mathcal{D}_n$ being the set of all non-outliers. Given any integer $m \geq 1$, let O_m denote the set of outliers among the objects in the top m positions returned by an outlier detection scheme. Then, we define precision and recall with respect to m as follows:

$$\text{precision} = \frac{|O_m|}{m}, \qquad \text{recall} = \frac{|O_m|}{|\mathcal{D}_o|}.$$

That is, *precision* measures the percentage of outliers among the top m ranked objects returned by the scheme, while *recall* measures the percentage of the total outlier set included within the top m ranked objects. These relative precision and recall measures have been used in evaluating performances of Web search algorithms (see, f.g., [11]). Note that the LOF (COF) scheme ranks objects according to the LOF (COF) values of the objects. Objects with larger LOF (COF) values are ranked higher than these with smaller values. The distance-based scheme can also rank objects according to the sparseness of these objects within a given distance. The coverage rate metric used in [1, 19, 20] is essentially the recall measure.

5.1.2 Rank power

Precision and recall metrics certainly measure the accuracy of a scheme, but do not reflect the satisfaction level of the users, because both metrics ignore the importance of the placements of the outliers returned by the scheme. For example, placing three outliers in the top three positions is considered by both metrics the same as placing in the bottom three positions among m objects returned. In reality, users are mostly interested in top ranked results. That is, not only how many results being returned is important, but also where they are placed is critical. Rank power is a metric proposed in [26] that considers both the placements and the number of results returned by a scheme, and is bounded from below by $\frac{1}{2}$. A survey of other performance metrics can also be found in [26]. Here, we give a slightly revised definition of rank power with values ranging from 0 to 1 so that a value of 1 indicates the best performance and 0 indicates the worst. Consider that a scheme returns m objects, placing from position 1 to position m. Assume that there are n outliers among these m objects. For $1 \leq i \leq n$, let L_i denotes the position of the ith outlier, define the rank power of the scheme with respect to m as

$$\text{Rank Power}(m) = \frac{n(n+1)}{2\sum_{i=1}^{n} L_i}.$$

As can be seen from the previous definition, rank power weighs the placements of the returned outliers heavily. An outlier placed earlier in the returned list adds less to the denominator of the rank power (and thus contributes more to the rank power metric) than placed later in the list.

5.1.3 Implementation-free metric

When evaluating the time performance of a proposed approach vs. the competing approaches, the evaluation results may depend on data structures and indexing methods and other concrete techniques used in implementation. It is stated in [22] that *Implementation bias is the conscious or unconscious disparity in the quality of implementation of a proposed approach, vs. the quality of implementation of the competing approaches*." It is pointed out in [22] that one possibility to avoid implementation bias is to design experiments that are free from the possibility of implementation bias. For example, in artificial intelligence, researchers often compare search algorithms by reporting the number of nodes expanded, rather than the CPU time. To evaluate the time performance of the COF scheme vs. the LOF scheme, we introduce an implementation-free metric. Given any parameter $k \geq 1$, for any object p in the dataset, the time needed to compute $COF_k(p)$ is proportional to the size of $N_k(p) \bigcup \cup_{o \in N_k(p)} N_k(o)$, and the time needed to compute $LOF_k(p)$ is propositional to the size of

$$N_k(p) \bigcup \cup_{o \in N_k(p)} \left(N_k(o) \bigcup \cup_{q \in N_k(o)} N_k(q) \right).$$

Denote the first size as $C(k, p)$ and the second as $L(k, p)$. These two sizes are independent of implementation techniques, and hence we can use them as implementation-free metrics to measure the time performance of COF and LOF schemes.

5.2 Wisconsin breast cancer data

The first dataset used is Wisconsin Breast Cancer Dataset, which has 699 records with 9 attributes. Each record is labeled as *benign* or *malignant*. We found that there are many records occurring more than once in the dataset. In order to avoid data bias, another important issue advocated in [22] in performance evaluation, we removed all the duplicated records and records with missing attribute values, and obtained a dataset with 213 records labeled as *benign* and 236 as *malignant*. We then follow the experimental technique used in [1, 19, 20] to remove some of the *malignant* records to form a very unbalanced distribution. The resulting dataset, as shown in Table 5, has 213 (91.4%) *benign* records and 20 (8.6%) *malignant* records.

We ran both COF and LOF schemes on this dataset to find the rare cases with different values of the parameter k. Table 6 shows the performance results of the COF scheme in comparison with the LOF scheme. Here, the performance is measured with the three metrics of recall, precision, and rank power. The value of the parameter k is 12, which is 5% of the number of records in the dataset. For other values of k, the performance results are consistent with Table 6. The value of m indicates top m ranked records returned by the COF (or LOF) scheme. The ratio of these top m ranked records to the size of the dataset is also given in column 1. Column 2 indicates the number of rare cases among top m ranked records returned by the COF scheme, while column 6 indicates the number of rare cases among top m ranked records returned by the LOF scheme. The other six columns show values of recall, precision, and rank power for both COF and LOF schemes.

Table 5 The case distribution of Wisconsin Breast Cancer Dataset

Case	Class Code	Percentage of records
Commonly occurring classes	2 (benign)	91.4
Rare classes	4 (malignant)	8.6

Table 6 Detected rare cases in Wisconsin Breast Cancer Dataset

	COF scheme				LOF scheme			
m (top ratio)	Malignant records	Recall (%)	Precision (%)	Rank power	Malignant records	Recall (%)	Precision (%)	Rank power
5 (2%)	3	15	60	0.60	1	5	20	0.25
10 (4.3%)	5	25	50	0.58	5	25	50	0.44
15 (6.4%)	8	40	53	0.56	8	40	53	0.49
20 (8.6%)	12	60	60	0.57	10	50	50	0.47
25 (10.7%)	13	65	52	0.57	11	55	44	0.43
30 (12.9%)	15	75	50	0.54	11	55	34	0.43
35 (15%)	17	85	49	0.51	14	70	40	0.44
40 (17%)	18	90	45	0.47	15	75	36	0.43
45 (19%)	19	95	42	0.46	15	75	33	0.43
56 (24%)	20	100	36	0.45	18	90	32	0.39

For example, among top 25 ranked records returned by the COF scheme, 13 are rare cases, with 65% recall, 52% precision, and 0.57 rank power; while among top 25 ranked records returned by the LOF scheme, 11 are rare cases, with 55% recall, 44% precision, and 0.43 rank power. Among top 15 ranked records returned by the two schemes, both schemes detect eight rare cases, with the same 40% recall and the same 53% precision. However, the COF scheme has 0.56 rank power, which is larger than the 0.49 rank power of the LOF scheme. This implies that the COF scheme places these rare cases higher than the LOF scheme, hence performs better than the LOF scheme when placements of the results are considered. The last row indicates that the COF finds all the 20 rare cases among top 56 ranked records, while the LOF scheme still misses 2. The recall and precision measurements exhibit that the COF scheme outperforms the LOF scheme except for the two cases of top 10 and 15 ranked records. For these two cases, both schemes performs equally well in terms of recall and precision; however, the COF scheme performs better in terms of rank power.

5.3 Image segmentation data

This dataset contains 210 records with 19 attributes. These records form seven equally sized classes labeled respectively as *brickface*, *sky*, *foliage*, *cement*, *window*, *path*, and *grass*. There are no duplicated records nor records with missing attributes values. Following the similar approach as in [1, 19, 20], we removed some records from the dataset to generate rare cases. Precisely, we removed 27 records from each of the *brickface*, *grass*, and *path* classes. The resulting dataset

Table 7 The case distribution of Image Segmentation Dataset

Case	Class Code	Percentage of records
Commonly occurring classes	Sky, foliage, cement, window	93
Rare classes	Brickface, path, grass	7

Table 8 Detected rare cases in Image Segmentation Dataset

	COF scheme				LOF scheme			
m (top ratio)	Brickface, grass, path records	Recall (%)	Precision (%)	Rank power	Brickface, grass, path records	Recall (%)	Precision (%)	Rank Power
5 (3.9%)	1	11	20	0.25	1	11	20	0.25
10 (7.8%)	3	33	30	0.27	2	22	20	0.25
15 (11.6%)	4	44	27	0.29	2	22	13	0.25
20 (15.5%)	5	56	25	0.27	4	44	20	0.21
25 (19%)	7	78	28	0.27	4	44	16	0.21
30 (23%)	7	78	28	0.27	4	44	13	0.21
35 (27%)	8	89	23	0.26	5	56	14	0.12
40 (31%)	8	89	23	0.26	6	67	15	0.18
45 (35%)	9	100	23	0.25	6	67	13	0.48

has 129 records with 9 records as rare cases (3 for each of the *brickface*, *grass*, and *path* classes). Table 7 shows the class distribution of the dataset.

As for the first dataset, we ran both COF and LOF schemes on the second dataset to find the rare cases with different values of the parameter k. Table 8 shows the performance results measured with recall, precision, and rank power. The value of the parameter k is 7, which is 5% of the size of the dataset. Consistent performance results are obtained for other values of k. Again, the value of m indicates top m ranked records (or top ratio of records) returned by the COF (or LOF) scheme. Column 2 indicates the number of rare cases among top m ranked records returned by the COF scheme, while column 6 does the same for the LOF scheme. The other six columns assess recall, precision, and rank power for both COF and LOF schemes. For example, among top 15 ranked records returned by both schemes, the COF scheme detects four rare cases, with 44% recall, 30% precision, and 0.29 rank power, while the LOF scheme detects two rare cases, with 22% recall, 13% precision, and 0.25 rank power. Among top 45 ranked records, the COF scheme detects all the 9 rare cases, while the LOF scheme still misses 3. Results in Table 8 show that the COF scheme outperforms the LOF scheme except for the case of top 5 ranked records, where both schemes perform equally well in terms of three metrics.

5.4 Johns hopkins university ionosphere data

This dataset has 351 records with 34 attributes. These records form two classes labeled respectively as *good* and *bad*. There are no duplicated records nor records with missing attributes values. Following the similar experimental method as in

Table 9 The case distribution of Johns Hopkins University Ionosphere Dataset

Case	Class Code	Percentage of records
Commonly occurring classes	Good	95.7
Rare classes	Bad	4.3

Table 10 Detected rare cases in Johns Hopkins University Ionosphere Dataset

	COF scheme				LOF scheme			
m (top ratio)	Bad records	Recall (%)	Precision (%)	Rank power	Bad records	Recall (%)	Precision (%)	Rank power
5 (2.1%)	5	50	100	1.00	4	40	80	0.83
10 (4.3%)	7	70	70	0.85	6	60	60	0.75
15 (6.4%)	9	90	60	0.76	8	80	53	0.71
20 (8.5%)	10	100	50	0.69	9	90	45	0.67

[1, 19, 20], we removed some records from the dataset to generate rare cases. The resulting dataset has 235 records with 215 records labeled as *good* and 10 labeled as *bad*. Table 9 shows the class distribution of the dataset.

Again, we ran both COF and LOF schemes on the third dataset to test the performance of the COF scheme vs. the LOF scheme in terms of recall, precision, and rank power. The results are given in Table 10, where the parameter k is set to 12, which is 5% of the size of the dataset. Consistent performance results are obtained for other values of k. Once again, the value of m indicates top m ranked records (or top ratio of records) returned by the COF (or LOF) scheme. Columns 2 and 6 indicate respectively the number of rare cases among top m ranked records returned by COF and LOF schemes. The other six columns assess recall, precision, and rank power for both schemes. For example, among top five ranked records returned by both schemes, the COF scheme detects five rare cases, with 50% recall, 100% precision, and 1.00 rank power, while the LOF scheme detects four rare cases, with 40% recall, 80% precision, and 0.83 rank power. Among the top 20 ranked records, the COF scheme detects all the 10 rare cases, while the LOF scheme still misses 1. Results in Table 10 shows that the COF scheme outperforms the LOF scheme in terms of recall, precision, and recall.

5.5 COF scheme performance vs. outlier isolativity

In the previous section, experimental results on three real-life datasets show that the COF scheme outperforms the LOF scheme in terms of recall, precision, and rank power. As addressed in Sect. 4, one major motivation for us to introduce the COF scheme is to deal with "isolativity" of outliers. Isolativity implies low density, but the latter does not always imply the former. We show in Sect. 4 that the COF scheme can perform well in detecting isolated outliers that deviate from connected patterns, but the LOF scheme cannot do so well. Here, we examine why the COF scheme outperforms the LOF scheme on these three datasets via analyzing

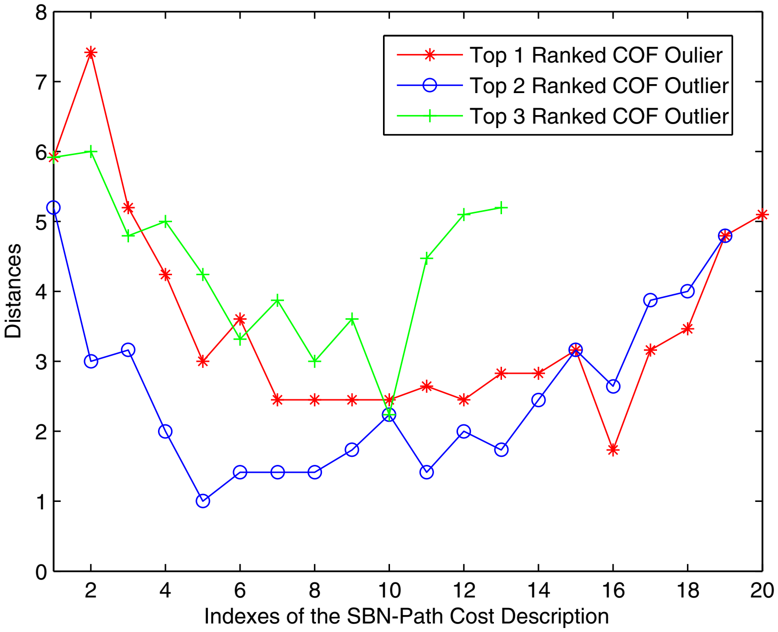


Fig. 13 Isolation evidences of Wisconsin Breast Cancer Dataset

isolativity of outliers. Since the three datasets have respectively dimensions of 9, 19, and 34, it is impossible to generate some direct visualization of the possible isolativity of the outliers. However, for an isolated outlier o, it is easy to see that the cost description of the SBN-path of o on $N_k(o)$ will reflect the isolativity of o. Therefore, we will analyze the cost descriptions of outliers in the datasets. To simplify the illustration, we chose top three ranked outliers returned by the COF scheme for each of the three datasets, and plotted the related cost descriptions in Figs. 13–15, respectively. In these figures, the x-axis indicates the indexes of the cost description of the SBN-path on the k-nearest neighborhood of the outlier, and the y-axis indicates the corresponding costs (i.e., the distances as defined in Definition 4.3), where $k = 12$ in Figs. 13 and 15, and $k = 7$ in Fig. 14. Recall from Definition 4.3 that, given an outlier o, the first cost in the cost description is the distance from o to its closest object in the dataset. In general, the ith cost is the distance from o and its $(i - 1)$ closest objects to the rest of the objects in the dataset. For example, in Fig. 14, for the top one ranked outlier returned by the COF scheme, the first cost is 111.317, meaning that this outlier is 111.37 distance away from the rest of the objects in the dataset. The second cost is 47.546, meaning that this outlier and its closest neighbor is 47.546 distance away from the rest of objects in the dataset. These costs in the three figures provide good evidences that the top three ranked outliers are isolated from the non-outliers. Similar isolations also exist for other outliers. These isolation properties of the outliers somehow provide evident support to the better performance of the COF scheme.

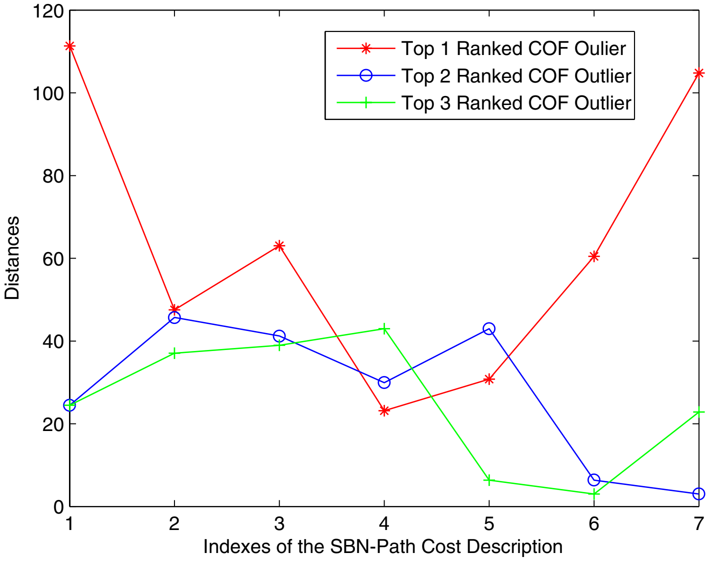


Fig. 14 Isolation evidences of the Ionosphere Dataset

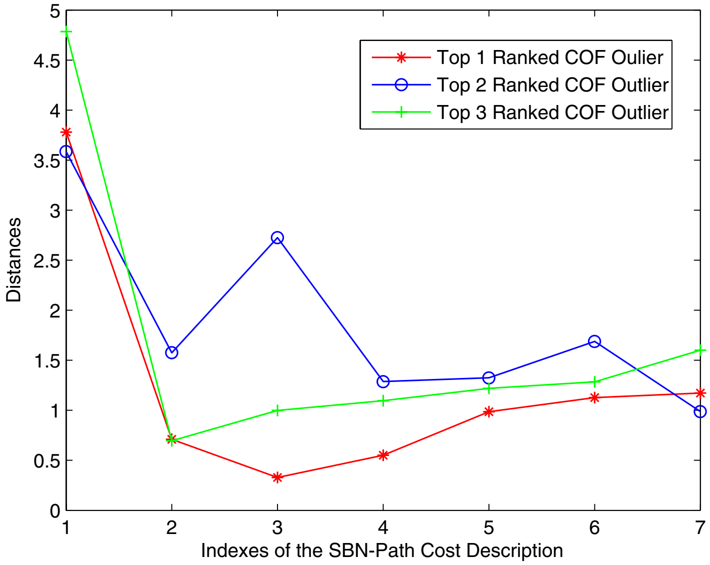


Fig. 15 Isolation evidences of Image Dataset

5.6 Implementation-free performance

In Sect. 5.1, we introduced an implementation-free metric to evaluate the time performance of both COF and LOF scheme. Given any parameter $k \geq 1$ and any object p in the dataset, the time needed to compute $COF_k(p)$ is proportional to

$$C(k, p) = |N_k p \bigcup \cup_{o \in N_k(p)} N_k(o)|,$$

and the time needed to compute $LOF_k(p)$ is propositional to

$$L(k, p) = |N_k p \bigcup \cup_{o \in N_k(p)} (N_k(o) \bigcup \cup_{q \in N_k(o)} N_k(q))|.$$

These two metrics measure the number of objects involved in computing respectively $COF_k(p)$ and $LOF_k(p)$, and are independent of implementation techniques. Figures 16–18 show the results of these two metrics for Wisconsin Breast Cancer, Image and Ionosphere Datasets. In these three figures, the x-axis indicates the index of object p in the dataset, y-axis indicates the values of $C(k, p)$ and $L(k, p)$, and k is set to 5% of the size of the dataset. In Fig. 16, $k = 12$, and $L(12, p)$ is on the average 109 times larger than $C(12, p)$. In Fig. 17, $k = 7$, and $L(7, p)$ is on the average seven times larger than $C(7, p)$. In Fig. 18, $k = 12$, and $L(12, p)$ is on the average 12 times larger than $C(7, p)$. Consistent results are obtained for other values of k. It is clearly that the COF scheme outperforms the LOF scheme based on the given implementation-free metrics. It is interesting to note that both $C(k, p)$ and $L(k, p)$ fluctuate dramatically for Wisconsin Breast Cancer Dataset, while remaining almost stable for the other two datasets. It is also interesting to note that both $C(k, p)$ and $L(k, p)$ (in particular the latter) can be larger than the size of the dataset, implying that numerous overlappings occur among the neighborhoods of p and p's nearest neighbors.

5.7 Time performance and scalability

Here, we report execution time performance and scalability of the COF and LOF schemes. Figure 19 shows execution times of the two schemes on Wisconsin Breast Cancer ($k = 12$), Image ($k = 7$), and Ionosphere Datasets ($k = 12$). As mentioned earlier, the parameter k is set to 5% of the dataset size. Note that both COF and LOF schemes need, for any object in the dataset, to find its k-nearest neighbors. and there are many different methods (see, for example, [29]) with variable time performances depending on the underlying data structures and indexing methods. In order to avoid implementation bias, we use the same method to find k-nearest neighbors in both schemes. It is clear that the COF scheme has better scalability than the LOF scheme with respect to k.

To evaluate the scalability of the two schemes about various parameter k values, we prepared a synthetic, 20-dimensional dataset with 5,000 objects. We ran both schemes for k from 10 to 250 with an increment of 10, and report the results in Fig. 20. The COF scheme has a very steady performance with respect to k, while the LOF has some rapid growth in time as k increases. We also prepared five synthetic, 20-dimensional datasets with respectively 1,000, 2,000, 3,000, 4,000, and 5,000 records. We first ran both schemes on these datasets with $k = 50$, and then repeated the experiments with $k = 100$. Figure 21 shows the scalability of both schemes for $k = 50$, while Fig. 22 shows the scalability for $k = 100$. In both experiments, the COF scheme has a slower growth in time than the LOF scheme as k increases; hence, the COF scheme has better scalability about dataset sizes.

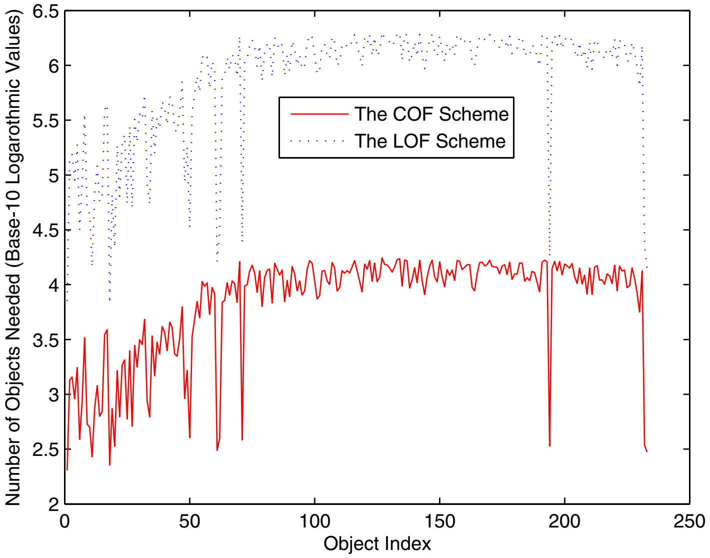


Fig. 16 $C(k, p)$ and $L(k, p)$ of Wisconsin Breast Cancer Dataset

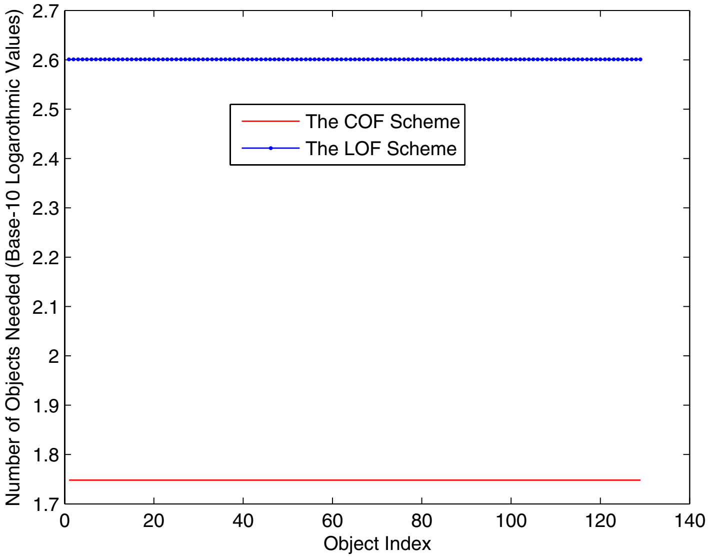


Fig. 17 $C(k, p)$ and $L(k, p)$ of Image Dataset

6 Conclusions

The existing outlier detection schemes are either distance based or density based. The evaluations of their capabilities are mostly ad hoc, and lack theoretical framework for effective analysis and synthesis. We propose a theoretical framework based on which the capabilities of various kinds of schemes can be analyzed. Based on this framework, we study the capabilities of these schemes on

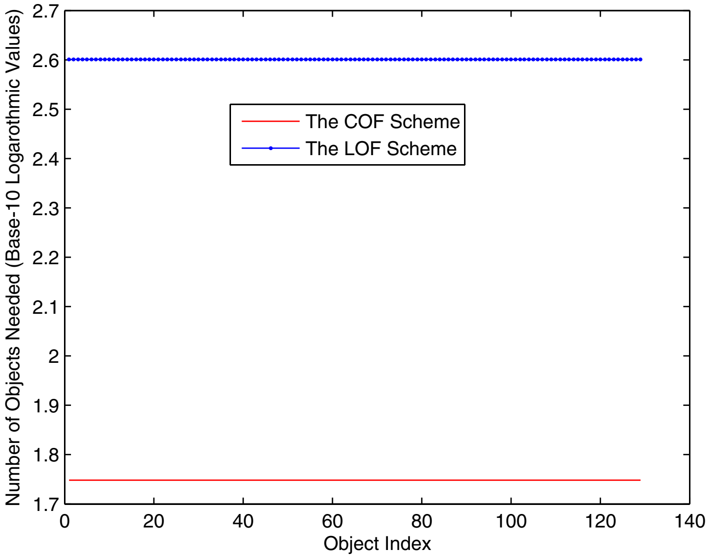


Fig. 18 $C(k, p)$ and $L(k, p)$ of the Ionosphere Dataset

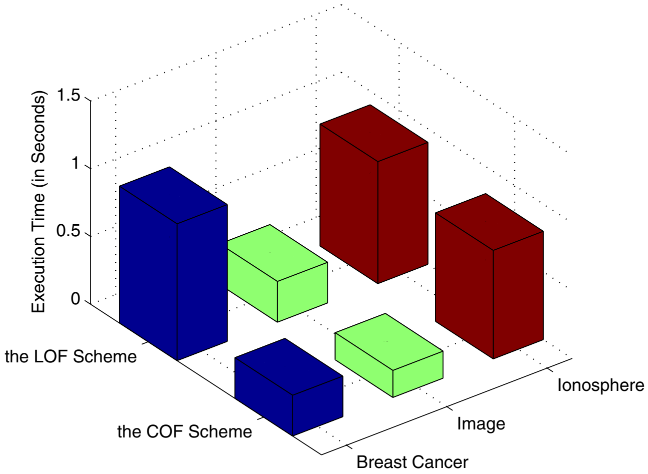


Fig. 19 Time performance comparison

datasets with different characteristics. We find that both the density-based and the connectivity-based schemes are more capable than the distance-based schemes. In comparing the former two schemes, we see that the density-based schemes are effective in an environment where patterns possess sufficiently higher densities than outliers, while the connectivity-based scheme works better for isolated outliers that possess comparable densities with the patterns.

Our study on the capabilities of density-based and connectivity-based schemes shows that we should not view one scheme as being superior to the other in all aspects. To enhance the effectiveness, therefore, one scheme should be used as a compliment, not a replacement, of the other in applications with different

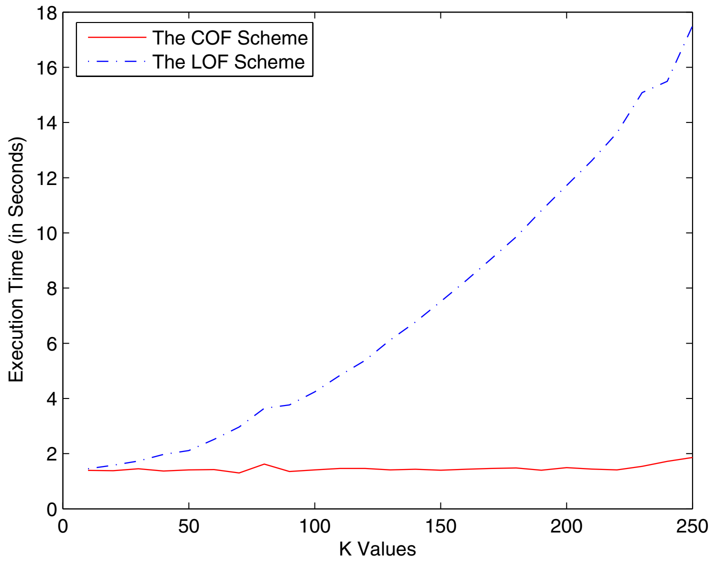


Fig. 20 Scalability of the parameter k

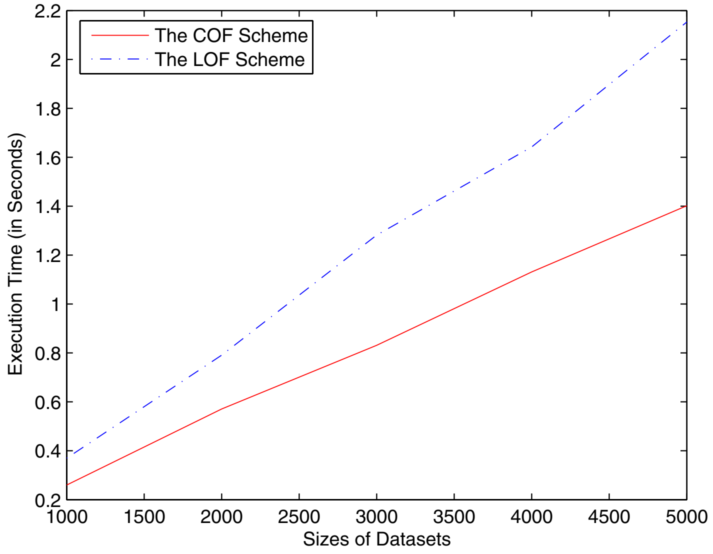


Fig. 21 Scalability of dataset sizes ($k = 50$)

requirements. Thus, it is interesting to develop an effective and efficient method by which the two schemes can be seamlessly integrated. Please note that simply wrapping them into one package does not work. This is because their views toward outlier formulations do not match, and sometime are conflicting. Thus, a naive approach may produce contradictory results, and incurs needlessly high overhead.

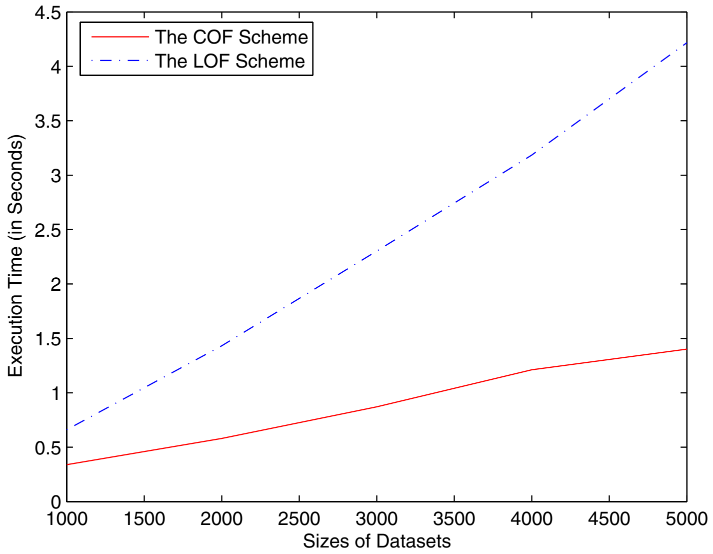


Fig. 22 Scalability of dataset sizes ($k = 100$)

Another interesting issue has to do with the refinement of the framework. Our compatibility theory is based on the precise matching between the outlier formulation schemes and users' intuitions. In reality, however, it is usually difficult to detect all the outliers that fit users' intuitions. Thus, it is probably meaningful to incorporate such a factor as the percentage of the outliers that a scheme can return into the framework. Furthermore, from our experimental results, we observe that for some datasets, for a small number of k values a majority of the covered points can be found. (This is evidenced by Figs. 8, 11, and 12, where the sharp slopes of the curves end at a small value of k, and then followed by more flat slopes that span the rest of the k values.) Thus, by considering only a portion of the outliers, there is a high potential to enhance the performance substantially.

We have shown that there are respective cases where the distance-based, density-based, and connectivity-based outlier detection schemes are not ON-compatible. It is interesting to know whether there exists a case for which all the schemes are not ON-compatible.

Acknowledgements The authors thank the anonymous referees for their valuable comments that help us to revise the early draft of the paper. The last problem in Sect. 6 was suggested to us by one of the referees. The authors also thank Ms. Yin Ling Cheung for her assistance in experiments reported in Sect. 4.3.3. Part of the second author's work was done while he was visiting the Chinese University of Hong Kong.

Appendix A: Proof of Result 3.3

Result 3.3

The $DB(n, v)$-outlier scheme for the dataset in Dataset Example 2 is not O-compatible.

Proof. Let $p \in C_3$ be the point satisfying the condition:

$$dist(o_2, p) = dist_{max}(o_2, C_3).$$

Let $n > 0$ be any value. We consider the following cases for the ranges of radius v.

Case 1: $0 < v \le dist_{max}(o_1, C_1)$. By condition 3, $| N_v(o_1) | < n \Rightarrow | N_v(p) | < n$.

Case 2: $dist_{max}(o_1, C_1) < v \le diam(\{o_2\} \cup C_2)$. We have $| N_v(o_1) | \ge | C_1 |$ and, by condition 5, $| N_v(p) | < | N_{\frac{1}{2}diam(C_3)}(p) |$. Thus, by condition 6, $| N_v(p) | < \frac{3}{4} | C_3 |$. By condition 1, $| N_v(p) | < | C_1 |$. Thus, $| N_v(p) | < | N_v(o_1) |$, meaning $| N_v(o_1) | < n \Rightarrow | N_v(p) | < n$.

Case 3: $diam(\{o_2\} \cup C_2) < v \le diam(\{o_1\} \cup C_1 \cup \{o_2\} \cup C_2)$. We have $| C_2 | \le | N_v(o_2) |$ by the first inequality. On the other hand, by condition 9, $v < dist(\{o_1\} \cup C_1 \cup \{o_2\} \cup C_2, C_3) \le dist(p, \{o_1\} \cup C_1 \cup \{o_2\} \cup C_2)$. Thus, $N_v(p) \cap (\{o_1\} \cup C_1 \cup \{o_2\} \cup C_2) = \phi$, implying $N_v(p) \subseteq C_3$. Thus, $| N_v(p) | \le | C_3 |$. By condition 1 and the inequality derived earlier in this case, $| N_v(p) | < | N_v(o_2) |$. This implies $| N_v(o_2) | < n \Rightarrow | N_v(p) | < n$.

Case 4: $diam(\{o_1\} \cup C_1 \cup \{o_2\} \cup C_2) < v \le dist(o_2, p)$. The first inequality implies $\{o_1\} \cup C_1 \cup \{o_2\} \cup C_2 \subseteq N_v(o_2)$. This means $| \{o_1\} \cup C_1 \cup \{o_2\} \cup C_2 | \le | N_v(o_2) |$. The second inequality and condition imply $v < dist(p, C_2)$, which in turn implies $N_v(p) \cap C_2 = \phi$. Thus, $N_v(p) \subseteq \{o_1\} \cup C_1 \cup \{o_2\} \cup C_3$. This means $| N_v(p) | \le | \{o_1\} \cup C_1 \cup \{o_2\} \cup C_3 |$. Since $C_2 \cap (\{o_1\} \cup C_1 \cup \{o_2\}) = \phi$, $C_3 \cap (\{o_1\} \cup C_1 \cup \{o_2\}) = \phi$, and by condition 1, $| C_3 | < | C_2 |$, we have $| \{o_1\} \cup C_1 \cup \{o_2\} \cup C_3 | < | \{o_1\} \cup C_1 \cup \{o_2\} \cup C_2 |$. Hence, $| N_v(p) | < | N_v(o_2) |$, implying $| N_v(o_2) | < n \Rightarrow | N_v(p) | < n$.

Case 5: $dist(o_2, p) < v$. By the definition of the distance between two set of objects, we have $dist(o_2, p) \ge dist(\{o_1\} \cup C_1 \cup \{o_2\} \cup C_2, C_3)$. Thus, it follows from condition 9 that $diam(\{o_1\} \cup C_1 \cup \{o_2\} \cup C_2) < v$. This, together with the fact that $dist(o_2, p) = dist_{max}(o_2, C_3)$, implies $N_v(o_2)$ contains the entire dataset, except o_2 itself. Thus, $| N_v(o_2) | \ge N_v(p)$, implying $| N_v(o_2) | < n \Rightarrow | N_v(p) | < n$.

Putting all the five cases together, by Theorem 2.2, the $DB(n, v)$-outlier scheme is not O-compatible. □

Appendix B: Proof of Result 3.4

Result 3.4

Assume that the dataset in Dataset Example 2 satisfies two additional conditions:

(a) $\frac{\text{1-distance}_{min(C_i)}}{\text{1-distance}_{max(C_i)}} \ge \frac{4}{5}$, where $1 \le i \le 3$; and
(b) $|N_1(o_j)| = 1$ for $j = 1, 2$.

Then, the LOF scheme is ON-compatible.

Proof. Consider $k = 1$. We first estimate $LOF_1(o_1)$. By conditions 2, 8, and 9, $N_1(o_1) \subseteq C_1$. By condition 2, for any $p \in N_1(o_1)$, $reach\text{-}dist_1(o_1, p) = dist(o_1, p) > dist(o_1, C_1)$. Hence,

$$lrd_1(o_1) = \frac{|N_1(o_1)|}{\sum_{p \in N_1(o_1)} reach\text{-}dist_1(o_1, p)} \le \frac{1}{dist(o_1, C_1)}.$$

For any $p \in N_1(o_1)$ and any $q \in N_1(p)$, by condition 2, $reach\text{-}dist_1(p, q) \le dist(o_1, C_1)/4$. Thus,

$$lrd_1(p) = \frac{|N_1(p)|}{\sum_{q \in N_1(p)} reach\text{-}dist_1(p, q)} \ge \frac{4}{dist(o_1, C_1)}.$$

Therefore,

$$LOF_1(o_1) = \frac{\sum_{p \in N_1(o_1)} lrd_1(p)}{lrd_1(o_1) \cdot |N_1(o_1)|} \ge 4.$$

Similarly, by conditions 4, 8, and 9, we have $LOF_1(o_2) \ge 4$.

For any $p \in C_i$, $i = 1, 2, 3$, and for any $q \in N_1(p)$, by the definition of reachability distance we have $\textit{1-distance}_{min} \le reach\text{-}dist_1(p, q) \le \textit{1-distance}_{max}$. Hence, $\frac{1}{\textit{1-distance}_{max}} \le reach\text{-}dist_1(p, q) \le \frac{1}{\textit{1-distance}_{min}}$. Thus,

$$\frac{1}{\textit{1-distance}_{max}} \le lrd_1(p) = \frac{|N_1(p)|}{\sum_{q \in N_1(p)} reach\text{-}dist_1(p, q)} \le \frac{1}{\textit{1-distance}_{min}}.$$

This implies

$$LOF_1(p) = \frac{\sum_{q \in N_1(p)} lrd_1(q)}{lrd_1(p) \cdot |N_1(p)|} \leq \frac{\textit{1-distance}_{max}}{\textit{1-distance}_{min}} \leq \frac{5}{4} = 1.25.$$

The last part of the previous expression was derived from the given additional condition (a).

Combining the aforementioned analysis together, the LOF scheme detects outliers o_1 and o_2 from the non-outliers with the parameter setting $(k, u) = (1, 2)$. □

Appendix C: Proof of Result 4.1

Result 4.1

Under the same conditions given in Result 3.4, the COF scheme is ON-compatible.

Proof. Consider $k = 1$. We first estimate $COF_1(o_1)$. By conditions 2, 8, and 9, $N_1(o_1) \subseteq C_1$. By condition 2 and the given additional condition (b) as stated in Result 3.4, $N_1(o_1)$ has only one p such that $dist(o_1, p) = dist(o_1, C_1) \geq 4 \times \textit{1-distance}_{max}(C_1)$. Thus, it follows from the definition of average chaining distance that $\textit{ac-dist}_1(o_1) = dist(o_1, p) \geq 4 \times \textit{1-distance}_{max}(C_1)$.

For any $q \in C_1$, by conditions 2, 8, and 9, $N_1(q) \subseteq C_1$. We have by the definition of average chaining distance that $\textit{1-distance}_{min}(C_1) \leq \textit{ac-dist}_1(p) \leq \textit{1-distance}_{max}(C_1)$. Hence, by the additional condition (a) as stated in Result 3.4,

$$COF_1(o_1) = \frac{dist(o_1, p)}{\textit{ac-dist}_1(p)} \geq \frac{4 \times \textit{1-distance}_{max}(C_1)}{\textit{1-distance}_{max}(C_1)} \geq 4,$$

$$COF_1(q) = \frac{|N_1(q)| \cdot \textit{ac-dist}_1(q)}{\sum_{x \in N_1(q)} \textit{ac-dist}_1(x)} \leq \frac{\textit{1-distance}_{max}(C_1)}{\textit{1-distance}_{min}(C_1)} \leq 1.25.$$

Similarly, by conditions 4, 8, and 9, and the additional conditions (a) and (b), we have $COF_1(o_2) \geq 4$ and for any $q \in C_i$, $i = 2, 3$, $COF_1(q) \leq 1.25$.

Combining the aforementioned analysis together, the parameter setting of $(k, u) = (1, 2)$ enables the COF scheme to detect the two outliers o_1 and o_2 from non-outliers. □

References

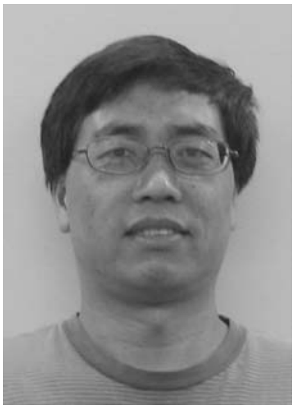
1. Aggarwal C, Yu P (2001) Outlier detection for high dimensional data. In: Aref WG (ed) Proceedings of the 2001 ACM-SIGMOD international conference on management of data, Santa Barbara, CA, USA, May 2001, ACM, pp 37–46
2. Angiulli F, Pizzuti C (2002) Fast outlier detection in high dimensional spaces. In: Tapio Elomaa, Heikki Mannila, Hannu Toivonen (eds) Principles of data mining and knowledge discovery, proceedings of the 6th European PKDD conference, Helsinki, Finland, August 2002. Lecture notes in computer science, vol 2431. Springer, Berlin Heidelberg New York, pp 15–26
3. Arning A, Aggarwal R, Raghavan P (1996) A linear method for deviation detection in large databases. In: Simoudis E, Han J, Fayyad UM (eds) Proceedings of the second international conference on knowledge discovery and data mining (KDD96), Portland, Oregon, USA, 1996. AAAI Press, pp 164–169
4. Baeza-Yates R, Ribeiro-Neto B (1999) Modern information retrieval. Addison Wesley, Reading, MA
5. Bay SD, Schwabacher M (2003) Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: Getoor L, Senator TE, Domingos P, Faloutsos C (eds) Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining, Washington, DC, USA, August 2003. ACM, pp 29–38
6. Barnett V, Lewis T (1994) Outliers in statistical data. Wiley, New York

7. Blake CL, Merz CJ (1998) UCI Repository of machine learning databases. http://www.ics.uci.edu/ mlearn/MLRepository.html. Department of Information and Computer Science, University of California, Irvine, CA
8. Breuning M, Kriegel H, Ng R, Sander J (2000) LOF: identifying density-based local outliers. In: Chen W, Naughton JF, Bernstein PA (eds) Proceedings of the 2000 ACM SIGMOD international conference on management of data, Dallas, Texas, USA, May 2000. ACM, pp 427–438
9. Chen Z, Fu A, Tang J (2003) On complementarity of cluster and outlier detection schemes. In: Kambayashi Y, Mohania MK, WößW (eds) Data warehousing and knowledge discovery, proceedings of the 5th international DaWaK conference, Prague, Czech Republic, September 2003. Lecture notes in computer science, vol 2737. Springer, Berlin Heidelberg New York, pp 234–243
10. Chen Z, Tang J, Fu A (2003) Modeling and efficient mining of intentional knowledge of outliers. In: Proceedings of the 7th international database engineering and applications symposium (IDEAS03), Hong Kong, China, July 2003. IEEE Computer Society, pp 44–53
11. Chen Z, Meng X, Fowler R, Zhu B (2001) FEATURES: real-time adaptive feature and document learning for web search. J Am Soc Inform Sci Technol 52(8):655–665
12. Cormen T, Leiserson C, Rivest R, Stein C (2002) Introduction to algorithms, 2nd edn. McGraw-Hill, New York
13. DuMouchel W, Schonlau M (1998) A fast computer intrusion detection algorithm based on hypothesis testing of command transition probabilities. In: Agrawal R, Stolorz PE, Piatetsky-Shapiro G (eds) Proceedings of the fourth international conference on knowledge discovery and data mining (KDD98), New York City, New York, USA, August 1998. AAAI Press, pp 189–193
14. Ester M, Kriegel H, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis E, Han J, Fayyad UM (eds) Proceedings of the second international conference on knowledge discovery and data mining (KDD96), 1996. AAAI Press, pp 226–231
15. Fawcett T, Provost F (1997) Adaptive fraud detection. Data Min Knowledge Discov J 1(3):291–316
16. Guha S, Rastogi R, Shim K (1998) Cure: an efficient clustering algorithm for large databases. In: Haas LM, Tiwary A (eds) Proceedings ACM SIGMOD international conference on management of data, Seattle, Washington, USA, June 1998. ACM Press, pp 73–84
17. Harkins S, He H, Williams CJ, Baster RA (2002) Outlier detection using replicator neural networks. In: Kambayashi Y, Winiwarter W, Arikawa M (eds) Data warehousing and knowledge discovery, proceedings of the 4th international DaWaK conference, Aix-en-Provence, France, September 2002. Lecture notes in computer science, vol 2454. Springer, Berlin Heidelberg New York, pp 170–180
18. Hawkins D (1980) Identification of outliers. Chapman & Hall, London
19. He Z, Xu X, Deng S (2003) Discovering cluster-based local outliers. Pattern Recog Lett 24:1641–1650
20. Hu T, Sung SY (2003) Detecting pattern-based outliers. Pattern Recog Lett 24:3509–3068
21. Jin W, Tung A, Han J (2001) Mining top-*n* local outliers in large databases. In: Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, USA, August 2001. ACM, pp 293–298
22. Keogh E, Kasetty S (2003) On the need for time series data mining benchmarks: a survey and empirical demonstration. Data Min Knowledge Discov 7(4):349–371
23. Knorr E, Ng R (1998) Algorithms for mining distance-based outliers in large datasets. In: Gupta A, Shmueli O, Widom J (eds) Proceedings of 24rd international conference on very large databases, New York City, New York, USA, August 1998. Morgan Kaufmann, pp 392–403
24. Knorr E, Ng R (1999) Finding intentional knowledge of distance-based outliers. In: Atkinson MP, Orlowska ME, Valduriez P, Zdonik SB, Brodie ML (eds) Proceedings of the 25th international conference on very large databases, Edinburgh, Scotland, UK, September 1999. Morgan Kaufmann, pp 211–222
25. Lazarevic A, Ertoz L, Ozgur A, Srivastava J, Kumar V (2003) A comparative study of anomaly detection schemes in network intrusion detection. In: Barbar D, Kamath C (eds)

Proceedings of the third SIAM international conference on data mining, San Francisco, CA, USA, May 2003. SIAM

26. Meng X, Chen Z (2004) On user-oriented measurements of effectiveness of web information retrieval systems. In: Arabnia HR, Droegehorn O (eds) Proceedings of the international conference on internet computing, Las Vegas, Nevada, USA, June 2004, vol 1. CSREA Press, pp 527–533
27. Ng R, Han J (1994) Efficient and effective clustering methods for spatial data mining. In: Bocca JB, Jarke M, Zaniolo C (eds) Proceedings of the 20th international conference on very large databases, Santiago de Chile, Chile, September 1994. Morgan Kaufmann, pp 144–155
28. Ramaswamy S, Rastogi R, Kyuseok S (2000) Efficient algorithms for mining outliers from large data sets. In: Chen W, Naughton JF, Bernstein PA (eds) Proceedings of the 2000 ACM SIGMOD international conference on management of data, Dallas, Texas, USA, May 2000. ACM, pp 427–438
29. Roussopoulos N, Kelley S, Vincent F (1995) Nearest neighbor queries. In: Carey MJ, Schneider DA (eds) Proceedings of the 1995 ACM SIGMOD international conference on management of data, San Jose, California, USA, May 1995. ACM, pp 71–79
30. Salton G (1989) Automated text processing: the transformation, analysis, and retrieval of information by computer. Addison Wesley, Reading, MA
31. Sheikholeslami G, Chatterjee S, Zhang A (1998) WaveCluster: a multi-resolution clustering approach for very large spatial databases. In: Gupta A, Shmueli O, Widom J (eds) Proceedings of 24rd international conference on very large databases, New York City, New York, USA, August 1998. Morgan Kaufmann, pp 428–439
32. Stolfo S, Fan W, Lee W, Prodromidis A, Chan P (2000) Cost-based modeling for fraud and intrusion detection: results from the JAM Project. In: Proceedings of DARPA information survivability conference and exposition, vol 2, pp 1130–1144
33. Tang J, Chen Z, Fu A, Cheung D (2002) Enhancing effectiveness of outlier detections for low density patterns. In: Cheng M-S, Yu PS, Liu B (eds) Advances in knowledge discovery and data mining, proceedings of the 6th Pacific-Asia PAKDD conference, Taipei, Taiwan, May 2002. Lecture notes in computer science, vol 2336. Springer, Berlin Heidelberg New York, pp 535–548
34. Zhang T, Ramakrishnan R, Linvy M (1996) BIRCH: an efficient data clustering method for very large databases. In: Jagadish HV, Mumick IS (eds) Proceedings of the 1996 ACM SIGMOD international conference on management of data, Montreal, Quebec, Canada, June 1996. ACM, pp 103–114

Jian Tang received an MS degree from the University of Iowa in 1983, and PhD from the Pennsylvania State University in 1988, both from the Department of Computer Science. He joined the Department of Computer Science, Memorial University of Newfoundland, Canada, in 1988, where he is currently a professor. He has visited a number of research institutions to conduct researches ranging over a variety of topics relating to theories and practices for database management and systems. His current research interests include data mining, e-commerce, XML and bioinformatics.



Zhixiang Chen is an associate professor in the Computer Science Department, University of Texas-Pan American. He received his PhD in computer science from Boston University in January 1996, BS and MS degrees in software engineering from Huazhong University of Science and Technology. He also studied at the University of Illinois at Chicago. He taught at Southwest State University from Fall 1995 to September 1997, and Huazhong University of Science and Technology from 1982 to 1990. His research interests include computational learning theory, algorithms and complexity, intelligent Web search, informational retrieval, and data mining.

Ada Waichee Fu received her BSc degree in computer science in the Chinese University of Hong Kong in 1983, and both MSc and PhD degrees in computer science in Simon Fraser University of Canada in 1986, 1990, respectively; worked at Bell Northern Research in Ottawa, Canada, from 1989 to 1993 on a wide-area distributed database project; joined the Chinese University of Hong Kong in 1993. Her research interests are XML data, time series databases, data mining, content-based retrieval in multimedia databases, parallel, and distributed systems.

David Wai-lok Cheung received the MSc and PhD degrees in computer science from Simon Fraser University, Canada, in 1985 and 1989, respectively. He also received the BSc degree in mathematics from the Chinese University of Hong Kong. From 1989 to 1993, he was a member of Scientific Staff at Bell Northern Research, Canada. Since 1994, he has been a faculty member of the Department of Computer Science in the University of Hong Kong. He is also the Director of the Center for E-Commerce Infrastructure Development. His research interests include data mining, data warehouse, XML technology for e-commerce and bioinformatics. Dr. Cheung was the Program Committee Chairman of the Fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2001), Program Co-Chair of the Ninth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2005). Dr. Cheung is a member of the ACM and the IEEE Computer Society.