# Enhancing Effectiveness of Outlier Detections for Low Density Patterns

Jian Tang[+][*], Zhixiang Chen[†], Ada Wai-chee Fu[+], David W. Cheung[‡]

[+]Department of Computer Science and Engineering
Chinese University of Hong Kong
Shatin, Hong Kong

[†]Department of Computer Science
University of Texas at Pan-America
Texas, U.S.A

[‡]Department of Computer Science and Information Systems
University of Hong Kong
Pokfulam, Hong Kong

**Abstract**

Outlier detection is concerned with discovering exceptional behaviors of objects in data sets. It is becoming a growingly useful tool in applications such as credit card fraud detection, discovering criminal behaviors in e-commerce, identifying computer intrusion, detecting health problems, etc. In this paper, we introduce a connectivity-based outlier factor (COF) scheme that improves the effectiveness of an existing local outlier factor (LOF) scheme when a pattern itself has similar neighbourhood density as an outlier. We give theoretical and empirical analysis to demonstrate the improvement in effectiveness and the capability of the COF scheme in comparison with the LOF scheme.

## 1 Introduction

Outlier detection is an important branch in the area of data mining. It is concerned with discovering the exceptional behaviors of certain objects. Revealing these behaviors is important since it signifies that something out of ordinary has happened and shall deserve people's attention. In many cases, such exceptional behaviors will cause damage to users and must be stopped. In other cases, there can be

[*] On leave from Memorial University of Newfoundland, Canada.

"good" outliers which can help users to make profits. Therefore, in some sense detecting outliers is at least as significant as discovering general patterns. Outlier detection is becoming a growingly useful tool in applications to which people have already paid attention, such as credit card fraud detection, calling card fraud detection, discovering criminal behaviors in e-commerce, discovering computer intrusion, and etc. [4, 6].

Hawkins [7] 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 under the general topic of clustering [5, 10, 13, 14, 16]. These algorithms can also generate outliers as by-products. However, the outliers discovered this way are highly dependent on the clustering algorithms used and hence subject to the clusters generated. Most methods in the early work that detects outliers independently have been developed in the field of statistics [2]. These methods normally assume that the distribution of a data set is known in advance and try to detect outliers by examining the deviations of individual data objects based on such a distribution. In reality, however, a priori knowledge about the distribution of a data set is not always obtainable. Besides, these methods do not scale well for even modest number of dimensions as the size of a data set increases.

More recently, researchers proposed distance based schemes, which distinguish objects that are likely to be outliers from those that are not based on the number of objects in the neighborhood of an object [8, 9, 11]. These schemes do not make any assumptions about the distribution of a data set. Furthermore, since the counting process is restricted only to the neighborhood of an object, the scalability of these methods is better than that of their predecessors. As a result, distance based schemes are more appropriate for detecting outliers in large data sets without assuming a priori knowledge about their distributions.

Knorr and Ng [8] propose a distance based scheme, called $DB(n,q)$-outlier. In this scheme, if the neighborhood with the radius of $q$ (called "$q$-neighborhood") of an object contains less than $n$ objects, then it is called an outlier with respect to $n$ and $q$, otherwise it is not. The advantage of this scheme is its simplicity while capturing the basic intuition given in Hawkins' definition. Its weakness is that it cannot deal with data sets that contain patterns with diverse characteristics. The scheme proposed by Ramaswamy, et al. [11], called $(t,k)$-nearest neighbor scheme, considers for each point its $k$-distance, i.e., the distance to its $k$th nearest neighbor(s). It ranks the top $t$ objects with the maximum $k$-distances as the outliers. If there are multiple objects with the same $k$-distance ranked as the top $k$, they are all considered as outliers. Therefore, the number of outliers returned may be greater than $t$. This scheme is actually a special case of $DB(n,q)$-outlier. Thus it shares the same weakness as $DB(n,q)$-outlier has.

Recently, Breuning, et al, [3] proposed a density based formulation scheme as follows.

Let $p, o \in \mathcal{D}$ and $k$ be a positive integer. Let *$k$-distance*$(o)$ be the distance from $o$ to its $k$-th nearest neighbor, where if two neighbors are at same distance from $o$, the ordering of "nearest" for them is arbitrary. The $k$-distance neighbourhood of an object $p$ is denoted by $N_{k\text{-}distance(p)}(p)$ and is the set of objects whose distance from $p$ is not greater than $k$-distance.

The reachability distance of $p$ with respect to $o$ for $k$ is defined as:

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

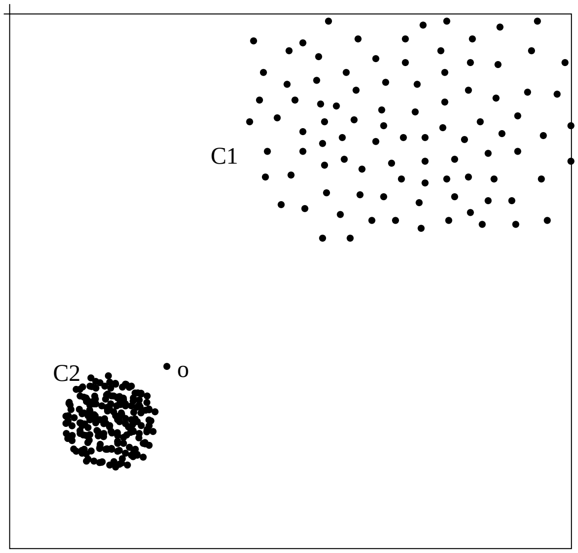

Figure 1: A Data Set showing the strength of LOF

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

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

That is, $lrd_k(p)$ is the inverse of the average reachability distance from $p$ to the objects in its $k$-distance neighborhood. For simplicity, we shall refer to the local reachability density of a point $p$ as the *density* of $p$. The local outlier factor of $p$ is defined as

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

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 [3], 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 $p$ has a higher degree of being an outlier.

Note that the density based scheme does not explicitly categorize the objects into either outliers or non-outliers. (If desired, a user can do so by choosing a threshold value to separate the LOF values of the two classes.) It uses the LOF to measure how strong an object can be an outlier. Since the LOF value of an object is obtained by comparing its density with those in its neighborhood, it has stronger modeling capability than a distance based scheme, which is based only on the density of the object itself.

In [3] the authors give an example, which we have duplicated in Figure 1. The data set contains an outlier $o$, and $C1$ and $C2$ are two clusters with very different densities. The authors show that the DB(n,q)-outlier method cannot distinguish $o$ from the rest of the data set no matter what values the parameters take. However, LOF method can handle it successfully.

The weakness of the density based scheme is that it considers solely the difference between the density of an object and those of its neighbors (we shall show such an example in the next section). Thus its effectiveness will diminish if the density of an outlier is close to those of its neighbors. In this paper, we introduce a connectivity-based outlier factor (COF) scheme for outlier formulation, We use empirical analysis to demonstrate the improvement in effectiveness and the capability of the COF scheme over the LOF scheme.

The rest of this paper is organized as follows. In Section 2 we propose a definition of ON-compatibility for the goodness of an outlier detection method. In Section 3, we revisit the density based schemes. In Section 4, we introduce the connectivity-based scheme. In Section 5, we compare the connectivity-based and density-based schemes using the experimental data. In Section 6, we discuss the complexity involved in calculating the COF. Finally in Section 7 we conclude the paper by summarizing the main results.

## 2 ON-Compatibility

All the previously introduced methods utilize an outlier measure function and a number of parameter settings. For the DB(n,q)-outlier method, the measurement for an object is the number of objects within a radius of $q$, and the outlier decision is based on whether the number is less than $n$. The parameters are $q$ and $n$. For the $(t, k)$-nearest neighbors, the measurement for an object is the distance to the $k$-th nearest neighbor, and decision is based on whether the distance is among the top $t$ such values. The LOF measurement is a little more complex and also utilize parameters of $k$ in $k$-distance. In all the above, the measurement is typically for a data point $p$ and its value depends on some set of parameters $S$, hence it can be denoted by $f(p, S)$. We would like $f(p, S)$ to be large when $p$ is an outlier, and small if it is not. Therefore for the DB(n,q)-outlier method, $f(p, S)$ can be set as $n$ divided by the number of objects within a radius of $q$.

In [15] we have developed a stack of measurements to evaluate the capabilities of outlier measure functions for different formulation schemes, with the increasingly relaxed requirements down the stack. Due to the space limitation we introduce only the measurement on the top of the stack, termed *ON-Compatibility* (ON stands for Outliers and Non-outliers). We will use ON-compatibility to evaluate the effectiveness of the density-based scheme and the connectivity based scheme.

**Definition 1** *The outlier measure function $f(p, S)$ is ON-compatible with a given set of data with outliers and non-outliers (we call this an interpretation $I$), if there exists a parameter setting $S$, and a value $u$, such that*

**(1).** *for each outlier $o$, the measure $f(o, S)$ has a value above $u$.*

**(2).** *for each of the non-outliers $n$, the measure $f(n, S)$ has a value below $u$.*

*The value $u$ is called cut-off value.*

ON-compatibility indicates the capability of an outlier measure function to use a single parameter setting to detect all outliers for a given interpretation. It is most desirable, but not often attainable. An interpretation is given by a set of data $\mathcal{D}$ together with its partitioning into the set of outliers $\mathcal{D}_o$ and the

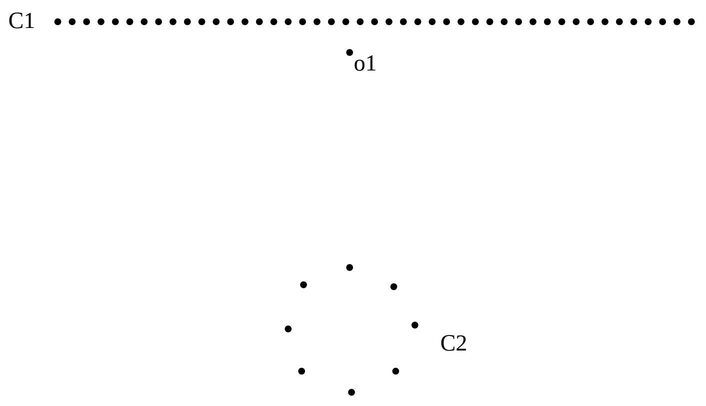

Figure 2: Failure of Outliers detection for LOF

set of non-outliers $\mathcal{D}_n$. With the following theorem, we introduce a method to determine when a function is not compatible with some given interpretation.

**Theorem 1** *Let $f(p, S)$ be an outlier measure function and $I$ be an interpretation: $\mathcal{D} = \mathcal{D}_o \cup \mathcal{D}_n$, then the following holds for $f(p, S)$: It is not ON-compatible with $I$ if for any setting $S$, there exist an object $a \in \mathcal{D}_n$, and an object $o \in \mathcal{D}_o$, such that $f(o, S) \leq f(a, S)$.*

## 3 Density Based Schemes Revisited

We have seen from the previous sections, the density based scheme, such as LOF, is more powerful than the previous methods. However, one weakness of the density based scheme is that it may rule out outliers close to some non-outliers pattern that has low density. 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 a pattern it deviates from. A typical example is shown in Figure 2.

In this figure, the pattern, $C_1$, is a straight line, which is of low density in a two dimensional space. Point $o1$ and the points in $C_2$ are outliers. Since $o1$ shifts away from a low density pattern, the density based outlier measure function will not be 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 must be identified using a value for $k$ larger than its cardinality. In the following, we assume some specific values for the variables of the data set.

**EXAMPLE 1** *$C_1$ contains 91 points, with distance one between adjacent points. o1 is a point closest to the middle of $C_1$. The circle $C_2$ with radius of one contains eight points evenly positioned on its circumference. The center of $C_2$, o1 and the middle point in $C_1$ are on a line. (Note that the circle for $C_2$ has been much enlarged in Figure 2.) The distance from o1 to $C_2$ is 1000. The distance from o1 to the middle point of $C_1$ is two. Let I be the interpretation: $\mathcal{D} = \mathcal{D}_o \cup \mathcal{D}_n$ where $\mathcal{D}_o = \{o1\} \cup C_2$ and $\mathcal{D}_n = C_1$.*

For the values given in the above example, we can formally prove

**ASSERTION 2** *The LOF outlier measure is not ON-compatible for I in the data set of Example 1.*

The proof of the above assertion is based on Theorem 1, and is omitted here. In a later section, we will show the ineffectiveness of LOF in handling a similar case. In the next section, we will introduce a scheme that can handle low density patterns such as the line of points in Figure 2, while at the same time does not compromise detecting a group of stay-together outliers like those in Figure 2.

# 4 Connectivity-Based Outliers

Our solution is based on the idea of differentiating *"low density"* from *"isolativity"*. While low density normally refers to the fact that the number of objects 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 Figure 2 point $o1$ is isolated, while any point $p$ in $C_1$ is not. But both of them are of roughly equal 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. An outlier indicator should take into consideration of both cases.

We observe that patterns that possess low densities usually exhibit low dimensional structures. For example, a pattern shown in Figure 2 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.

We first introduce some notations and then formulate our connectivity-based outlier scheme.

**Definition 2** *Let $P, Q \subseteq \mathcal{D}$, $P \cap Q = \emptyset$ and $P, Q \neq \emptyset$. We define $dist(P, Q) = min\{dist(x, y) : x \in P \;\&\; y \in Q\}$, and call $dist(P, Q)$ the distance between P and Q. For any given $q \in Q$, we say that q is the nearest neighbor of P in Q if there is a $p \in P$ such that $dist(p, q) = dist(P, Q)$.*

In the following definitions, let $G = \{p_1, p_2, \ldots, p_r\}$ be a subset of $\mathcal{D}$.

**Definition 3** *A set based nearest path, or SBN-path, from $p_1$ on G is a sequence $\langle p_1, p_2, \ldots, p_r \rangle$ such that for all $1 \leq i \leq r-1, p_{i+1}$ is the 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 pre-defined order among its neighbors to break tie. Thus an *SBN*-path is uniquely determined. An *SBN*-path indicates the order in which the nearest objects are presented.

**Definition 4** *Let $s = \langle p_1, p_2, \ldots, p_r \rangle$ be an SBN-path. 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 uniquely determined, we should break tie by a pre-defined order. Thus the *SBN*-trail is unique for any *SBN*-path.

**Definition 5** *Let $s = \langle p_1, p_2, \ldots, p_r \rangle$ be an SBN-path from $p_1$ and $e = \langle e_1, \ldots, e_{r-1} \rangle$ be the SBN-trail with respect to s. The average chaining distance from $p_1$ to $G - \{p_1\}$, denoted by $ac\text{-}dist_G(p_1)$, is defined as*

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

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

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

and viewing the fraction following the summation sign as the weight, the average chaining distance 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. Thus if the edges close to $p_1$ are substantially larger than those away from $p_1$, then they contribute more in the $ac\text{-}dist_G(p_1)$. This is consistent with our motivation. In the special case where $dist(e_i)$ is the same for all $e_i$, we have $ac\text{-}dist_G(p_1) = dist(e_i)$.

**Definition 6** *Let $p \in \mathcal{D}$ and $k$ be a positive integer. The connectivity-based outlier factor (COF) at p with respect to its k-neighborhood is defined as*

$$COF_k(p) \quad = \quad \frac{|N_k(p)| \cdot ac\text{-}dist_{N_k(p)}(p)}{\sum_{o \in N_k(p)} ac\text{-}dist_{N_k(o)}(o)}.$$

The connectivity-based outlier factor at $p$ is the ratio of the average chaining distance from $p$ to $N_k(p)$ 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.

Consider the data set in Figure 3. 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.

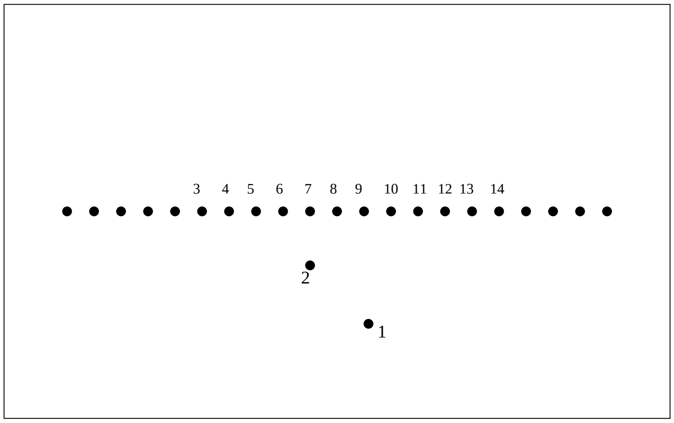

Figure 3: Calculating COF

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

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

The *SBN*-trail for $s_1$ is

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

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

For point 2: $N_k(2) = \{7, 6, 8, 5, 9, 4, 10, 3, 11, 1\}$. The *SBN*-path from 2 on $N_k(2) \cup \{2\}$ 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 $ac\text{-}dist_{N_k(2) \cup \{2\}}(2) = 1.46$.

For point 7: $N_k(7) = \{6, 8, 5, 9, 4, 10, 2, 3, 11, 12\}$. The *SBN*-path from 7 on $N_k(7) \cup \{7\}$ 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 $ac\text{-}dist_{N_k(7) \cup \{7\}}(7) = 0.98$.

The average chaining distances for the other points on the line can be calculated similarly. The above 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 tend to be larger values 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 sums 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 $k$-distance 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 $k$-distance 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 above example, we have the following:

$$COF_k(1) = 2.1,\ \ COF_k(2) = 1.35\ \ and\ \ COF_k(7) = 0.96.$$

## 5 Comparison of COF and LOF

The connectivity-based scheme has some important properties like the density-based scheme, for example the COF value for an object deep inside a cluster being close to 1. For instance, for an object $p$ and a cluster $C$ such that $N_k(p) \subseteq C$, we can prove that $\frac{1}{1+\epsilon} \leq \mathrm{COF}_k(p) \leq 1+\epsilon$ where $\epsilon$ is a small value. We follow the approaches in [3] to show those similar bounds for *COF*. But first we give the following definition.

**Definition 7** *Given any object $p \in \mathcal{D}$, let $s = \{e_1, \ldots, e_{r-1}\}$ be the SBN-trail with respect to the SBN-path from $p$ on $N_k(p)$. We define*

$$\begin{aligned} path\text{-}min(p) &= min\{dist(e_1), \ldots, dist(e_{r-1})\}, \\ path\text{-}max(p) &= max\{dist(e_1), \ldots, dist(e_{r-1})\}. \end{aligned}$$

**Theorem 3** *Given any set $C \subseteq \mathcal{D}$, let $path\text{-}min = min\{\ path\text{-}min(p) : p \in C\}$ and $path\text{-}max = max\{\ path\text{-}max(p) : p \in C\}$. Let $\epsilon = \frac{path\text{-}max}{path\text{-}min} - 1$, then for every object $p \in C$ such that*

*(i) $N_k(p) \subseteq C$, and*

*(ii) for every $q \in N_k(p)$, $N_k(q) \subseteq C$, we have*

$$\frac{1}{1+\epsilon} \leq COF_k(p) \leq 1+\epsilon.$$

The above theorem, together with the illustration in the previous section, indicate that the connectivity based scheme has the similar power to that of the density based scheme in detecting outliers which deviate from high density patterns. On the other hand, recall that the motivation for introducing the connectivity based scheme is to handle outliers deviating from low density patterns. We showed previously in Figure 3 an example of a low density pattern. We now present a similar example in Figure 4. (We have assumed special geometric shapes and distances for the data set. These are used only for convenience of plotting the results.)

**EXAMPLE 2** *In Figure 4, $C_2$ contains 8 points lying on the circle with its center at $(1, 0)$ and a radius of 1. Distances between any two adjacent points on the circle are the same. $C_1$ contains 91 points lying*

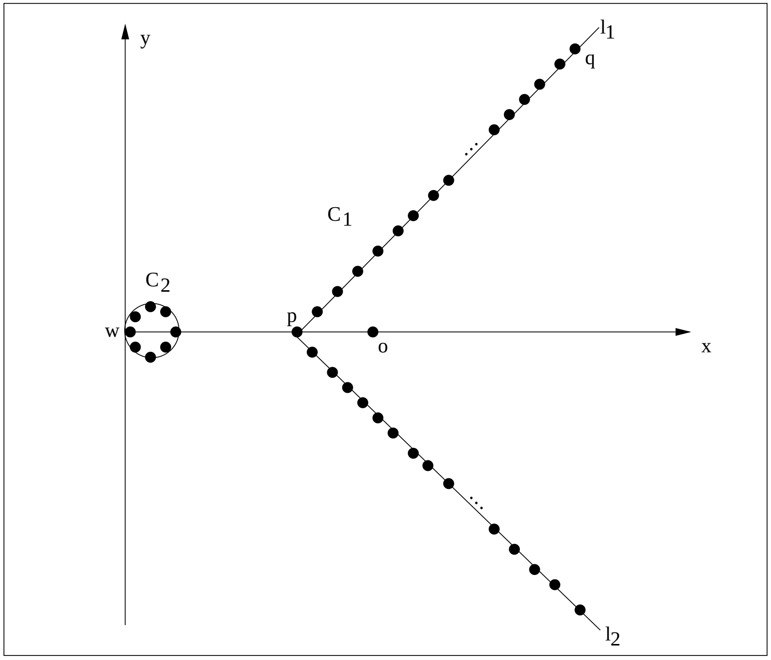

Figure 4: Data Set for Comparison

*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 $C_2$ can be considered as outliers while others are non-outliers. Thus, we have an interpretation $I : \mathcal{D} = \mathcal{D}_o \cup \mathcal{D}_n$, where $\mathcal{D}_o = \{o\} \cup C_2$ and $\mathcal{D}_n = C_1$.*

Our result is contained in the following assertion.

**ASSERTION 4** *The LOF outlier measure is not ON-compatible for $I$ in the above example.*

We support the above assertion by the experimental data. We choose 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 Figure 4. Note that $q$ is the end point of $C_1$ on line $l_1$. Note also that the total number of points in the data set is 100. We have calculated the LOF values for all those four points for $k = 1, 2, \ldots, 99$. The calculation has been done by a C++ program with a precision of 10 decimal digits. The computing environment is a Dell Precision 530 running SuSE Linux 7.2 with two 1.5 GHz Pentium Xeon Processors, 2 GB RAM and 40 GB hard disk. The LOF values for the four points are reported in Figure 5. For $1 \leq k \leq 7$, we have $\text{LOF}_k(q) > \text{LOF}_k(w)$. For $8 \leq k \leq 98$, we have $\text{LOF}_k(q) > \text{LOF}_k(o)$. For $k = 99$, we have $\text{LOF}_k(p) = 1.0013753983 > \text{LOF}_k(w) = 0.9992365171$. Because $p$ and $q$ are non-outliers and $o$ and $w$ are outliers, it follows from the definition of the outlier measure function of the LOF scheme that this measure is not ON-compatible with the interpretation given above. On the other hand, we have

**ASSERTION 5** *For the data set in Example 2 shown in Figure 4, COF is ON-compatible with $\mathcal{I}$*

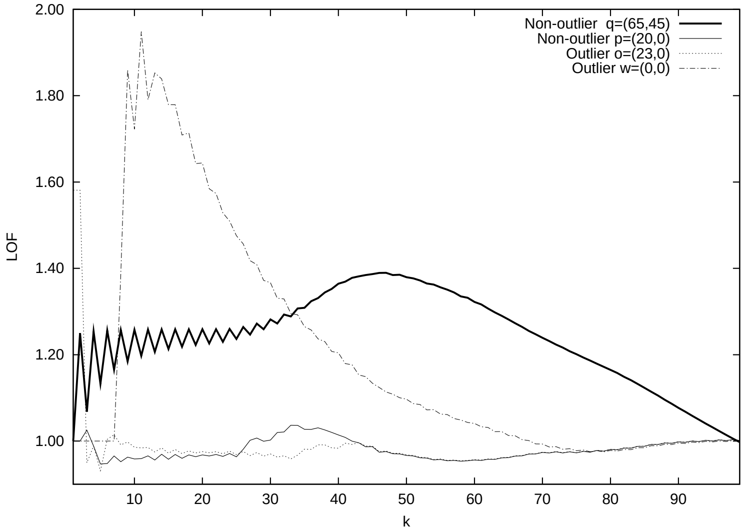

Figure 5: LOF Values for Four Points on Different Settings

This assertion is supported by the experimental result shown in Figure 6. We choose $k = 13$ and calculate COF values for all points in the data set. All calculations were done by a C++ program with a precision of 10 decimal digits. The computing environment is the same as that for Assertion 4. All the 8 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 $\ell_1$ and $\ell_2$ 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, COF is ON-compatible with $\mathcal{I}$ as defined in Assertion 4.

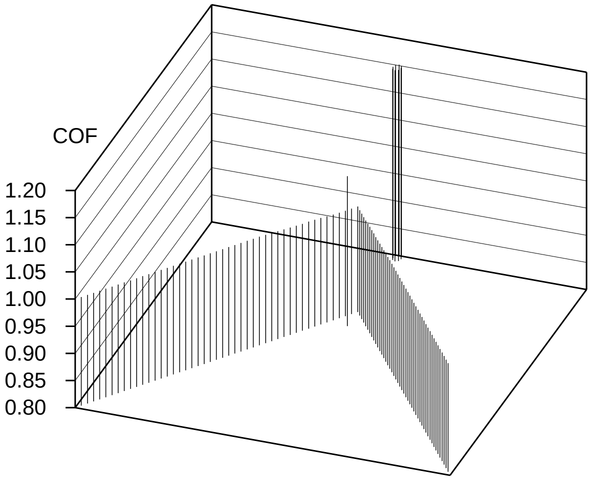

Figure 6: COF Values of All Points When $k$=13

# 6 Time Complexity

Suppose that the database $\mathcal{D}$ has $n$ objects. As the LOF algorithm in [3], the COF algorithm can be divided into two steps. In the first step, the COF method finds the $k$-nearest neighborhoods and the SBN-trails. Precisely, the COF algorithm finds, for any object $p \in \mathcal{D}$, the $k$-nearest neighborhoods together with their distances to $p$, and the SBN-trail together with the costs of the objects in the trail. The result of this step is materialization database $\mathcal{M}$ of size $2 \times n \times k$. Similar to the LOF algorithm, the size of this intermediate database is independent of the dimensionality of the original data. The running time of this step is $O(n \times (\text{time for a } k\text{-nn query}))$. Depending on the particular implementations of the *k-nn* query, its time complexity can vary from constant time for low-dimensional data, to $\log n$ for medium-dimensional data, and to $n$ for extremely high-dimensional data. Hence, the time complexity of the COF algorithm in the first step can vary from $O(n)$ for low-dimensional data, to $O(n \log n)$ for medium dimensional data, and to $O(n^2)$ for extremely high-dimensional data.

In the second step, the COF method computes the COF values with the help of the materialization database $\mathcal{M}$. The original database is not needed in this step. The COF algorithm scans the database $\mathcal{M}$ twice. In the first scan, the algorithm finds the average chaining distance for every objects. In the second scan, the COF values of every objects are computed and written to a file. The time complexity of this step is $O(n)$. Notice that the time complexity for computing COF is similar to that for LOF.

# 7 Conclusions

While the field of data mining has been studied extensively, most work has concentrated on the discovery of patterns. Outlier detection as a branch of data mining has many important applications, and deserves more attention from data mining community. The existing work on outlier detection is either distance based or density based. In essence, these schemes all assume patterns have high (relative) densities. Therefore they do not work adequately where the patterns are of low densities. We propose a scheme that overcomes this weakness. Our scheme separates the notion of density from that of isolation. It can therefore detect outliers independently of the densities of the patterns from which they deviate. To measure the capabilities of outlier detection schemes, we introduce a notion of ON-compatibility. We show that while our scheme preserves the same nice properties as those of the density based method, it can achieve better results for data sets with connectivity characteristics in the data patterns.

ACKNOWLEDGMENT. The authors thank Ms. Yin Ling Cheung for preparing several *gnuplot* templates for plotting

# References

[1] A. Arning, R. Agrawal, P. Raghavan: "A Linear Method for Deviation detection in Large Databases", Proc. of 2nd Intl. Conf. On Knowledge Discovery and Data Mining, 1996, pp 164 - 169.

[2] V. Barnett, T. Lewis: "Outliers in Statistical Data", John Wiley, 1994.

[3] M. Breuning, Hans-Peter Kriegel, R. Ng, J. Sander: "LOF: Identifying density based Local Outliers", Proc. of the ACM SIGMOD Conf. On Management of Data, 2000.

[4] W. DuMouchel, M. Schonlau: "A Fast Computer Intrusion Detection Algorithm based on Hypothesis Testing of Command Transition Probabilities", Proc.of 4th Intl. Conf. On Knowledge Discovery and Data Mining, 1998, pp. 189 - 193.

[5] M. Ester, H. Kriegel, J. Sander, X. Xu: "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Proc. of 2nd Intl. Conf. On Knowledge Discovery and Data Mining, 1996, pp 226 - 231.

[6] T. Fawcett, F. Provost: "Adaptive Fraud Detection", Data Mining and Knowledge Discovery Journal, Kluwer Academic Publishers, Vol. 1, No. 3, 1997, pp 291 - 316.

[7] D. Hawkins: "Identification of Outliers", Chapman and Hall, London, 1980.

[8] E. Knorr, R. Ng: "Algorithms for Mining Distance based Outliers in Large Datasets", Proc. of 24th Intl. Conf. On Very Large Data Bases, 1998, pp 392 - 403.

[9] E. Knorr, R. Ng: "Finding Intensional Knowledge of Distance-based Outliers", Proc. of 25th Intl. Conf. On Very Large Data Bases, 1999, pp 211 - 222.

[10] R. Ng, J. Han: "Efficient and Effective Clustering Methods for Spatial Data Mining", Proc. of 20th Intl. Conf. On Very Large Data Bases, 1994, pp 144 - 155.

[11] S. Ramaswamy, R. Rastogi, S. Kyuseok: "Efficient Algorithms for Mining Outliers from Large Data Sets", Proc. of ACM SIGMOD Intl. Conf. On Management of Data, 2000, pp 427 - 438.

[12] N. Roussopoulos, S. Kelley, F. Vincent, "Nearest Neighbor Queries", Proc. of ACM SIGMOD Intl. Conf. On Management of Data, 1995, pp 71 - 79.

[13] G. Sheikholeslami, S. Chatterjee, A. Zhang: "WaveCluster: A multi-Resolution Clustering Approach for Very Large Spatial Databases", Proc. of 24th Intl. Conf. On Very Large Data Bases, 1998, pp 428 - 439.

[14] S. Guha, R. Rastogi, K. Shim: "Cure: An Efficient Clustering Algorithm for Large Databases", In Proc. of the ACM SIGMOD Conf. On Management of Data, 1998, pp 73-84.

[15] J. Tang, Z. Chen, A. Fu and D. Cheung: "A General Framework for Outlier Formulations: Density versus Connectivity", Manuscript.

[16] T. Zhang, R. Ramakrishnan, M. Linvy: "BIRCH: An Efficient Data Clustering Method for Very Large Databases", Proc. of ACM SIGMOD Intl. Conf. On Management of Data, , 1996, pp 103 - 114.