An Efficient Reference-based Approach to Outlier Detection in Large Datasets

Yaling Pei
Department of Computing Science
University of Alberta
Edmonton, AB, Canada T6G 2E8
yaling@cs.ualberta.ca

Osmar R. Zaïane
Department of Computing Science
University of Alberta
Edmonton, AB, Canada T6G 2E8
zaiane@cs.ualberta.ca

Yong Gao
Computer Science
Irving K. Barber School of Arts and Sciences
University of British Columbia Okanagan
Kelowna, BC, Canada V1V 1V7
yong.gao@ubc.ca

Abstract

A bottleneck to detecting distance and density based outliers is that a nearest-neighbor search is required for each of the data points, resulting in a quadratic number of pairwise distance evaluations. In this paper, we propose a new method that uses the relative degree of density with respect to a fixed set of reference points to approximate the degree of density defined in terms of nearest neighbors of a data point. The running time of our algorithm based on this approximation is $O(Rn \log n)$ where n is the size of dataset and R is the number of reference points. Candidate outliers are ranked based on the outlier score assigned to each data point. Theoretical analysis and empirical studies show that our method is effective, efficient, and highly scalable to very large datasets.

1 Introduction

Detecting distance-based outliers in very large datasets has attracted much attention over the past several years in data mining. Compared to traditional outliers studied in statistics [1, 6], the definition of distance-based outliers is distribution-free, more flexible, and more computationally feasible. A bottleneck to the detection of distance-based outliers is that a nearest-neighbor search is required for each of the n data points. Consequently, straightforward implementations such as the Nested-Loop method need to compute the distance between each pair of data points, resulting in an $O(n^2)$ running time.

Since the seminal work of Knorr and Ng [8], much effort has been devoted to improving the efficiency of algorithms for detecting distance-based outliers. By using spatial index data structures such as the k-d tree and its variants, the average running time can be reduced to $O(n \log n)$ with a hidden constant depending exponentially on the dimensionality of the data. Several heuristics have also been proposed to reduce the number of required nearest neighborhood search. In [8], a cell-based approach for detecting distance-based outliers was investigated, which is still exponential in the dimensions, but linear in the size of the dataset under the assumption that both of the two negatively-correlated parameters (percentage p and radius D) of the algorithm are set to their ideal values. In [12], some clusters of data points are eliminated from consideration based on the result of pre-clustering the dataset, a task that is dual to outlier detection. In [2], it is observed that by keeping track of the closest neighbors found so far, the nearest-neighbor search for a specific data point can be stopped if it becomes clear that the data point cannot be one of the pre-specified number of outliers. While the algorithm can indeed prune many distance calculations, the worst-case running time is still $O(n^2)$. Empirical evidence and theoretical arguments under some assumptions in [2] show that the algorithm based on this observation may have a sub-quadratic execution time in practice.

In this paper, we propose a new approach to reduce the number of distance evaluations. The idea is to rank the data points based on their relative degree of density with respect to a fixed set of reference points. For each reference point, we calculate its distance to each of the data points and transform the original data space into a one dimensional dataset. Based on the obtained one-dimensional dataset that contains the distances from a reference point to each data point, the relative degree of density (w.r.t the reference point) of each data point is calculated. The overall relative degree of density of a data point is defined as the minimum relative degree of density over all the reference points. The execution time of the algorithm is in $O(Rn \log n)$ where R is the number of reference points and n is the size of the dataset. Theoretical analyses and extensive empirical studies indicate that the proposed approach is effective, efficient, and highly scalable in detecting global and local outliers in large datasets.

The paper is organized as follows. In Section 2, we review the basic definitions of distance-based outliers and their extensions in the literature. In Section 3, we describe the proposed reference-based outlier detection method, including the definition of the reference-based outlier score and its relationship with the traditional distance-based outlier detection approaches. Also discussed in Section 3 are (1) an algorithm that compute the reference-based outlier score in $O(Rn \log n)$ time and several implementation details; and (2) a comparison study on the benefit of using the reference-based approach in detecting local outliers. In Section 4, we report empirical performance evaluation of our proposal on synthetic and real-world data. Section 5 is the conclusion.

2 Review of Distance-based Outliers

In the following discussion, let $X = \{x_1, \cdots, x_n\}$ be the dataset and let $d(\cdot,\cdot)$ be a distance metric.

Distance-based outliers are defined by using some degree of density relative to the nearest neighbors, or the so called neighborhood density of a data point, under a distance metric [8, 12, 4]. Let x be a data point in X, k be an integer and t be a real number, the neighborhood density of a data point $x \in X$ can be defined in three ways:

Definition 2.1. *0-1 Density $D(x, k, t)$: $D(x, k, t) = 0$ if there are fewer than k other data points within distance t and $D(x, k, t) = 1$ otherwise;*

The definition considers being an outlier as a binary property such that the obtained density for each data point divides the whole dataset into exactly two groups: inliers and outliers. There is no measure of how much a data point is outlying and the identified outliers are not ranked.

Definition 2.2. *Max Density $D^m(x, k)$: $D^m(x, k)$ is the reciprocal of the distance to the k-th nearest neighbor.*

Definition 2.3. *Average Density $D^a(x, k)$: $D^a(x, k)$ is the reciprocal of the average distance to the k nearest neighbors.*

Definitions 2.2 and 2.3 introduce the ranking mechanism based on the distance to the k nearest neighbors of a point. The identified outliers are more meaningful since the information of the degree of being an outlier has been integrated into the analysis process.

Some local outlier detection methods [3, 7, 14] generalize the above concepts further. For example, the well-known local outlier factor (LOF) introduced in [3] measures the degree of being an outlier by taking into consideration the data point's relative density as compared to those of its nearest neighbors. The advantage of LOF is that the local densities of the non-outlier datapoints will have less impact on the ranking of the outliers. The major parameter in LOF is *MinPts*, the minimum number of the nearest neighbors to consider. This parameter is highly application-dependent and some insight into the structure of the dataset is required in order to set it correctly. What makes the selection of *MinPts* even harder is the fact that the LOF of a given data point is not monotone in *MinPts*, as has been observed by the authors [3]. Another related issue with LOF is the existence of duplicated data in a dataset. Roughly speaking, the LOF of duplicated data points is infinity unless the *MinPts* is larger than the number of duplicated data points. As has been mentioned in [3], this difficulty can be overcome by slightly changing the original definition of LOF to ignore the neighboring data points that are duplicated.

Our approach follows Definition 2.3. Based on the average distance to the k nearest neighbors, each data point is assigned an outlier score indicating the degree of its deviation from its close neighbors. Outliers are those with a low neighborhood density but high outlier scores.

3 Reference-based Outlier Detection Method

We use the relative degree of neighborhood density with respect to a fixed set of *reference points* to approximate the degree of density defined in the distance-based method. Let $X = \{x_1, ..., x_n\}$ be a dataset and p be a point (not necessarily in X). Consider the vector that consists of the distances between p and each of the data points in X:

$$X^p = \{d(x_i, p), 1 \leq i \leq n\},$$

which can be viewed as a one-dimensional representation (w.r.t p) of the original data.

Definition 3.1. *Given a data point $x \in X$. A data point $y \in X, (y \neq x)$ is the reference-based nearest neighbor of x with respect to the vector X^p if*

$$|d(x, p) - d(y, p)| = \min_{1 \leq i \leq n} |d(x, p) - d(x_i, p)|$$

where the minimum is taken over all the $x_i \in X$ and $x_i \neq x$.

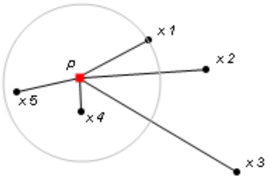


Figure 1. Reference-based nearest neighbors in the one-dimensional dataset X^p

The above idea is illustrated in Figure 1 with a small dataset $X = \{x_1, x_2, x_3, x_4, x_5\}$. For the given reference point p, we have a one-dimensional dataset, whose values are the distances to p, i.e., $X^p = \{d(x_1, p), d(x_2, p), d(x_3, p), d(x_4, p), d(x_5, p)\}$. For a given data point, the reference-based nearest neighbor is the closest point to it in the one dimensional data space X^p. For example, the reference-based nearest neighbor of x_1 is x_5 and the reference-based nearest neighbor of x_2 is x_1. Intuitively, reference-based nearest neighbors with respect to p are not necessarily the closest in the original data space. For 2D data, points located on the same circle (p is the center) have a reference-based distance of 0. This property indicates that we usually need more than one reference points to improve the distance approximation in order to find global outliers. On the other hand the use of reference points is a major factor to contribute to the successful identification of local outliers in a complex dataset. We will discuss this in detail in the next section.

Definition 3.2. *Let x be a data point in X and $\{x_1, ...x_k\}$ be the set of k reference-based nearest neighbors to x. The relative degree of density for x in the one-dimensional data space X^p, denoted as $D(x, k, p)$, is defined as*

$$D(x, k, p) = \frac{1}{\frac{1}{k}\sum_{j=1}^{k} \mid d(x_j, p) - d(x, p) \mid}.$$

Given a reference point, the neighborhood density of x is the reciprocal of the average distance to its k reference-based nearest neighbors in the one-dimensional space $X^p = \{d(x_i, p), 1 \leq i \leq n\}$.

Definition 3.3. *Let $P = \{p_1, \cdots, p_R\}$ be a set of R reference points. We define the neighborhood density of a data point x w.r.t. P as*

$$D^P(x, k) = \min_{1 \leq r \leq R} D(x, k, p_r)$$

where k is a fixed parameter, indicating the number of reference-based nearest neighbors considered for each reference point.

Essentially, data points deviated from their surrounding data patterns will get lower neighborhood density values. Based on the neighborhood density, each data point is assigned a reference-based outlier score, or ROS, which is defined as

$$ROS(x) = 1 - \frac{D^P(x, k)}{\max_{1 \leq i \leq n} D^P(x_i, k)}$$

Data points in a given dataset are ranked according to their relative degree of density computed on a set of reference points. Outliers are those with higher values of outlier scores.

3.1 Compatibility with Distance-based Method

By using a set of reference points, we intend to best approximate the k nearest neighbor search in the original data space so that global outliers can be identified in a more efficient way. If the distance metric satisfies the triangle inequality, we have the following observations which show evidence on the effectiveness of using the relative degree of density to approximate the traditional definition of density.

Consider a reference point p and two data points $x_i, x_j \in X$. Based on the triangle inequality, we have

$$|d(x_i, p) - d(x_j, p)| \leq d(x_i, x_j).$$

Note that the two sides in the above equation are equal when p, x_i and x_j are on the same line.

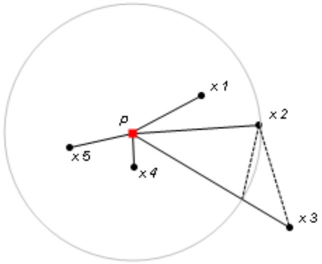


Figure 2. Reference-based nearest neighbors satisfying the triangle inequality

Using the dataset in Figure 1 as a simplified example, we can see from Figure 2 that the reference-based nearest neighbor of x_3 is x_2, and

$$|d(x_3, p) - d(x_2, p)| \leq d(x_2, x_3).$$

Formally, we have the following

Lemma 3.1. *For any set of reference points P and any data point $x_i \in X = \{x_1, x_2, \cdots, x_n\}$,*

$$D^P(x_i, k) \geq D^a(x_i, k)$$

where $D^a(x_i, k)$ is the average density as defined in Definition 2.3.

Proof. Let $\{x_{i_1}, \cdots, x_{i_k}\}$ be the k nearest neighbors of x_i with respect to the distance in the original data space X. Then, based on the triangle inequality,

$$\frac{1}{k}\sum_{j=1}^{k} |d(x_i, p) - d(x_{i_j}, p)| \leq \frac{1}{k}\sum_{j=1}^{k} d(x_i, x_{i_j}). \qquad (1)$$

Since $\{x_{i_1}, \cdots, x_{i_k}\}$ are not necessarily the k referenced-based nearest neighbors of x_i in the data space X^p, we have by Definition 3.2

$$D(x_i, k, p) \geq \frac{1}{\frac{1}{k}\sum_{j=1}^{k} |d(x_i, p) - d(x_{i_j}, p)|} \qquad (2)$$

From equations (1) and (2), we get

$$D(x_i, k, p) \geq \frac{1}{\frac{1}{k}\sum_{j=1}^{k} d(x_i, x_{i_j})}$$

Since above holds for any reference point in P, it follows that

$$D^P(x_i, k) \geq D^a(x_i, k).$$

□

The above shows that for a given data point x, the reference-based density is lower bounded by the neighborhood density computed using the traditional k nearest neighbor search method. If a data point has a small enough density to be identified as outliers using the reference-based approach, it will have a even smaller density value using the distance-based method. If, instead of ranking, we are to use a density threshold α to determine the outliers, then the above analysis indicates that outliers detected under our reference-based density definition are also outliers under the original density definition.

The following special cases are also interesting and provide further justification to our proposal. When the data set is one-dimensional, a single reference point (say 0) is sufficient and our approach is equivalent to the traditional distance-based approach. On the other hand, if the dataset itself is used as the set of reference points, then our approach is identical to the traditional KNN approach. Formally, we have the lemmas which are easy to prove.

Lemma 3.2. *Assume that the dataset $X = \{x_1, \cdots, x_n\}$ is one dimensional. Then, $D^P(x_i, k) = D^a(x_i, k), \forall x_i \in X$ for any set P of reference points.*

Lemma 3.3. *Assume that the set P of reference points contains a data point x_i in the dataset $X = \{x_1, \cdots, x_n\}$. Then, $D^P(x_i, k) = D^a(x_i, k)$.*

3.2 Algorithm and Its Implementation

Let $P = \{p_r, 1 \leq r \leq R\}$ be a given set of reference points. The algorithm finds the potential outliers in the dataset X in three major steps:

1. For each reference point $p \in P$, sort the original dataset X in the one-dimensional space $X^p = \{d(x_i, p), 1 \leq i \leq n\}$, i.e., data points in X are ordered according to the distances to p.

2. For each data point $x \in X$, find the k reference-based nearest neighbors and compute the average neighborhood density $D(x, k, p)$;

3. Set $D^P(x, k)$ of each point x to be the minimum of $D(x, k, p_r)$ w.r.t. P and compute the outlier score ROS.

Outliers tend to have a higher value of ROS and they are ranked according to their ROS value. See Algorithm 1 for the details.

Algorithm 1 The Algorithm for computing $D^P(x, k)$

```
Input: dataset $X = \{x_i, 1 \leq i \leq n\}$, reference point set $P = \{p_r, 1 \leq r \leq R\}$
Let $X^p$ be one-dimensional data space containing distances to a reference point $p$ and $k$ be a positive integer

$p = p_1$;
$X_s$ = mergeSort($X$) according to $X^{p_1}$;
for each $x \in X$ do
    $D(x, k, p)$=computeDensity($x, X_s$);
    $D^P(x, k) = D(x, k, p)$;
end for
for each $2 \leq r \leq R$ do
    $p = p_r$;   //next reference point closest to $p$
    $X_s$ = Sort($X_s$) according to $X^{p_r}$ using an appropriate sorting algorithm;
    for each $x \in X$ do
        $D(x, k, p)$=computeDensity($x, X_s, k$);
        if $D(x, k, p) < D^P(x, k)$ then
            $D^P(x, k) = D(x, k, p)$;
        end if
    end for
end for
```

It takes $O(n)$ time to compute the distance vector X^p for each reference point $p \in P$. The calculation of the average neighborhood density $D(x, k, p)$ involves finding the k reference-based nearest neighbors. Since the reference-based nearest neighbors are calculated on the one-dimensional space X^p, it suffices to find them by sorting the original dataset X using the values in X^p (i.e., the distances to the given reference point p) as the key. Sorting the distance vector X^p can be done in $O(n \log n)$ time. Once the distance vector X^p has been sorted, the calculation of $D(x, k, p)$ for all the data points $x \in X$ can be done in $O(n)$ time. Thus, the overall time complexity of our algorithm is in $O(Rn \log n)$ where R is the number of reference points.

3.2.1 Further Speedup

To further improve the efficiency in computing $D^P(x, k)$, we make the following observation. Assume that p_1 and p_2 are two reference points and that $d(p_1, p_2)$ is small. Then, data points in X sorted according to their distances to p_1 is usually "almost" sorted according to their distance to p_2. Thus, if we have processed p_1 and recorded the corresponding sorted order $X_{p_1} = \{x_{i_1}, \cdots, x_{i_n}\}$, we can calculate $D(x, k, p_2)$ by sorting the ordered list $X_{p_1} = \{x_{i_1}, \cdots, x_{i_n}\}$ with the various adaptive sorting algorithms that can take advantage of the "near sortedness" of the vector X_{p_1}. One example of such adaptive sorting algorithms is the simple insertion sort whose running time is in $O(n + REV)$ where REV is the number of pairs of elements whose relative order is wrong [5]. Therefore, while the worst case execution time of computing $D^P(x, k)$ is $O(Rn \log n)$, the practical execution time of our algorithm can be much lower.

3.2.2 Determination of Reference Points

The determination of suitable reference points plays an important role in terms of both the effectiveness and the efficiency of our algorithm. In our implementation, each reference point is not necessarily a data point in X. We use as reference points the vertices on a grid obtained by partitioning the axes in the data space to facilitate the selection of the closest next reference point in the second for loop of the above algorithm. The advantages of using vertices on the grid over randomly selecting reference points is two-fold: (1) Reference points are evenly distributed in the whole data space, and (2) the outlier detection result is deterministic, i.e., the obtained results would be the same with each run.

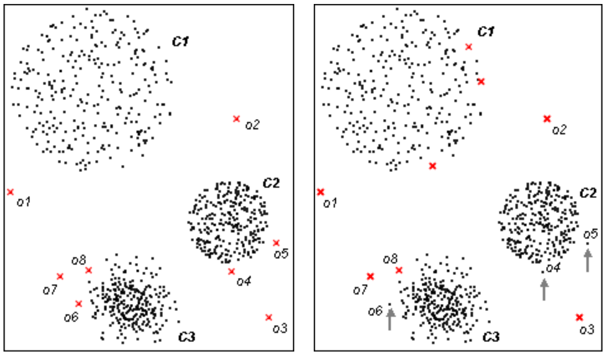
Recall that the overall running time of our algorithm is in $O(Rn \log n)$ where R is the number of the reference points that is determined by the characteristics of the dataset rather than the size n of the dataset. For simple datasets that contain only one cluster, a few reference points are enough to correctly detect the outliers in even very large datasets. For more complicated datasets, domain knowledge can usually help in determining the number of reference points. By using the grid vertices, we are able to partition the space incrementally from coarse resolution to fine resolution to determine the appropriate number of reference points. Also, notice that all calculations in the current partition is not lost in the next partition, and only calculation for additional reference points is computed. Our experiments show that with datasets containing a single data distribution, a few reference points are enough to correctly detect all the outliers even with very large dataset.

3.3 Detecting Global and Local Outliers in Complex Datasets

The distance-based method is static in that it uses parameters with fixed values for all the data points in a dataset. It ignores the cases where data patterns have different densities, thus considers all data points in a dataset in the same setting. Consequently, it always assigns low density values to data points located in sparse regions even though some of them are deep in a sparsely distributed clusters. The authors in [3] also argue that the distance-based outlier detection method can only take a global view of the dataset, resulting in its failure to identify outliers local to certain clusters in a complex dataset.

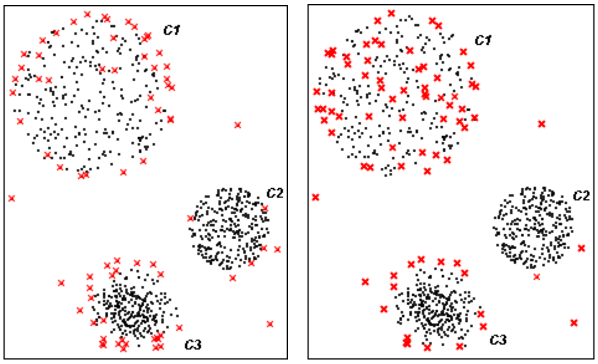
By using a set of reference points, our reference-based approach is dynamic and able to see the whole dataset from various viewpoints. It is possible that the reference-based nearest neighbors of a given data point are different with respect to different reference points. Therefore, at one reference point, the local outliers may have a high neighborhood density, while at another reference point, it may be shown lying in a very sparse neighborhood. Since the reference-based neighborhood density of a data point x is determined by the minimum among all its densities computed based on the set of reference points, it is guaranteed that with a set of reference points evenly covering the data space, data deviated from the surrounding data patterns will be assigned lower density values. In the following, we will use examples to show that reference-based method can successfully identify local outliers as well as global outliers in complex datasets that contain clusters of different densities.

To facilitate the evaluation of data clustering and outlier analysis, we developed a synthetic data generation system that can produce datasets with various cluster and outlier patterns. Interested readers are referred to [10, 11] for the details. Using our synthetic data generator, we generated a small 2D dataset X that contains 850 data points. There are three clusters $C1, C2$ and $C3$, where data in clusters $C1$ and $C2$ are uniformly distributed and data in $C3$ are in normal distribution. Cluster $C1$ has a smaller density compared to $C2$ and $C3$. In addition to the three clusters which form the main body of the dataset, there are a few local outliers as well as some global outliers in this dataset. In our experiment, we set the number of reference-based nearest neighbors $k = 4$. To ensure that the reference points are evenly positioned in the 2D data space, we set the number of reference points to be the 16 points on the 4 by 4 grid that divides the data space evenly. In the first run to mine the top 3 outliers, the three global outliers o_1, o_2 and o_3 are found. See Figure 3 for the details of the dataset. Since we want to check if the local outliers specific to the clusters can be found, the program is set to mine the top eight outliers in the second run. The result is displayed in Figure 3 (a), where the outliers are marked with a cross.



(a) (b)

Figure 3. Finding top eight outliers (a) using reference-based method and (b) using the traditional KNN method



(a) (b)

Figure 4. Finding top 76 outliers (a) using reference-based method and (b) using the traditional KNN method

We tested the distance-based outlier detection method with the same dataset. The implementation is based on Definition 2.2, where the neighborhood density $D^m(x, k)$ is the reciprocal of the distance to the k^{th} nearest neighbor. As above, we set $k = 4$. The top 3 outliers identified are the same as those identified using our method. However, distance-based method fails to find the two local outliers specific to cluster $C2$ when mining the top 8 outliers. The result is shown in Figure 3 (b), where the undetected local outliers are marked with arrows. It is observed that $D^m(x, k)$-based method tends to identify data points in sparse regions as outliers such as those located in the perimeter of cluster $C1$. The ranking result shows that using $D^m(x, k)$ method, o_5 is ranked in the 9^{th} place but o_4 is ranked in the 76^{th} place. In order to find the local outlier o_4, the distance-based method has to falsely mark many other data points as outliers.

This actually poses an interesting question: what will happen if the number of real outliers in a dataset is less than the number of top outliers the program is set to mine? Such issue can be important with large real world datasets whose domain knowledge is unknown. We test both reference-based approach and the distance-based approach to mine the top 76 outliers, when o_4 can only be identified using the distance-based method. The results are shown in Figure 4. We can see that in Figure 4 (b) many data points deep in the relatively sparse cluster $C1$ are falsely marked as outliers before distance-based method is able to find the real local outlier o_4. Although there are a few points inside cluster $C1$ that are marked as outliers by our approach, they ranked after most of the real outliers. Careful observation shows that they are the next best outliers within $C1$ compared to those points on the border of $C1$. The overall outliers identified indicate that the reference-based approach is more likely to consider those data points deviating from or lying on the edge of the data patterns as outliers.

The above examples demonstrate that compared to the distance-based approach, reference-based method is not only superior in differentiating data deep inside a sparsely pact cluster from local outliers deviated from a dense pattern in a dataset, but also capable of eliminating false identification of outliers inside the sparse data patterns of a complex dataset.

One of the motivations of the reference-based approach is to best approximate the neighborhood density obtained using the traditional k nearest neighbor approach so that global outliers can be effectively identified in a rather efficient way. In this sense, the more the number of reference points, the better the distance approximation and the more accurate the identification of global outliers in a dataset. However as shown earlier, when the whole dataset is used as the set of reference points, reference-based approach is reduced to the traditional distance-based approach, which can lead to false identification of local outliers in a complex dataset. While this is an intrinsic problem with the existing distance-based approaches, it can be easily solved in our reference-based method by starting with a few number of reference points and incrementally increasing the number of reference points. Since the reference-based nearest neighbors of the data points computed with respect to a given set of reference points can be reused for the next round when more reference points are added, such adjustment will have little impact on the performance of the outlier detection method. The detection process stops when certain outliers have been found and such inspection often involves human intervention.

To further explain why the reference-based density definition can facilitate the identification of local outliers, we use the previous dataset as an example. Suppose we have a set of 16 reference points $P = \{p_1, ..., p_{16}\}$, which are the vertices on a grid obtained by partitioning the axes in the data space as shown in Figure 5.

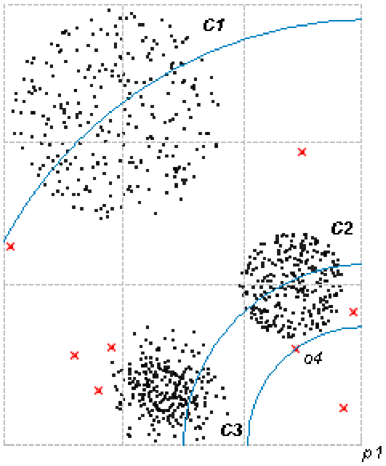


Figure 5. Finding local outliers using reference-based approach

Take the reference point p_1 as an example. If we draw circles with p_1 as the center and the distance to each data point as the radius, the k reference-based nearest neighbors of a certain data point x with regard to p_1 would be those on or closest to the circle where x is located. As is shown in the plot, there are no other data points that fall on the circle that passes through the local outlier o_4. In contrast, data deep in clusters $C1, C2$, and $C3$ generally have nearest neighbors with little or no distance difference. Consequently, o_4 will have a relatively smaller neighborhood density in the one dimensional data space X^{p_1} than data in the clusters. Though with another reference point, say, p_2, it is possible that o_4 may be assigned a high density if the circle (p_2 as the center and $d(p_2, o_4)$ as the radius) pass through one or more dense data patterns, the overall neighborhood density is determined by the minimum among all the calculated densities for x in terms of X^{p_r}. Assume that the reference points are sparsely distributed such that each data point in the data patterns are not isolated by the circle around a reference point, i.e., the number of reference points accounts only a small portion of the number of the whole data, then data deep in clusters are ensured to have very close neighbors which contribute to high neighborhood densities with regard to each reference point. Therefore, o_4 will have a lower neighborhood density and a higher outlier score than data in the clusters.

It is noticed that there is indeed a tradeoff between the number of reference points and the ability of the reference-based method to detect global and/or local outliers. On the one hand, if all the data points in the original dataset are used as the reference points, then our approach is equivalent to the traditional KNN approach. On the other hand, by using a small set of reference points, local outliers will have a better chance of being detected at the potential cost of some inaccuracy in detecting global outliers. We leave it as a future research topic to investigate how to achieve such a tradeoff and how to integrate our approach with the various approaches to local and/or global outlier detection in the literature such as LOF [3].

4 Empirical Evaluation

In this section, we show experimently that the proposed method can efficiently identify local and global outliers in various datasets. We compare the performance of our approach with the existing distance and density based methods.

4.1 Results on Synthetic Datasets

To compare the performance of the proposed reference-based approach with the existing k nearest neighbor approach, our first test is to see how fast each method can find outliers in large datasets. We generated a set of synthetic datasets using the synthetic data generating system [10, 11]. It is well accepted that the evaluation of outlier detection involves human intervention. We limit our experiments to two-dimensional synthetic data so that the evaluation can be performed by simple visual inspection. The sizes of these datasets range from 1,000 to 500,000 data points. To be consistent, each dataset has a major data pattern that is normally distributed. Based on the fact that outliers accounts for only a very small portion in a dataset, the number of outliers to be mined is set to be 1% of the data size in all the following experiments. For the reference-based approach, the number of reference points can be set to a constant for all these datasets. This is because all the test datasets have similar probability distribution and each of them contains only one normally distributed cluster. In such cases, a few reference points that cover different areas in the data space are sufficient to ensure the successful detection of outliers. As discussed before, the reference points are evenly located in the grid vertices of the data space. In our experiment, the number of reference points is set to 9 for all these datasets. We implemented the reference-based method in Java to facilitate the visualization of the outlier detection results. For the distance-based method, we downloaded the executable version of Orca, the C implementation of the distance-based method discussed in [2]. Since Orca is based on the distance-based algorithm with a near-linear running time, it is believed to be one of the most efficient KNN-based outlier detection methods. To compare the two programs, we run our method in command line mode so that the execution time includes time for input and output as Orca does. Orca also requires a preprocessing step to randomize the order of the original data and convert it to binary format. In our experiments, the cost of data preprocessing for Orca is not counted when recording Orca's execution time. The number of nearest neighbors is set to be $k = 6$ for both programs.

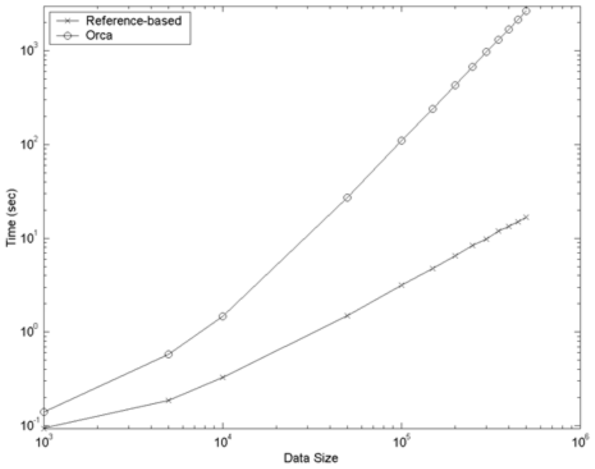


Figure 6. Log-scale execution time vs data size for reference-based approach and Orca

Since each dataset contains a single cluster that is normally distributed, both programs can effectively mark the data lying farther away from the mean as outliers in a dataset. There is hardly any difference in the identified outliers using the two methods. However, the difference of execution time between our reference-based approach and Orca gets bigger and bigger with the increase of the data size, despite the fact that our implementation of ROS is in Java while Orca is in C. Figure 6 is the log-scale plot of execution time vs data size for the two methods. Although the plot for Orca does not include the data preprocessing time, Figure 6 shows that with large datasets, reference-based approach has orders of magnitude improvements in execution speed compared to Orca, one of the optimized implementation of the distance-based approach.

Outliers are defined as data points that deviate from the main patterns of a dataset. They are most likely to be considered in the context of clusters with different types of data distributions. That is, an object is marked as an outlier if it is isolated from the clusters in a given dataset. To test if the reference-based approach can effectively find meaningful outliers in a complex dataset, we generated a dataset of 10,000 datapoints. There are six data patterns. Three are uniformly distributed and the other three are normally distributed. The densities of the six data patterns are different.

Unlike the datasets containing only normally distributed data where outliers lie on the outer fringe of each cluster, this dataset has two types of outliers: outliers uniformly distributed around the main data patterns and outliers lying on the outer skirt of the three normally distributed clusters. To demonstrate the effectiveness of reference-based approach in finding both global and local outliers, we implemented LOF according to [3] that is well known for its performance in detecting local outliers. We compare the effectiveness of the distance-based method, LOF, and our ROS on this dataset. For the distance-based method, and ROS, we set $k = 4$. For LOF, we set $MinPts = 30$ as recommended in [3]. The number of reference points for ROS is set to be 2% of the data size.

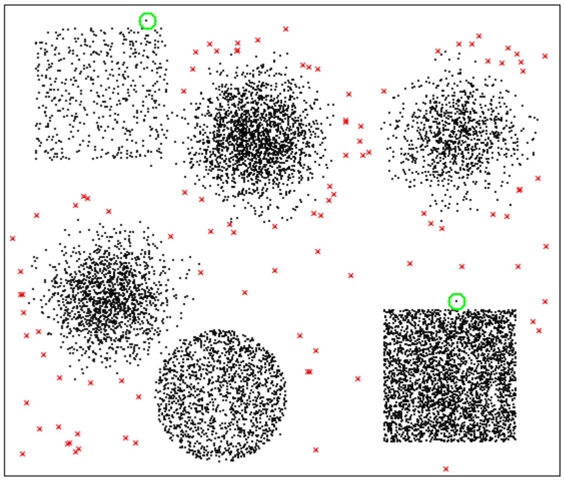


Figure 7. Outlier detection result from distance-based approach

Figure 7, 8 and 9 are the screen shots showing the results for mining the top 100 (1% of data size) outliers. In all the three figures, detected outliers are marked with crosses. Visual inspection shows that the KNN-based method, LOF, and our referenced-based method are all able to discover

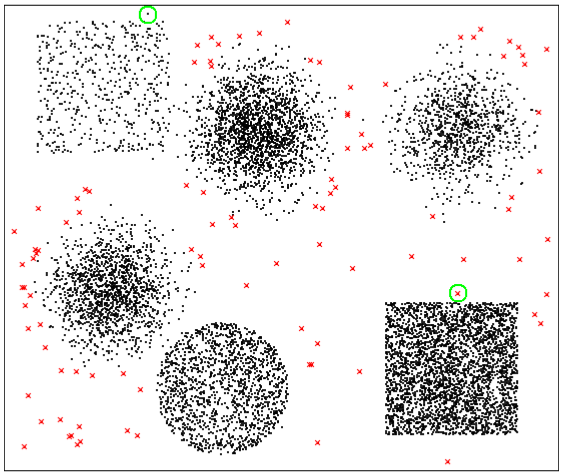


Figure 8. Outlier detection result from LOF

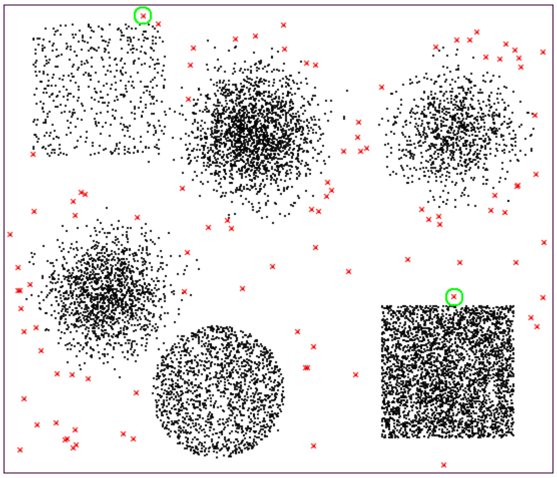


Figure 9. Outlier detection result from reference-based approach

the global outliers. Furthermore, two local outliers (the two crosses surrounded by small circles) are discovered by our reference-based method, while LOF can only find one of them and the KNN-based method fails to find either of them. Therefore, our approach is not only effective in finding outliers that lie far away from the major patterns in the dataset, it also achieves similar or even better results than LOF in detecting local outliers.

4.2 Results on Hockey Data

The National Hockey League (NHL) data have been used as a benchmark in several outlier analysis works [8, 3, 13]. We use the statistics of NHL 2003-2004 season obtained from NHL website [9]. The dataset contains 916 entries. We compare our approach with both Orca and LOF on this dataset. The goal is to see if our approach can efficiently find the outliers identified by Orca and/or LOF in real-world datasets. We set the number of reference points for reference-based approach to be two to the power of the data dimensionality. For example, given a 3D set of data, the reference points are the 8 vertices of a cube in the 3D space with all data inside the cube. We set $k = 4$ for both Orca and our approach and $MinPts = 30$ for LOF to ensure LOF finding meaningful outliers.

We conduct the experiment in a similar way as other outlier analysis works. It consists of two tests. The first test mines the outliers based on the three attributes: games played, goals scored and shooting percentage. LOF, Orca and our reference-based approach achieve identical results and the top three outliers are listed in Table 1(a). The outlier status of the three identified players are obvious. They only played a few games and scored once, but their shooting percentage is unusually high.

The second test is to mine outliers based on the three attributes: points scored, plus-minus statistic and penalty minutes. The top 3 outliers found by our reference-based approach are listed in Table 1(b). Sean Avery is on top because his points and plus-minus figures are moderate but the number of penalty minutes is the highest among all the players. LOF gets the similar result as ours while Orca's result is slightly different. The top rank from Orca, Zdeno Chara, is ranked as the 9^{th} outlier by our approach and 26^{th} outlier by LOF. Careful investigation shows that outliers identified by Orca tend to be in a sparse region regardless of the data distribution. Reference-based method instead favors outliers that deviate from the main data patterns and the results are close to LOF in this aspect. For example, Jody, who is ranked third by our approach lies far away from the trend of the data body due to his low points and plus-minus statistic but extremely high penalty minutes. With 3D datasets of size about 1,000, the execution time for both Orca and our reference based approach is within 0.1 seconds and can be neglected.

5 Conclusion

In this paper, we have proposed an efficient reference-based outlier detection method that uses the relative degree of density with respect to a fixed set of reference points to calculate the neighborhood density of a data point. In addition to being compatible with the traditional distance-based outlier detection methods, our approach performs better in identifying local outliers that deviate from the main patterns in a given dataset. The execution time of our algorithm is $O(Rn \log n)$ where n is the size of dataset and R is the number of reference points. Candidate outliers are ranked based on the reference-based outlier score that is assigned to

Table 1. Outlier detection on NHL(03/04) data

(a) Outlier detection result 1 on NHL(03/04) data

Our Rank	LOF Rank	Orca Rank	Player	Games Played	Goals scored	Shooting Percentage
1	1	1	Milan Michalek	2	1	100
2	2	2	Pat Kavanagh	3	1	100
3	3	3	Lubomir Sekeras	4	1	50

minimum	*1*	*0*	*0*
median	*57*	*4*	*6.6*
maximum	*83*	*41*	*100*

(b) Outlier detection result 2 on NHL(03/04) data

Our Rank	LOF Rank	Orca Rank	Player	Points Scored	Plus- Minus	Penalty minutes
1	1	3	Sean Avery	28	2	261
2	2	2	Chris Simon	28	15	250
3	7	15	Jody Shelley	6	-10	228
⋮	⋮	⋮	⋮		⋱	
9	26	1	Zdeno Chara	41	33	147

minimum	*0*	*-46*	*0*
median	*12*	*-1*	*26*
maximum	*94*	*35*	*261*

each data point. Theoretical analysis and empirical studies show that the reference-based approach is not only highly efficient and scalable to very large datasets, but also able to achieve better results than distance-based approaches. When all the data points are used as references points, our method becomes a distance-based approach finding global outliers. When only one reference point is used, local outliers relative to the reference point are discovered. We advocate the use of a small number of reference points uniformly distributed over the data (using a grid) to find global and local oultiers effectively and efficiently.

Acknowledgment

The authors thank the support from NSERC and the anonymous referees for their comments. Part of the work was done while Yaling Pei was with the University of British Columbia Okanagan.

References

[1] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley & Sons, 1994.

[2] S. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the ninth ACM SIGKDD*, 2003.

[3] M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proceedings of the ACM SIGMOD*, pages 93–104, May 2000.

[4] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. A geometric framework for unsupervised anormaly detection: Detecting intrusions in unlabeled data. In *Proc. Data Mining for Security Applications*, 2002.

[5] V. Estivill-Castro and D. Wood. A survey of of adaptive sorting algorithms. *ACM Computing Surveys*, 24, 1992.

[6] D. Hawkins. *Identification of Outliers*. Chapman and Hall, 1980.

[7] W. Jin, A. K. H. Tung, and J. Han. Mining top-n local outliers in large databases. In *Proceedings of the seventh ACM SIGKDD*, August 2001.

[8] E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24th VLDB Conference*, pages 392–403, August 24-27 1998.

[9] NHL.com. `http://www.nhl.com`.

[10] Y. Pei. Discovering and ranking outliers in very large datasets. Master's thesis, Department of computing Science,University of Alberta., 2006.

[11] Y. Pei and O. Zaïane. A synthetic data generator for clustering and outlier analysis. Technical report, TR06-15, Department of computing Science, University of Alberta, 2006.

[12] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the ACM SIGMOD*, pages 427–438, June 2000.

[13] D. Ren, B. Wang, and W. Perrizo. Rdf: A density-based outlier detection method using vertical data representation. In *Proc. of the Fourth IEEE ICDM*, pages 503–506, 2004.

[14] P. Sun and S. Chawla. On local spatial outliers. In *Proc. of the Fourth IEEE ICDM*, pages 209–216, 2004.