Subspace Outlier Detection in Data with Mixture of Variances and Noise

Minh Quoc Nguyen
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA
quocminh@cc.gatech.edu

Leo Mark
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA
leomark@cc.gatech.edu

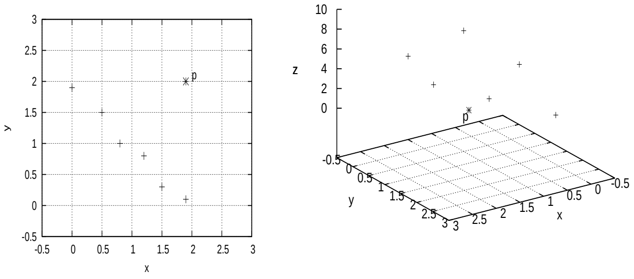
Edward Omiecinski
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA
edwardo@cc.gatech.edu

ABSTRACT

In this paper, we introduce a bottom-up approach to discover clusters of outliers in any m-dimensional subspace from an n-dimensional space. First, we propose a method to compute the outlier score for all points in each dimension. We show that if a point is an outlier in a subspace, the score must be high for that point in each dimension of the subspace. We then aggregate the scores to compute the final outlier score for the points in the dataset. We introduce a filter threshold to eliminate the high dimensional noise during the aggregation. The concept of outlier is extended to allow the discovery of clusters of outliers. An $oscore(C/S)$ function is introduced to rank the clusters accordingly. In addition, the outliers can be easily visualized in our approach.

1. INTRODUCTION

Outlier detection is an interesting problem in data mining since outliers can be used to discover anomalous activities. Historically, the problem of outlier detection or anomaly detection has been studied extensively in statistics by comparing the probability of data points against the underlying distribution of the data set. The data points with low probability are outliers. However, this approach requires a prior underlying distribution of the dataset to compute the outlier scores, which is usually unknown. In order to overcome the limitations of the statistical-based approaches, the distance-based [7] and density-based [3] approaches were introduced to detect outliers , which use k-nearest neighbors (KNN) to compute the similarity between data points. The points that are most dissimilar from the others are considered to be the outliers [7, 3]. The main advantage of this approach over the statistical-based ones is that no prior knowledge of the model nor the distribution of data set is required in order to compute the outliers. In this paper, we will discuss the situations when outliers can not be detected by using these traditional outlier detection methods. From those observations, we introduce an outlier score function based on Chebyshev (L_∞ norm) distance in order to properly rank the outliers. The method can be used to discover and rank clusters of outliers. In addition, it also allows us to visualize the outliers to support the study of the outliers.



(a) 2D. (b) 3D.

Figure 1: The 2D outlier is suppressed in 3D space.

2. PROBLEMS

The purpose of outlier detection is to discover anomalous activities among the set of normal activities. Each activity consists of a set of features representing the information about the activity. We are interested in the unsupervised learning problem where we do not know which features will be useful in determining the anomalous activities. By dismissing any feature, we may not be able to discover the anomalous activities [3]. Unfortunately, the problem of feature selection, i.e. finding the appropriate sets of features for computation, is NP-hard. Thus, it is essential to run the algorithm on the entire feature space to detect outliers. However, this approach may affect the quality of outlier detection because of the problems which we call a mixture of variances and accumulated subdimensional variations.

2.1 Mixture of Variances in Multiple Features

We use a dataset with seven data points to illustrate the first problem of using k-nearest neighbors (L_2) to detect outliers. The data has three features x, y and z in which the domain of x and y is the interval $[0, 2]$ and that of z is the interval $[0, 8]$.

Figure 1a shows a 2D plot for the data points for features x and y. According to the figure, the nearest neighbor distance of any points excluding p is less than 0.64. The nearest neighbor distance of p is 1.39. From those two values, we see that p has an unusually high nearest neighbor distance compared with the other points. Point p is an outlier in this figure. Figure 1b shows the complete plot for the data points for all of three features x, y and z. The range of z is four times that of x and y, which makes the difference in the distance between p and the other points in features x and y insignificant compared with that in feature z. As we can see, the nearest neighbor distance of p is very similar to or less than the average nearest neighbor distance of six other points in the data. According to this figure, p is a normal point.

Those two figures illustrate the problem of using pairwise distance to detect outliers. One may ask if we can normalize the dataset to solve the problem. However, if those points are taken from a larger dataset and they are nearest neighbors of each other, the problem still remains. We can generalize the problem into any arbitrary number of features as follows. Let say $\{\sigma_i\}$ the variances of the features in a subspace that point q is an outlier. If there is a feature j with the variance of σ_j, where $\sigma_j = k_i \times \sigma_i$ and k_i is large, q becomes normal in the new subspace that contains feature j. The variances can be computed from the local area of point q or from the entire dataset, which corresponds to the problem of local outlier and global outlier detection respectively.

An approach to solve the problem is to compute the outlier score for the data points for all possible combinations of features separately. If a point is an outlier in a subspace of the entire feature space, the outlier score of the point is high. However, the problem of feature selection is NP-hard.

2.2 Accumulated subdimensional variations

Let consider three points p, q and r in an n-dimensional space. In this example, p and q are normal points; whereas r is an outlier in an m-dimensional subspace. We denote the i^{th} feature of a point by subscript i. We assume that the difference between p_i and q_i is δ_n for all $i \in [1, n]$. Thus, we have

$$d(p,q) = \sqrt{\sum_i^n \delta_n^2} = \delta_n \sqrt{n} \qquad (1)$$

We further assume that $|p_i - r_i| = \delta_m$ for $i \in [1, m]$ and $|p_i - r_i| = 0$ for $i \in [m+1, n]$. We have

$$d(p,r) = \sqrt{\sum_i^m \delta_m^2} = \delta_m \sqrt{m} \qquad (2)$$

If $d(p,r) = d(p,q)$, we have

$$\delta_n \sqrt{n} = \delta_m \sqrt{m} \Longrightarrow \frac{\delta_m}{\delta_n} = \sqrt{\frac{n}{m}}, \text{where } \delta_m, \delta_n \neq 0 \qquad (3)$$

Let define $r = \frac{\delta_m}{\delta_n}$. We obtain the following expression:

$$r = \sqrt{\frac{n}{m}} \qquad (4)$$

Expression 4 implies that the ratio of the nearest neighbor distance between an outlier and normal points can be as large as $\sqrt{\frac{n}{m}}$ so that the outlier in an m-dimensional space will look normal in n-dimensional space. With $n = 100$ and $m = 4$, we will have $r = \sqrt{\frac{100}{4}} = \sqrt{25} = 5$. Hence, outliers which have a ratio of 5 : 1 or less of the distance of their nearest normal group of points to the density of the group may not be detected. The number of 5d-subspaces is approximately 4×10^6. We call the problem that we can not distinguish if an outlier is a true outlier or a normal point in this example is the problem of accumulated subdimensional variations.

Table 1: Notations and Basic Definitions

$KNN(p)$	$\{q^j \mid q^j \equiv j^{th} nn(p)\}$
p_i	feature i^{th} of a given point p
$d_i(p,q)$	$\lvert p_i - q_i \rvert$
$L_i(p)$	ordered set of point q $\in KNN(p)$ ordered by $d_i(p,q)$
ξ_i^j	$\frac{d_i^{j+1}(p,q^{j+1}) - d_i^j(p,q^j)}{d_i^j(p,q^j)}$

3. OUR APPROACH

3.1 Outlier criteria in high dimensions

In this section, we will provide concrete intuitive criteria for what it means to be an outlier in high dimensions. The next sections will give precise definitions of our outlier score function based on those criteria. In previous works, the distance between a point and its neighbors is used to define the degree of being an outlier for a point. The results are based on the Euclidean distance. This approach is self-explanatory and intuitive in low dimensions. However, it is a problem in high dimensions as shown in section 2.2. Thus, we choose the Chebyshev distance to measure the distance in our method because the variances are not cumulative in high dimensions in L_∞ (by definition, the Chebyshev distance between any two points p and q is the maximum of $|p_i - q_i|, \forall i = 1 \ldots n$). Let say we have a sample S such that each feature of the points in S follows the distribution $N(\mu_i, \sigma), \forall i = 1 \ldots n$. With the L_2 norm, the distance between two points can vary from 0 to $\sigma\sqrt{2n}$. However, the range of difference will be limited to the interval $[0, 2\sigma]$ in L_∞.

We use an axis-parallel hyper squared rectangle R (or hypercube) to represent the local region of a point p where p is its center in Chebyshev space. The rectangle defines the bounds on how much a point can deviate from the center on the axes such that it is still considered a near neighbor of p. A point q is an outlier with respect to p in region R with length 2d (the distance between any two parallel sides) if its distance to R is significantly larger than the bounds, denoted by $||p - R|| >> d$. To be more precise, we have the following postulate:

POSTULATE 1. *Given a boundary hyper squared rectangle R with length 2d of a point p, A point q is an outlier with respect to point p if distance$(p, R) > \kappa d$ for some large κ.*

THEOREM 1. *A point q is an outlier with respect to p in region R with length 2d in n-dimensional space iff q is an outlier w.r.t p in at least one dimension i, where $i \in [1, n]$.*

PROOF. The projected rectangle into a dimension i is a line segment D_i where p is its center. Since the length of the rectangle is 2d, the length of the line segment is $2d$. Since q is an outlier w.r.t. p, we have $distance(p, R) > \kappa d$. As defined, the distance from a point to a rectangle is the maximum distance from the point to the surfaces of the rectangle in the Chebyshev space. Since the surfaces are orthogonal or parallel to the line segment, $\exists i$: $distance(p_i, D_i) > \kappa d$. Thus, p is an outlier in at least one dimension i. Conversely, if q is an outlier w.r.t p in at least one dimension i, we have $distance(p, R) > \kappa d$ by the Chebyshev distance definition. Therefore, q is the outlier w.r.t p in the n-dimensional space. □

The discussion above gives us a basis to talk about the outlier relative to a point. We can extend the concept to a set of points S.

POSTULATE 2. *Given a set of points S, if a point q is an outlier with respect to all points p in S for some rectangle R of p, then q is an outlier in S.*

It is straightforward to see that if p is an outlier to all points, then p is an outlier of S. However, if p is outlier only to a few points in S, p is not an outlier. The criteria is self-explanatory which allows us to capture the concept of an outlier in high dimensions. The next section will provide the intuition on how the subspace outliers can be detected according to this criteria.

3.2 Intuition

In figure 1a, if we project point p into any dimension x or y, the outlier information about point p will be lost, which means that we could not discover it as an outlier. In [3], Breung et al proposes the use of KNN to detect outlier p in such cases. However, KNN performs poorly in high dimensional data. One of the reasons is that the Euclidean distance accumulates all the variances in each dimension into the entire space (section 2.2). From theorem 1 in section 3.1, we observe that we can compute the outlier score in each dimension instead of computing the outlier in all dimensions so that the dimensions where the points do not show up as outliers are not included in the outlier score. Then, we can aggregate all the scores into a final score. This approach can prevent noise from being accumulated. In order to compute the outlier score in each dimension without the loss of information, we compute the deviation of the points in each dimension with respect to its neighbors in the entire space, which corresponds to the boundary rectangle of the points.

From the problem of mixtures of variances in figure 1b, we observe that the differences in the variances suppress the subspace outliers. The dimension with high variance will dominate those with low variance. Since the outlier detection is unsupervised learning, we treat all dimensions equal. In other words, the rate of deviation is more important than the module of variances. This suggests that we compute the dissimilarity of the points with respect to the average variance of the points in each dimension in the local region where the points belong. Thus, the hyper squared rectangle in section 3.1 given above can be generalized to hyper rectangle and the outlier criteria can be expressed in terms of the ratios of the distances.

3.3 Definitions

We use $k^{th}nn(p)$ to denote the k^{th} nearest neighbor of p in L_∞ and $kdist(p)$ (k-distance) is the distance from p to its k-nearest neighbor. The k-distance defines the relative density of the points in a dataset. Next, we want to compute the projected density of a point p into each dimension which we call dimensional density. The densities form the boundary hyperrectangle for the point. A simple approach to compute the dimensional densities is to average the local distances from a point p to its neighbors for the dimension under consideration. However, the result depends on parameter k. The key question is how many neighbors should we consider in computing the dimensional densities. With small k, the dimensional density is less biased but the variance is high. In contrast, the dimensional density will be more biased with large k. In [13], the authors introduce a definition of adaptive nearest neighbors which allows us to determine the natural dimensional density in terms of the level of granularity at each point. According to Nguyen et al [13], if a point is in a uniformly distributed region, k should be small since the distance between the point and its few nearest neighbor is approximately the local density of the point. Otherwise, the decision boundary [13] and the level of granularity are used to select k. We adapt these concepts to define local dimensional density.

We create an ordered list L_i of the nearest neighbors of p ordered by d_i for each dimension. All q $\in KNN(p)$, where KNN is the list of nearest neighbors, whose $d_i(p, q) = 0$ should be eliminated from the list. To simplify the problem, we assume that there is no q such that $d_i(p, q) = 0$. Let say we have $L_i \equiv \{q^1, \ldots, q^k\}$ where $q^j \in KNN(p)$. For each $j \in [2, \ldots, k]$, we compute the ratio ξ_i^j which is $\frac{d_i^j(p,q^j) - d_i^j(p,q^{j-1})}{d_i^j(p,q^{j-1})}$. If p is in a uniformly distributed region, ξ_i^j will uniformly increase with j in such cases we can use $d_i(p, q^1)$ to represent the local dimensional density of p in dimension i regardless of the level of granularity. A point where there is a sharp increase in ξ_i^j is called the decision boundary of the local distance of point p. We can measure the sharpness by a parameter λ, i.e. $\xi_i^j \geq \lambda$. The decision boundaries are used to adjust the level of granularity.

We use a parameter **z** to determine the level of granularity in detecting the outliers. We then define the local dimensional density of a point p with a granularity of level **z** as follows:

DEFINITION 1. *Given q^{j_z} is the z^{th} decision boundary point of a point p, the local dimensional density of p with the granularity level **z** in dimension i is*

$$\gamma_i(p) = \begin{cases} d_i(p, q^1) & , \xi_i^j < \lambda \vee z = 1, \forall j \in [1, \ldots, k] \\ d_i(p, q^{j_z}) & , \textit{otherwise} \end{cases} \tag{5}$$

Next, we compute the average local distance in each dimension for a local region S. Region S is a set of points in a local region of the dataset. With $|S|$ large enough, formula 6 is the estimate of the expected mean of local dimensional densities of the points in the region. In the formula, the local distances whose value is zero are removed from the computation.

Definition 2. *Dimensional average local distance*

$$\bar{\delta}_i = \frac{\sum \gamma_i(q)}{m}, m = |\{\gamma_i(q)/q \in S \wedge \gamma_i(q) \neq 0\}| \tag{6}$$

Definition 3. *Dimensional variance ratio*

$$r_i(p, q) = \frac{|p_i - q_i|}{\bar{\delta}_i} \tag{7}$$

Formula 7 measures the deviation of point p from point q with respect to the average variance of the points in the i^{th}-dimension. It follows the outlier criteria where $\{\bar{2\delta_i}\}$ is the length of the rectangle of q. On the average, the ratio is close to 1 if p is within the proximity of q. In contrast, those with $r_i >> 1$ imply that they deviate greatly from the normal local distance in terms of dimension i. They are outliers with respect to q in dimension i. Since it has been proven in theorem 1 that an outlier in an m-dimensional space will be an outlier in at least one dimension. Formula 7 is sufficient to detect outliers w.r.t. q in any subspace which can be shown in the following theorem.

Theorem 2. *Let denote $\tau(p, q) = \max\{r_i(p, q)\}, \forall i$. If $\tau(p, q) > \kappa$, for some large κ, then p is an outlier to q.*

Proof. We can consider that $\{\bar{\delta}_i\}$ as the normalizing constants for all points in region S. Since S is small, we can approximately consider that the points within a rectangle R with unit length of 2 where q is its center are normal neighbors of q. Then, $\tau(p, q)$ is the distance from p to rectangle R. Since $\tau(p, q) > \kappa$, for some large κ, then p is an outlier to q according to postulate 1. □

Theorem 3. *Given a set S, a point q is an outlier in S if $\tau(p, q) > \kappa$, $\forall p \in S$.*

Proof. The result follows directly from postulate 2 and theorem 2. □

Since a point can be an outlier in some subspaces, it is natural to aggregate the dimensional variance ratios into one unified metric to represent the total deviation of point p. However, a naive aggregation of the ratios in all dimensions can lead to the problem of overlooking the outliers as discussed in subsection 2.2. If the dimensional variance ratios in the sample follow the distribution $N(1, \varepsilon)$, the total ratio can be as large as $(1 + \varepsilon)\sqrt{n}$ for normal points according to formula 8, which is quite significant when n is large. The ratio is large not because the point deviates from others but because the small dimensional variations are accumulated during the aggregation, which makes the total ratio large. Therefore, we introduce a cutoff threshold ρ_0. Only ratios that are greater than ρ_0 are aggregated in order to compute the total value.

Definition 4. *Aggregated variance ratio*

$$r(p, q) = \sqrt{\sum_i r_i^2(p, q)} \quad , \forall r_i(p, q) > \rho_0 \tag{8}$$

Property 1. *If p is an outlier with respect to q, $r(p, q) > \rho_0$.*

Proof. If p is an outlier with respect to q, there is at least one dimension i such that $r_i(p, q) > \kappa$. If we set $\rho_0 = \kappa$, $r_i(p, q) > \rho_0$. Thus, $r(p, q) > r_i(p, q)$. Since $r_i(p, q) > \rho_0$, we have $r(p, q) > \rho_0$. □

Property 2. *If p is not an outlier with respect to q, $r(p, q) = 0$.*

Proof. If p is not an outlier with respect to q, then $r_i(p, q) \leq \kappa$, $\forall i$. If we set $\rho_0 = \kappa$, $r_i(p, q) \leq \rho_0$, $\forall i$. Thus, from definition 8, we have $r(p, q) = 0$. □

According to property 1, if a point is an outlier in some subspace, its aggregated ratio should be greater than ρ_0 with respect to all points within its proximity. Therefore, we can define a score function to measure the outlierness of point p as follows:

Definition 5. *Outlier score*

$$oscore(p/S) = \min_{q \in S} r(p, q) \tag{9}$$

Formula 9 aggregates the outlier information for a point from all dimensions. Since the dimensions where p is not an outlier are excluded, we can guarantee that p is an outlier in S if its *oscore* is high. In addition, if p is an outlier in any subspace, the value of *oscore* for p must be high (theorem 3). Thus, *oscore* is sufficient to measure the outlierness of a point in any subspace.

Formula 9 defines the local degree to which a single point in the data set is considered an outlier. However, it is possible for points to appear as *a group of outliers*. In such cases, the value of *oscore* will be zero. We observe that a point in a small group C of outliers should have a large value for *oscore* large if we compute the value of *oscore* for that point without considering the points in its cluster. This fact must be true for all points in that cluster. If there exists a point q in the cluster whose *oscore* value with respect to $S - C$ is zero, the group is actually a set of normal points. The reason is that q is normal and all points that are close to q in terms of the aggregated variance ratio are also normal. Therefore, we can define a cluster of outliers as follows:

Definition 6. *Outlier cluster in a set S is a set of points C such that $oscore(p/S - C) > \rho_0, \forall p \in C$ and $r(p, q) = 0, \forall p, q \in C$.*

When the pairwise deviation between the outliers is small with respect to the average local distance in all dimensions, the outliers naturally appear as a cluster. This fact is captured by the second condition in the formula. The "outlierness" of an outlier cluster is defined in the following definition.

Definition 7. *Outlier cluster score*

$$oscore(C/S) = \min_{p \in C} oscore(p/S - C) \qquad (10)$$

Thus far, we have introduced the definitions to detect outliers which conform to the intuitive outlier criteria in section 3.1. The rectangles for points in a sample are bounded by $\{\bar{\delta}_i\}$. Definition 3 defines the ratio of deviation between any two points with respect to the average local variance in a dimension. We can interpret this as a similarity function between two points relative to the average variance in one dimension. As given in section 3.1. If a point is dissimilar to all points in at least one dimension, it is an outlier. Definitions 6 and 7 extend the concept of outlier to an outlier cluster, which provides complete information about the clusters of outliers in a data set. With definition 6, we can discover the clusters of outliers where their "outlierness" can be computed by $oscore(C/S)$. A nice feature of this approach is that we can identify which dimensions that a point is an outlier by using the dimensional ratio. This can then be used to visualize the outliers.

3.4 Clustering

As discussed above, the clusters of outliers can be detected by using the outlier score function. We can use an edge to represent the link between two points. If the aggregated variance ratio between two points is zero, there will be an edge connecting two points. A cluster is a set of connected points. When the size of a cluster grows large, we are certain that the points in the cluster are normal since a point can always find at least one point close to it in the graph. However, if the points are outliers, there will be no edge that connects the outliers with other points. Thus, the cluster will be small. We apply the clustering algorithm in [13] to cluster the dataset by using the computed aggregated variance ratio values in linear time. First, we put a point p into a stack S and create a new cluster C. Then, we take point p and put it in C. In addition, all of its neighbors which are connected to p are put into S. For each q in S, we expand C by removing q from S and adding q to C. The neighbors of q which are connected to q are then put into S. These steps are repeated until no point can be added to C. We then create a new cluster C'. These steps are repeated until S is empty. The pseudocode of the algorithm is shown in algorithm 1.

Algorithm 1 Clustering Pseudocode

```
 1: procedure CLUSTER($HashSet\ D$)
 2:     $Stack\ \ S$
 3:     $Vector\ \ clsSet$
 4:     $HashSet\ \ C$
 5:     while $D \neq \emptyset$ do
 6:         $p \leftarrow remove\ D$
 7:         $push\ p \rightarrow S$
 8:         $C \leftarrow new\ HashSet$
 9:         $add\ C \rightarrow clsSet$
10:         while $S \neq \emptyset$ do
11:             $q \leftarrow pop\ S$
12:             $add\ q \rightarrow C$
13:             for $r \in neighbors(q) \wedge r(q,r) \equiv 0$ do
14:                 $push\ r \rightarrow S$
15:                 $remove\ r\ from\ D$
16:             end for
17:         end while
18:     end while
19: end procedure
```

Theorem 4. *Let say $\{C_i\}$ is the set of clusters produced by the algorithm, C_i contains no outlier with respect to C_i, $\forall i$.*

Proof. Assuming that a point $r \in C_i$ is an outlier in C_i , we have $r(q, r) > \rho_0$, $\forall q \in C_i$ (property 1 and postulate 2). According to clustering algorithm 1 from lines 10 to 17, a neighbor r of a point q is put into C_i iff $r(q, r) = 0$, which contradicts the condition above. Therefore, C_i contains no outlier with respect to C_i. □

Theorem 4 shows that the clusters produced by algorithm 1 do not contain outliers. If a cluster C is large enough, we consider it as the set of normal points. Otherwise, we will compute the outlier cluster score for C. If the score is large, C is a cluster of outliers. Therefore, it is guaranteed that algorithm 1 returns the set of outlier clusters.

4. EXPERIMENTS

4.1 Synthetic Dataset

We create a small synthetic data set D to illustrate our outlier score function. We use a two dimensional data set so that we can validate the result of our algorithm by showing that the outliers and groups of outliers can be detected. The data consists of 3000 data points following a normal distribution $N(0, 1)$. Three individual outliers $\{p_1, p_2, p_3\}$ and a group C_1 of 10 outliers $\{q_1, \ldots, q_{10}\}$ are generated for the data set. The data set is illustrated in figure 2. First, we compute the $oscore$ for all the points in D with $\rho = 2$ and $\alpha = 0.4$. The algorithm detected 5 outliers. Our manually generated outliers appear in the top three outliers. The next two outliers are generated from the distribution. However, their score is low, which is approximately half of the scores of the manually generated outliers as shown in table 2. Next, we run the clustering algorithm based on the computed $oscore$ as described in the clustering section. The algorithm detected 9 clusters. Among them, two clusters have the score of zero. Thus, seven outlier clusters are detected. Table 3 shows the score of the outliers. As we can see, ten points in the manually generated cluster are detected and correctly grouped into a cluster. In addition, it appears to be the highest ranked outliers. A micro cluster C_2 of five outliers is also detected. Its low score is due to the fact that it is randomly generated from a normal distribution. In this example, we have shown that our algorithm can discover micro clusters. However, it should be noted our algorithm can detect clusters of any size, which makes it also suitable to detect outlier clusters for any application for which the size of the outlier clusters is large but still small relative to the size of the entire dataset.

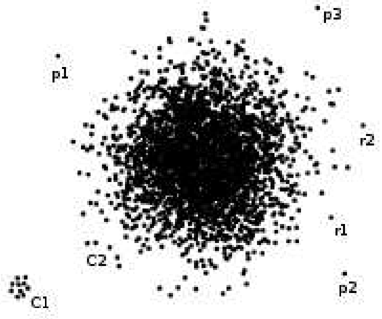


Figure 2: Points p_1, p_2, p_3 and cluster C_1 are generated outliers.

Table 2: Seven outliers are detected.

Point	Score
p_1	7.24
p_2	6.68
p_3	5.98
r_1	2.97
r_2	2.92
others	0

Table 3: Nine outlier clusters are detected in 2D dataset.

Cluster	Size	Score	Items
1	10	7.7	q_1, q_2,q_3,q_4,q_5,q_6, q_7,q_8,q_9,q_{10} (C_1)
2	1	7.24	p_1
3	1	6.68	p_2
4	1	5.98	p_3
5	1	2.98	r_1
6	1	2.92	r_2
7	5	2.43	r_3, r_4, r_5,r_6,r_7 (C_2)
8	2	0.0	r_8, r_{10}
9	1	0.0	r_{11}

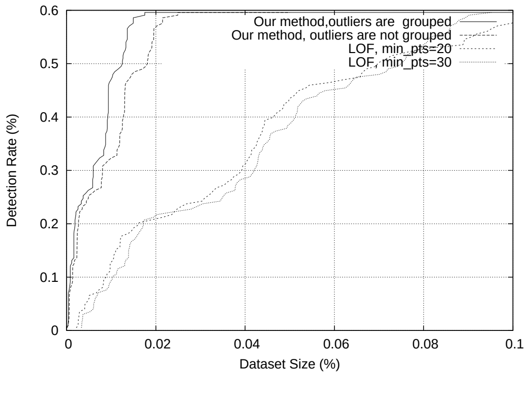


Figure 3: Detection Curve.

Table 4: Detected Attack Connections in KDD Cup Dataset.

Rank	Size	Score	Rank	Size	Score
7^{th}	1	152.6	72^{nd}	1	15.7
30^{th}	1	38.7	79^{th}	6	14.8
32^{nd}	1	34.4	80^{th}	1	14.7
36^{th}	1	32.5	111^{st}	1	11.9
37^{th}	1	32.2	113^{rd}	1	11.5
38^{th}	9	32.1	158^{th}	1	8.5
54^{th}	1	22.3	159^{th}	9	8.5
62^{th}	1	19.4	163^{th}	1	8.3

4.2 KDD Cup '99 Dataset

In this experiment, we use the KDD CUP 99 Network Connections Data Set from the UCI repository [14] to test the ability of outlier detection in detecting the attack connections without any prior knowledge about the properties of the network intrusion attacks. The detection will be simply based on the hypothesis that the attack connections may behave differently from the normal network activities which makes them outliers. The KDD CUP 99 data was compiled from a wide variety of intrusions simulated in a military network environment prepared by MIT Lincoln Labs. We create a test dataset from the KDD original dataset with 97,476 connections. Each record has 34 continuous attributes representing the statistics of a connection and its associated connection type, i.e. normal, buffer overflow attack. A very small number of attack connections are randomly selected. There are 22 types of attacks with the size varying from 2 to 16. Totally, there are 198 attack connections which account for only 0.2% of the dataset.

In this experiment, we run the LOF algorithm as a baseline to test our approach since it is the well-known outlier detection method that can detect density based outliers. First, we run LOF on the dataset with different values of min_pts from 10 to 30. The experiment with $min\_pts = 20$ has the best result. In this test, no attack is detected in the top 200 outliers. In the next set of outliers, 20 attacks are detected. The ranking of those attacks are distributed from 200 to 1000. In the top 2000 outliers, only 41 attacks are detected. We then ran our algorithm on the dataset with the same value of ρ_0 and α. Since the data set for KDD is larger than the synthetic dataset, the sample size is 100. The algorithm returns the list of outlier clusters ordered by score. The size of those clusters are small and most of them are single outlier clusters. According to the results, one attack is found in the top 10 outlier clusters and 16 attacks are found in the top 50 outlier clusters. Among them, 9 attacks are grouped into one cluster and its ranking is 38^{th}. We found that all outliers in this group are *warezmaster* attacks. Since there are only 12 *warezmaster* connections in the dataset, the clustering achieves high accuracy for this tiny cluster. *In addition, 42 attacks are found in the top 200 outliers and 94 attacks are detected in top 1000. Comparing with the results from LOF where no outliers are detected in top 200 outliers and only 20 outliers are detected in the top 1000 outliers, our algorithm yields a higher order of magnitude for accuracy.* Figure 3 shows the detection curve with respected to the number of the outliers. In this curve, we show the detection rate for LOF with $min\_pts = 20$ and

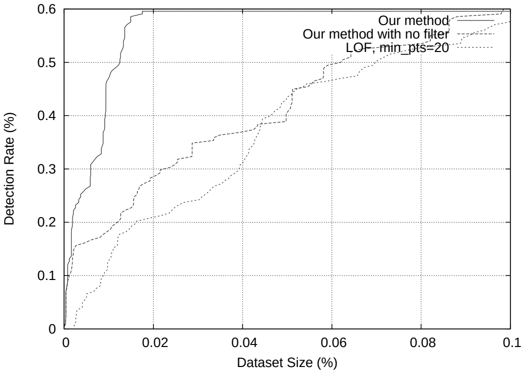


Figure 4: Detection Rate for the algorithm with and without using the filter.

$min\_pts = 30$. In addition, we show the curves for our algorithms with the ranking in terms of individual outliers and in terms of outlier clusters where the individual outlier is cluster whose size is 1. As we can see, the recall rate of our algorithm is consistently higher than that of the LOF. The recall rate of our algorithm is 60% when the size of outliers is 0.02% of the dataset, whereas that of LOF is 21%. *Given the context that outlier detection approach in general has very high false alarm rate, our method can detect a very small number of attack connections in a large dataset.*

Table 4 shows the ranking and the cluster size for the top 200 outlier clusters. According to the table, three clusters of attacks are found. The first cluster whose ranking is 38^{th} contains nine *warezmaster* attacks (recall rate = 75%). The next cluster contains six *satan* attacks (recall rate = 75%). The last cluster in the table contains 9 *neptune* attacks (recall rate = 100%). The misclassification rate for those clusters is zero. The recall rate for those attacks is very high given that each of them accounts for less than $1.2 \times 10^{-4}\%$ of the dataset.

4.3 The Effect of the Filter Parameter

The experiment above shows the result of the algorithm when the filter is applied with $\rho_0 = 2.2$. In this experiment, we want to study the effectiveness of the filter parameter on the quality of the detection rate of our method. Therefore, we ran the algorithm without the filter by setting $\rho_0 = 1$, which means the ratios in all dimensions are aggregated. Figure 4 shows the detection rate for our algorithm when $\rho_0 = 2.2$, $\rho_0 = 1$ and the detection rate for LOF with $min\_pts = 20$. According to the figure, our algorithm without using the filter parameter still consistently performs better than the LOF algorithm. The graph also shows that the algorithm can discover 27 attacks in the top 200 outliers. The better performance can be explained by the fact that the variances for all dimensions are normalized by using the dimensional ratios. However, the algorithm with the filter parameter outperforms the algorithm without the filter. In the top 200 outliers, the detection rate for the filter approach is double that of the test without the filter. The experiment shows that the filter has the effect of eliminating the noise attributes in computing outlier scores. Thus, the quality of detecting true outlier is significantly improved.

Table 5: Subspace outliers.

Point	Rank	Total score	r_i	r_i
p_7	7	152.6	$r_2 = 152.57$	$r_{29} = 2.3$
p_{36}	36	32.5	$r_1 = 32.4$	$r_{26} = 2.3$

4.4 Visualization

Theorem 3 shows that if a point is an outlier in an n-dimensional space, it must be an outlier in at least one dimension. This result implies that we can use lower dimensional spaces, i.e. 2D and 3D to visualize the outliers in order to study the significance of the outliers. We take the results of the KDD experiments to study the outliers. In addition to the ranking of the outliers, our algorithm also returns the dimensions in which a point p becomes an outlier by checking for dimensions i in which $r_i(p) > \rho_0$. Table 5 shows the dimensional score for two points p_7 and p_{36} which are *multihop* and *back* attacks respectively. In the table, p_7 is an outlier in the 2^{nd} and 29^{th} dimensions which correspond to the attribute dist_bytes and dst_host_srv_diff_host_rate, whereas p_{36} is the outlier in the 1^{st} (src_bytes) and 26^{nd} (dst_host_same_srv_rate) dimensions. Figures 5, 6 and 7 show the 2D-subspace for point p_{36} and its nearest neighbors (Chebyshev space). Figure 5 shows two dimensions in which p_{36} is not an outlier. As we can see, we can not recognize p_{36} from its neighbors. However, p_{36} appears as an outlier in the 1^{st} (src_bytes) and 26^{nd} (dst_host_same_srv_rate) dimensions as shown in figure 6. Point p_{36} is clearly distinct from its surrounding points. Figure 7 shows the distribution of p_{36}'s neighbors in this 2D-space without point p_{36}. Figures 5, 6 allow us to explain why p_{36} is not an outlier when computed by LOF. According to LOF, its score is 2.1 and it ranks 6793^{th} in the list of outliers. The score implies that its *kdist* (Euclidean space) is only twice the average of $k - dist$ of its neighbors. In Chebyshev space, $k - dist(p_{36}/k = 30)$ is 0.066 and the average $k - dist(q_i/k = 30)$ is 0.04 for $\{q_i\}$ are the 4-nearest neighbors of p. The $k - dist$ of p_{38} approximates that of its surrounding neighbors in both Euclidean and Chebyshev space. As a result, p_{36} can not be detected in the traditional approach. Whereas in our sub dimensional score aggregation approach, p_{36} is a strong outlier in the 1^{st} dimension. Thus, p_{36} can be detected.

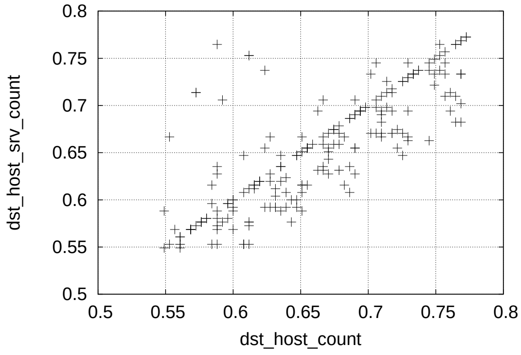


Figure 5: Point p_{36} is not an outlier in this 2d-subspace.

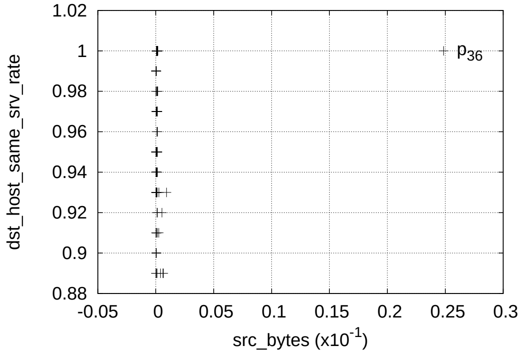


Figure 6: Point p_{36} is an outlier in this 2d-subspace.

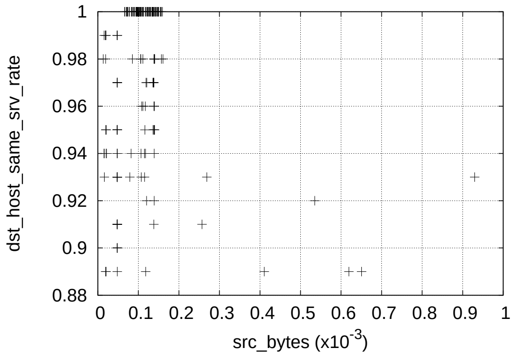


Figure 7: Point p_{36} is excluded in figure 6.

5. RELATED WORKS

Distance-based [7] and density-based [3] approaches are introduced to detect outliers in datasets. In these approaches, if the distances between a point and all its other points (distance-based) or its neighbors (density-based) are large, the point is considered an outlier. Since all dimensions are considered, the outliers in subspaces can not be detected. Recently, Papadimitriou et al [17] have introduced the use of local correlation integral to detect outliers. The advantage of the local correlation integral approach is that it can compute outliers very fast. However, similar to the approaches mentioned above, this method does not focus on subspace outlier detection.

The problem of feature selection and dimensionality reduction, e.g. PCA, have been studied extensively in classification and clustering in order to select a subset of features of which the loss function for losing some features is minimized. This approach is inappropriate for outlier detection since the outliers are rare relative to the size of dataset. The set of features that minimize the loss function may not be the features for which the points become outliers. Thus, we may not be able to detect those outliers. Another approach is to randomly select a set of features to detect the outliers [11]. Since the number of possible subspaces is large, the points may not be the outliers in the chosen subspaces and there is no guarantee that the points appearing to be outliers in the remaining subspaces can be detected.

Another work similar to the problem of subspace outlier detection is the problem of subspace clustering [15] [1] [4] which focuses on detecting clusters in the subspaces by detecting the dimensions that a set of points are dense. However, they do not show the dimensions for which a point deviates from others. In addition, their primary focus is to cluster the dataset rather than detect outliers. Therefore, they are not optimized for outlier detection.

6. CONCLUSION

In this paper, we have shown that the Chebyshev metric is superior to the Euclidean metric in detecting outliers in high dimensional data since it can overcome the curse of dimensionality by introducing the filtering threshold to remove noise during the outlier score computation. *According to the experiment, the introduction of the filter on the random deviation of a point to its neighbor has significantly boosted the performance of outlier detection. It is also the main contribution of our paper.* The property of our score function allows us to compute the outlier score in each dimension and then aggregate them to a final score. The dimensions in which a point is not an outlier is excluded from the final outlier score. Only dimensions in which p is a strong outlier are considered. Thus, noise is not detected as outliers in the dataset. In addition, this property of an outlier in Chebyshev space also allows us to visualize the outliers by drawing the graphs in the dimensions that the points deviate from others. By studying the graphs, we can eliminate the dimensions in which the outliers are not interesting to us and we can explain why the outliers are important. In this paper, we also apply the clustering technique from [13] to cluster the outliers. Two points whose oscore with respect to those points is zero are considered close and will be in the same cluster. Thus, our algorithm can also produce clusters of outliers.

7. REFERENCES

[1] C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. *SIGMOD Rec.*, 29(2):70–81, 2000.

[2] C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. In *In Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 37–46, 2001.

[3] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, 2000.

[4] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, and Philip S. Yu. On High Dimensional Projected Clustering of Data Streams. *Data Mining and Knowledge Discovery*, 10(3):251–273, Mar. 2005.

[5] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer. *SMOTEBoost: Improving prediction of the minority class in boosting*, volume 2838/2003 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, Germany, 2004.

[6] K. Das and J. Schneider. Detecting anomalous records in categorical datasets. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 220–229, New York, NY, USA, 2007. ACM.

[7] Edwin M. Knorr and Raymond T. Ng. Algorithms for

mining distance-based outliers in large datasets. In *VLDB '98: Proceedings of the 24rd International Conference on Very Large Data Bases*, pages 392–403, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[8] R. A. Jarvis and E. A. Patrick. Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers*, C-22(11):1025–1034, 1973.

[9] Y. Ke, J. Cheng, and W. Ng. Mining quantitative correlated patterns using an information-theoretic approach. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 227–236, New York, NY, USA, 2006. ACM.

[10] F. Korn, B.-U. Pagel, and C. Faloutsos. On the 'dimensionality curse' and the 'self-similarity blessing'. *IEEE Transactions on Knowledge and Data Engineering*, 13(1):96–111, 2001.

[11] A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 157–166, New York, NY, USA, 2005. ACM.

[12] H. Mannila, D. Pavlov, and P. Smyth. Prediction with local patterns using cross-entropy. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 357–361, New York, NY, USA, 1999. ACM.

[13] Minh Quoc Nguyen, Leo Mark, and Edward Omiecinski. Unusual Pattern Detection in High Dimensions. In *The Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2008.

[14] C. B. D. Newman and C. Merz. UCI repository of machine learning databases, 1998.

[15] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery*, 2005.

[16] U. Shaft and R. Ramakrishnan. Theory of nearest neighbors indexability. *ACM Trans. Database Syst.*, 31(3):814–838, 2006.

[17] Spiros Papadimitriou, Hiroyuki Kitagawa, Philip B. Gibbons, and Christos Faloutsos. LOCI: Fast outlier detection using the local correlation integral. In *Proceedings of the international conference on data engineering*, pages 315– 326. IEEE Computer Society Press, Mar. 2003.

[18] I. Steinwart, D. Hush, and C. Scovel. A classification framework for anomaly detection. *J. Mach. Learn. Res.*, 6:211–232, 2005.

[19] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos. Online outlier detection in sensor data using non-parametric models. In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, pages 187–198. VLDB Endowment, 2006.