


2019

High Dimensional Outlier Detection

Omid Khormali

Let us know how access to this document benefits you.

Follow this and additional works at: <https://scholarworks.umt.edu/etd>

 Part of the [Applied Statistics Commons](#), [Probability Commons](#), and the [Theory and Algorithms Commons](#)

Recommended Citation

Khormali, Omid, "High Dimensional Outlier Detection" (2019). *Graduate Student Theses, Dissertations, & Professional Papers*. 11377.
<https://scholarworks.umt.edu/etd/11377>

This Professional Paper is brought to you for free and open access by the Graduate School at ScholarWorks at University of Montana. It has been accepted for inclusion in Graduate Student Theses, Dissertations, & Professional Papers by an authorized administrator of ScholarWorks at University of Montana. For more information, please contact scholarworks@mso.umt.edu.

High Dimensional Outlier Detection

By

Omid Khormali

B.S., University of Tabriz, Tabriz, Iran
M.S., Tarbiat Modares University, Tehran, Iran

Professional Paper

presented in partial fulfillment of the requirements
for the degree of

Master of Science
in Data Science

The University of Montana
Missoula, MT

May 2019

Approved by:

Scott Whittenburg, Associate Dean of the Graduate School
Graduate School

Dr. Brian Sreele, Chair
Mathematical Sciences

Dr. Emily Stone
Mathematical Sciences

Dr. Javier Perez Alvaro
Mathematical Sciences

© COPYRIGHT

by

Full Legal Name

Year

All Rights Reserved

High Dimensional Outlier Detection

Chairperson: Dr. Brian Steele

In statistics and data science, outliers are data points that differ greatly from other observations in a data set. They are important attributes of the data because they can dramatically influence patterns and relationships manifested by non-outliers. It is therefore very important to detect and adequately deal with outliers. Recently, a novel algorithm, the ROMA algorithm, has been proposed [11]. In this paper, we propose a modification of the ROMA algorithm that reduces its computational complexity from $O(n^2m)$ to $O((n/(2^m - o(1)))^2m)$ where n is the number of data points and m is the dimension of the space. And as a consequence, if $\log(n) < 2^m$, then the improved complexity is $O((n/\log(n))^2m)$.

1 Introduction

A data point that is significantly different from the remaining data is an outlier. Identifying an observation as an outlier often depends on non-apparent assumptions regarding the data structure and the applied detection method [1]. Hawkins defined an outlier as an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism than the rest of the observations [14]. Barnett and Lewis indicate that an outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs [3]. Similarly, Johnson defines an outlier as an observation that appears to be inconsistent with the remainder of data [4].

Outliers can cause a serious loss of information in statistical analyses, but also may contain a useful information about unusual characteristic of the data. Thus, identification of outliers may provide useful insights, and so outlier detection has emerged as an important research area in data mining. In [1], a variety of methods for outlier detection are discussed and loosely categorized as univariate, multivariate, parametric, and non-parametric procedures. Also, it is mentioned that outlier detection methods are often based or involve on distance measures, cluster, and ideas from analysis methods. The distance-based methods are usually based on local distance measures and are appropriate for large data sets [5, 6, 7]. Another class of outlier detection methods is founded on clustering techniques, where a cluster of a few observations can be identified as a cluster of outliers [8, 9]. Another related class of methods consists of detection techniques for spatial outliers. These methods search for extreme observations relative to neighboring observations. Such outliers in total, may not otherwise be significantly different from the rest of the data set [8, 10]. It should be noted that other categorizations of outlier detection methods have been introduced recently and a large number of algorithms exist.

Recently, in [11], a new method of outlier detection based on the angles between observations points (viewed as vectors) was introduced. They presented a two-step algorithm to determining structured and unstructured outliers. The main feature of the algorithm is that it does not have any dependencies on the unknown parameters. The algorithm requires only a threshold determined by number of data points and the dimension of the observation vectors its computation. The technique proposed for removing structured outliers is also parameter-free. Once all the outliers have been identified and removed, the remaining observation vectors are used to obtain a low rank representation via a singular value decomposition of the data. In this paper, we will improve on this algorithm with respect to computational complexity.

2 Notations and the Algorithm

Suppose that we are given n observation vectors belonging to m dimensional space \mathbb{R}^m . The observation vectors are collected in a set $X = \{y_1, \dots, y_n\}$ where $y_i \in \mathbb{R}^m$ for $1 \leq i \leq n$. In this paper, we work with ℓ_2 -normed, namely $x_i = \frac{y_i}{\|y_i\|_2}$, where $\|\cdot\|_2$ denotes the ℓ_2 norm. Let $X_N = \{x_1, \dots, x_n\}$ denote the ℓ_2 -normed data set.

Let $E[Y]$ denote the expectation of a random vector Y , $var(Y)$ denote the variance, and σ_Y denote the standard deviation of observation vector Y . Let $\mathcal{N}(\mu, \sigma^2)$ denote a normal distribution with mean μ and variance σ^2 and $F_{\mathcal{N}}(\cdot)$ denote the standard normal cumulative distribution function:

$$F_{\mathcal{N}}(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^y e^{-\frac{x^2}{2}} dx.$$

In addition, *w.p.* indicates with probability and $\lfloor x \rfloor$ denotes the largest integer smaller than or equal to $x \in \mathbb{R}$. Let $\Gamma(\cdot)$ denote the gamma function and $O(\cdot)$ denotes the big O notation for complexity, and $abs(x)$ denote the absolute value of x .

Let S^{m-1} denote the unit hypersphere in \mathbb{R}^m ; i.e. $S^{m-1} = \{x | x \in \mathbb{R}^m, \|x\|_2 = 1\}$. Note that $X_N \in S^{m-1}$ for X_N as defined above.

Let \mathcal{I} denote the index set of inliers and \mathcal{O} denote the index set of outliers, for a given X_N . Then

$$\mathcal{I} = \{i | x_i \in X_N \text{ is an inlier}\}$$

and

$$\mathcal{O} = \{i | x_i \in X_N \text{ is an outlier}\}.$$

Hence the set X_N can be partitioned as $X_N = X_{\mathcal{I}} \cup X_{\mathcal{O}}$, where $X_{\mathcal{I}}$ are the set of inlier points and $X_{\mathcal{O}}$ are the set of outlier points. The parameter γ is the ratio of number of outliers to the total number of data points, and it is unknown. Let $n_{\mathcal{I}} = |\mathcal{I}| = (1 - \gamma)n$ and $n_{\mathcal{O}} = |\mathcal{O}| = \gamma n$ where $|\cdot|$ denotes the cardinality of a set.

In the following we mention essential definitions and an assumption from [11].

Definition 2.1. Let θ_{ij} denote the principal angle between two data points x_i and x_j , i.e.,

$$\theta_{ij} = \cos^{-1}(x_i^T x_j),$$

and $\theta_{ij} \in [0, \pi]$.

Definition 2.2. The acute angle between two data points x_i and x_j denoted by ϕ_{ij} is

$$\phi_{ij} = \cos^{-1}(|x_i^T x_j|) = \begin{cases} \theta_{ij} & \text{if } \theta_{ij} \leq \frac{\pi}{2} \\ \pi - \theta_{ij} & \text{if } \theta_{ij} > \frac{\pi}{2} \end{cases},$$

and $\phi_{ij} \in [0, \frac{\pi}{2}]$.

Definition 2.3. For all $i \in \{1, \dots, n\}$, the minimum angle subtended by data point x_i is

$$a_i = \min_{j=1,2,\dots,n,j \neq i} \phi_{ij}.$$

Definition 2.4. The number of acute angles formed by an observation vector x_i larger angel than the threshold ζ is

$$na_i^\zeta = |\{\phi_{ij} | \phi_{ij} > \zeta, j = 1, 2, \dots, n\}|.$$

Definition 2.5 (Outlier Identification Property, $OIP(\alpha)$). An algorithm for outlier removal is said to have Outlier Identification Property $OIP(\alpha)$, when the outlier index set estimate of the algorithm contains all the true outlier indices i.e. $\mathcal{O} \subseteq \hat{\mathcal{O}}$, where $\hat{\mathcal{O}}$ is the estimated index set for outliers, with a probability at least $1 - \alpha$.

Definition 2.6 (Exact recovery Property, $ERP(\alpha)$). An algorithm for outlier removal is said to have Exact Recovery Property, $ERP(\alpha)$ when it recovers all the inlier points or $\mathcal{I} = \hat{\mathcal{I}}$, where $\hat{\mathcal{I}}$ is the estimated index set for inliers, with a probability at least $1 - \alpha$.

Note that $ERP(\alpha)$ is a stronger condition than $OIP(\alpha)$ because if an algorithm has $ERP(\alpha)$, then it also has $OIP(\alpha)$. And in this case, $\mathcal{O} = \hat{\mathcal{O}}$ with a probability at least $1 - \alpha$.

Assumption 1. The subspace \mathcal{U} is chosen uniformly at random from the set of all r dimensional subspaces and the normalized inlier points are sampled uniformly at random from the intersection of \mathcal{U} and S^{m-1} . The normalized outlier points are sampled uniformly at random from S^{m-1} .

Assumption 2. The normalized structured outlier set is a subset of points sampled from points distributed uniformly on S^{m-1} such that the maximum principal angle in the outlier set is bounded between $[\theta_{min}^{\mathcal{O}}, \theta_{max}^{\mathcal{O}}]$ where $\theta_{max}^{\mathcal{O}} < \frac{\pi}{2}$. It can be defined as

$$X_{\mathcal{O}} = \{x_1, x_2, \dots, x_{n_{\mathcal{O}}} | x_i \in S^{m-1} \forall i, \theta_{ij} \in [\theta_{min}^{\mathcal{O}}, \theta_{max}^{\mathcal{O}}] \forall i, j \in \mathcal{O}, i \neq j\}.$$

As in [11] is mentioned, for unstructured outliers, the outlier angles are distributed around $\frac{\pi}{2}$ and lie between $[0, \pi]$, but here a structure causes the angles to be lie in the interval

$[\theta_{min}^{\mathcal{O}}, \theta_{max}^{\mathcal{O}}]$ with the mean angle being less than $\frac{\pi}{2}$. The outlier generating mechanism may be anything that can generate such an outlier set. As the outliers become more clustered $\theta_{max}^{\mathcal{O}}$ reduces and $\theta_{min}^{\mathcal{O}} \rightarrow 0$.

In [12], it is proved that two high dimensional points are almost always orthogonal to each other. And this is what the authors used in [11] motivate in their algorithm and it works on the principle (by Assumption 1) that outlier points subtend larger angles (close to $\frac{\pi}{2}$) inliers, but inlier points, since they lie in a smaller dimensional subspace, subtend smaller angles with other inlier points and hence would have a smaller score a_i as compared to an outlier.

The algorithm in [11] is

Step 1: The Removal of Outlier using Minimum Angle (ROMA) algorithm

Input: The set observation vectors $X = \{y_1, \dots, y_n\}$ where $y_i \in \mathbb{R}^m$ for $1 \leq i \leq n$

Procedure:

1. Construct $m \times n$ matrix X_N , with columns $x_i = \frac{m_i}{\|m_i\|_2}$
2. Calculate ϕ_{ij} for $i, j = 1, 2, \dots, n$
3. Determine the threshold, $\zeta = \frac{\pi}{2} - \frac{C_n}{\sqrt{m-2}}$, where $C_n = F_N^{-1}(1 - \frac{1}{2n^2(n-1)})$.
4. Calculate a_i for $i = 1, 2, \dots, n$.
5. Calculate the outlier index set as $\hat{O} = \{i | a_i > \zeta\}$, and inlier index set as $\hat{I} = \{i | a_i \leq \zeta\}$.

Output: \hat{I}, \hat{O}

The second step of the algorithm is based on Assumption 2, and it is

Step 2: ROMA with number of angles greater than a threshold ζ

Procedure:

1. Calculate $na_i^\zeta, \forall i \in \hat{I}$.
2. Set $i^* = \underset{i, j \in \hat{I}, i \neq j}{\operatorname{argmin}} \phi_{ij}$.
3. Set $o^* = \underset{j \in \hat{I}}{\operatorname{argmax}} \phi_{i^*j}$.
4. Set $\hat{O}_{op} = \{i \in \hat{I} | \operatorname{abs}(na_i^\zeta - na_{i^*}^\zeta) > \operatorname{abs}(na_i^\zeta - na_{o^*}^\zeta)\}$.

5. Set $\hat{I}_{op} = \{i \in \hat{I} | \text{abs}(na_i^\zeta - na_{i^*}^\zeta) \leq \text{abs}(na_i^\zeta - na_{o^*}^\zeta)\}$.

Output: $\hat{I}_{op}, \hat{O}_{op}$

The algorithm in [11] focused on removing the set of outliers from the data set or finding \mathcal{O} without the knowledge of both the parameters γ and r .

In [11], the theoretical analysis of the algorithm and its guarantee to capture the outliers are stated under Assumptions 1 and 2. In the following we mention some those results and we refer the reader to [11] for additional results.

Lemma 2.7 ([12]). *Let $x_1, x_2, \dots \in S^{m-1}$ be random points independently chosen with uniform distribution in S^{m-1} , and let θ_{ij} be defined in Definition 2.1. Then, the pdf of θ_{ij} is given by:*

$$h(\theta) = \frac{1}{\sqrt{\pi}} \frac{\Gamma(\frac{m}{2})}{\Gamma(\frac{m-1}{2})} (\sin(\theta))^{m-2},$$

for $\theta \in [0, \pi]$.

Remark 2.8 ([12]). *$h(\theta)$ can be approximated by the pdf of normal distribution with mean $\frac{\pi}{2}$ and variance $\frac{1}{m-2}$ for higher dimensions, say, for $m \geq 5$. In fact θ_{ij} converges weakly in distribution to $\mathcal{N}(\frac{\pi}{2}, \frac{1}{m-2})$ as $m \rightarrow \infty$.*

Lemma 2.9 ([11]). *Let $U \sim \mathcal{N}(\mu, \sigma^2)$ be a random variable V defined by*

$$V = \begin{cases} U & \text{for } U \leq \mu \\ 2\mu - U & \text{for } U > \mu \end{cases}.$$

The expectation and variance of V are given by $E(V) = \mu - \sqrt{\frac{2}{\pi}}\sigma$ and $\text{var}(V) = \sigma^2(1 - \frac{2}{\pi})$. Also $V > \mu - c\sigma$ w.p. $2F_{\mathcal{N}}(c) - 1$.

Corollary 2.10 ([11]). *Because of the density of θ_{ij} and its normal distribution approximation, when x_i, x_j are two points chosen uniformly at random from S^{m-1} , we have $\mathbb{E}(\phi_{ij}) \approx \frac{\pi}{2} - \sqrt{\frac{2}{\pi(m-2)}}$, $\text{var}(\phi_{ij}) \approx \frac{1-\frac{2}{\pi}}{m-2}$, and $\phi_{ij} > \frac{\pi}{2} - \frac{c}{\sqrt{m-2}}$ with probability $2F_{\mathcal{N}}(c) - 1$.*

Theorem 2.11 ([11]). *The algorithm that classifies x_i as an outlier when $a_i > \zeta$, identifies all the outliers with probability at least $1 - \frac{1}{n}$ gives that $\zeta = \frac{\pi}{2} - \frac{C_n}{\sqrt{m-2}}$, where $C_n = F_N^{-1}(1 - \frac{1}{2n^2(n-1)})$.*

The ROMA algorithm is a simple to implement algorithm and the main complexity lies in computing all the angles. This requires computation of $\frac{n(n-1)}{2}$ angles as the inner product of two m dimensional vectors and hence the complexity is $O(n^2m)$. In the next section we reduce the complexity of this algorithm.

3 Reducing the complexity

As it is shown in last section, the primary computational effort of the ROMA algorithm lies in computing the angles. Our idea for improving the complexity is to partition the data set and running the ROMA algorithm in each subset.

We assume that $n \gg 2^m$, and that $\alpha > 0$ satisfies $\frac{1}{\alpha} > 2^{m+1}$. Note that we can consider αn as a small possible number of observation vectors that we desire to run the ROMA algorithm on them. Our partition is constructed by slicing the m -dimensional space according to quadrants. The axes of a m -dimensional Cartesian system divide the m -dimensional space into 2^m infinite regions, called quadrants Q_i where $1 \leq i \leq 2^m$, each bounded by m half-axes. The quadrant of a observational vector can be identified according to the signs of coordinates of the vector, in the following way; we define a sign function $S : \mathbb{R}^m \rightarrow B = \{(s_1, \dots, s_m) | s_i \in \{-1, +1\}\}$ such that $S(x_i) = (sgn(x_{1i}), sgn(x_{2i}), \dots, sgn(x_{mi}))$ where $x_i = (x_{1i}, \dots, x_{mi})$, $1 \leq i \leq n$, and $sgn(y) = -1$ if $y < 0$, and $+1$, if $y \geq 0$. Since each binary vector in $\{-1, +1\}^m$ represent a quadrant, the quadrant of observation vectors x_i are identified by $S(x_i)$. In the following, we propose the partition algorithm.

Step 1: Partition-ROMA algorithm

Input: The set observation vectors $X = \{y_1, \dots, y_n\}$ where $y_i \in \mathbb{R}^m$ for $1 \leq i \leq n$

Procedure:

1. Define X_N , with $x_i = \frac{y_i}{\|y_i\|_2}$.
2. Center observation vectors at origin by computing $X_c = \{x_i - \mu | x_i \in X_N\}$ (or $X_c = \{x_i - med | x_i \in X_N\}$) where μ is the m -dimensional mean vector and med is the

m -dimensional median vector.

3. Find $Ind_i = \{j \mid \text{the quadrant } Q_i \text{ containing } x_j\}$.
4. Find the subsets of X_N due to Ind_i , i.e. $X_{N,i} = \{x_j \mid x_j \in X_N, j \in Ind_i\}$ for $1 \leq i \leq 2^m$ and $1 \leq j \leq n$.
5. Run the ROMA algorithm on each set of observation vectors $X_{N,i}$ and record the outlier as \mathcal{O}_{1,Q_i} for $1 \leq i \leq 2^m$, the set of outliers contained in quadrant Q_i .
6. Rotate the quadrants by 45 degree and repeat steps 4 and 5
7. Run again the ROMA algorithm on the data points of new $X_{N,i}$ and record the outliers as \mathcal{O}_{2,Q_i} for $1 \leq i \leq 2^m$

Output: $\cup_{i=1}^{2^m} (\mathcal{O}_{1,Q_i} \cap \mathcal{O}_{2,Q_i})$

In the following we mention to some theoretical results about the algorithm.

For $1 \leq i \leq n$ and $1 \leq j \leq 2^m$, define

$$u_{ij} = \begin{cases} 1 & \text{if } x_i \text{ in } Q_j \\ 0 & \text{Otherwise} \end{cases} .$$

Suppose x_i is randomly selected from X_N implies that $u_{ij} \sim \text{Bernolli}(p)$ where $p = \frac{1}{2^m}$ because there are 2^m quadrants.

Now suppose $U_j = \sum_{i=1}^n u_{ij}$ counts the number of points in quadrant Q_j . Then, $U_j \sim \text{Binom}(n, p)$.

Theorem 3.1 ([13]). *Let $X \sim \text{Binom}(n, p)$ be a binomial random variable with parameters p and n . For $K \geq np$, the following inequality holds:*

$$Pr(X \leq K) > 1 - \frac{e^{-nD(p, k/n)}}{\max\{2, \sqrt{4\pi nD(p, k/n)}\}}$$

where $D(p, c) = c \cdot \ln(c/p) + (1 - c) \cdot \ln((1 - c)/(1 - p))$.

For using the probability bound in Theorem 3.1, note that $e^{-nD(p, c)} = (\frac{c}{p})^{-nc} (\frac{1-c}{1-p})^{-n(1-c)}$.

Theorem 3.2. For a $\epsilon > 0$ and large enough n , the number of centered observation vectors in X_c in each quadrant is at most $\frac{n}{2^m - \epsilon}$.

Proof. We show $Pr(U_j < \frac{n}{2^m - \epsilon}) \approx 1$ for large enough n . By Theorem 3.1, we have

$$\begin{aligned} Pr(U_j < \frac{n}{2^m - \epsilon}) &> 1 - \frac{e^{-nD(1/2^m, 1/(2^m - \epsilon))}}{\max\{2, \sqrt{4\pi nD(1/2^m, 1/(2^m - \epsilon))}\}} = \\ &1 - \frac{(\frac{2^m}{2^m - \epsilon})^{-\frac{n}{2^m - \epsilon}} (\frac{2^m - \epsilon - 1}{2^m - 1})^{-n(1 - \frac{1}{2^m - \epsilon})}}{\max\{2, \sqrt{4\pi nD(1/2^m, 1/(2^m - \epsilon))}\}} \approx 1 \end{aligned}$$

for large enough n . Then the desired result holds. \square

In the following we state the Chernoff-Hoeffding Theorem.

Theorem 3.3 ([14]). Let X_1, \dots, X_n be independent binary random variables and let a_1, \dots, a_n be coefficients in $[0, 1]$. Let $X = \sum_i a_i X_i$. Then

1. For any $\mu \geq E[X]$ and any $\delta > 0$, $Pr[X > (1 + \delta)\mu] \leq \left(\frac{e^\delta}{(1 + \delta)^{(1 + \delta)}}\right)^\mu$.
2. For any $\mu \leq E[X]$ and any $\delta > 0$, $Pr[X < (1 - \delta)\mu] \leq e^{-\mu\delta^2/2}$.

By using Theorem 3.3, we have the following result.

Theorem 3.4. For a $\alpha > 0$, $\frac{1}{\alpha} > 2^{m+1}$ and large enough n , the number of observation vectors in X_c in each quadrant is at least αn .

Proof. We show $Pr(U_j < \alpha n) \approx 0$. By Theorem 3.3 part 2 and taking all $a_i = 1$, we have

$$Pr(U_j < \alpha n) = Pr(U_j < (1 - \delta)\mu) \leq e^{-\mu\delta^2/2}$$

where $\mu = E[U_j] = \frac{n}{2^m}$, and $\delta = 1 - \alpha 2^m$. Note that since $\frac{1}{\alpha} > 2^{m+1}$, $\frac{1}{2} < \delta < 1$. Then

$$e^{-\mu\delta^2/2} < e^{-\frac{n}{2^m \times 8}}.$$

For sufficiently large n , $Pr(U_j < \alpha n) \approx 0$ and the desired result is obtained. \square

Therefore we can find the complexity our algorithm.

Corollary 3.5. Under the assumptions stated above, the computational complexity of Partition-ROMA algorithm is $O((\frac{n}{2^m - o(1)})^2 m)$.

Proof. By Theorem 3.2 and Theorem 3.4, $\alpha n \leq |X_{N,i}| \leq \frac{n}{2^{m-\epsilon}}$. Clearly the complexity is $O((\frac{n}{2^{m-o(1)}})^2 m)$. \square

Observation 3.6. *Corollary 3.5 implies that if $\log(n) < 2^m$, then the computational complexity of Partition-ROMA algorithm is $O((\frac{n}{\log(n)})^2 m)$.*

In the Partition-ROMA algorithm, the rotation of quadrants are discussed. The rotation can be done by using the unit standard bases of \mathbb{R}^m and the usual rotation techniques. To elucidate, suppose that the axes are represented with the unit standard basis of the space, i.e. $B = \{e_1, e_2, \dots, e_m\}$ where that i^{th} entry of e_i is 1 and the other entries are 0.

So we rotate the e_i 's by 45 degree by the rotation matrix R, we use either

1. If m is even:

$$R = \begin{bmatrix} \cos(45) & \sin(45) & 0 & 0 & \dots & 0 & 0 \\ -\sin(45) & \cos(45) & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \cos(45) & \sin(45) & 0 & \dots & 0 \\ 0 & 0 & -\sin(45) & \cos(45) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \cos(45) & \sin(45) \\ 0 & 0 & 0 & 0 & \dots & -\sin(45) & \cos(45) \end{bmatrix}$$

2. If m is odd:

$$R = \begin{bmatrix} \cos^2(45) & \sin(45) & 0 & 0 & \dots & 0 & 0 & \cos(45)\sin(45) \\ -\cos(45)\sin(45) & \cos(45) & 0 & 0 & \dots & 0 & 0 & -\sin^2(45) \\ 0 & 0 & \cos(45) & \sin(45) & 0 & \dots & 0 & 0 \\ 0 & 0 & -\sin(45) & \cos(45) & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & \cos(45) & \sin(45) & 0 \\ 0 & 0 & 0 & \dots & 0 & -\sin(45) & \cos(45) & 0 \\ -\sin(45) & 0 & 0 & \dots & 0 & 0 & 0 & \cos(45) \end{bmatrix}$$

We show that the Partition-ROMA algorithm can detect the inlier data points with probability $1 - \beta$. Then this gives us the guarantee for detecting the outliers by the Partition-ROMA algorithm.

Lemma 3.7 ([11]). *Under Assumption 1, $Pr(\hat{\mathcal{I}} = \mathcal{I}) \geq 1 - n_{\mathcal{I}}Pr(a_{i,i \in \mathcal{I}} > \zeta)$. Hence ROMA has the property of ERP($n_{\mathcal{I}}Pr(a_{i,i \in \mathcal{I}} > \zeta)$).*

Theorem 3.8. *For uniformly distributed data, $Pr(\hat{\mathcal{I}} = \mathcal{I}) \geq 1 - \beta$, where $\beta = \sum_{j=1}^{2^m} n_{\mathcal{I}}Pr(a_{i,i \in \mathcal{I}} > \zeta)$. Hence, partition-ROMA has the property of ERP(β).*

Proof. Suppose n is the number of data points and m is the dimension of the space. We have

$$Pr(\hat{\mathcal{I}} = \mathcal{I}) = Pr(\cup_{j=1}^{2^m} (\hat{\mathcal{I}}_j = \mathcal{I}_j)) = \sum_{j=1}^{2^m} Pr(\hat{\mathcal{I}}_j = \mathcal{I}_j).$$

Suppose s_i are the number of data points in the i th quadrant. Then by Lemma 3.7, we have

$$\sum_{j=1}^{2^m} Pr(\hat{\mathcal{I}}_j = \mathcal{I}_j) \geq \sum_{j=1}^{2^m} \left(\frac{s_i}{n} - n_{\mathcal{I}}Pr(a_{i,i \in \mathcal{I}} > \zeta) \right) = 1 - \sum_{j=1}^{2^m} n_{\mathcal{I}}Pr(a_{i,i \in \mathcal{I}} > \zeta)$$

Then the desired result holds by taking $\beta = \sum_{j=1}^{2^m} n_{\mathcal{I}}Pr(a_{i,i \in \mathcal{I}} > \zeta)$. □

4 Numerical example

We generate the data randomly with multivariate normal distribution and test outlier detection of the ROMA and our Partition-ROMA algorithm.

We considered $m = 6$ and $n = 1000$, and generated 950 many 6-dimensional observation vectors from the $\mathcal{N}(\mu = 20, \sigma^2 = (0.1)^2)$ (by `np.random.normal(location = 20, scale = 0.1, size = 6)`), and 50 many 6-dimensional observation vectors from the $\mathcal{N}(\mu = 0, \sigma^2 = (5)^2)$ (by `np.random.normal(0, 5, 6)`) in Python. In addition, the 50 observation vectors that created from the $\mathcal{N}(\mu = 0, \sigma^2 = (5)^2)$ were considered as outliers. We simulated 10 times and the results are tabled below:

Algorithms	# Out.	# Out.	# Out.	# Out.	# Out.	# Out.	# Out.	# Out.	# Out.	# Out.
ROMA	12	6	3	10	11	5	7	8	6	4
Partition-ROMA	9	5	26	22	33	27	12	28	11	26

Note that all estimated outliers by both algorithms in simulations are in the outlier set of 50 observation vectors from `np.random.normal(0, 5, 6)`.

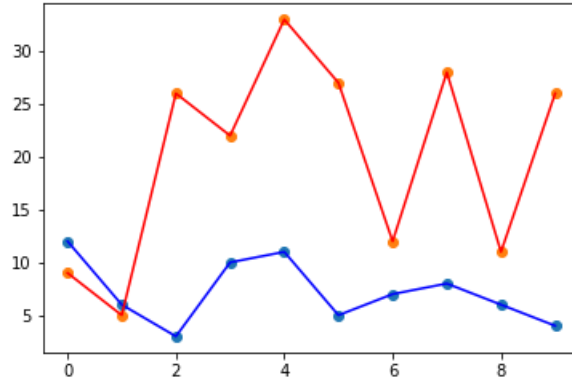


Figure 1: The blue color graph is related to ROMA algorithm , and red color graph is for the Partition-ROMA algorithm.

5 Conclusion

The proposed algorithm, Partition-ROMA algorithm, can improve the complexity of ROMA-algorithm from $O(n^2m)$ to $O((n/(2^m - o(1)))^2m)$ where n is the number of data points and m is the dimension of the space. And, as a consequence, if $\log(n) < 2^m$, then the improved complexity is $O((n/\log(n))^2m)$. Since Partition-ROMA algorithm is based on ROMA-algorithm, its performance is depends of the performance of ROMA-algorithm which was analyzed both theoretically and numerically in [11].

References

- [1] I. Ben-Gal, *Outlier detection*, In: Maimon O. and Rockach L. (Eds.) *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*, Kluwer Academic Publishers, 2005.
- [2] D. Hawkins, *Identification of Outliers*, Chapman and Hall, 1980.
- [3] V. Barnett and T. Lewis, *Outliers in Statistical Data*, John Wiley, 1994.
- [4] R. Johnson, *Applied Multivariate Statistical Analysis*, Prentice Hall, 1992.
- [5] E. Knorr and R. Ng, *A unified approach for mining outliers*, In *Proceedings Knowledge Discovery KDD*, 219-222, 1997.
- [6] E. Knorr and R. Ng, *Algorithms for mining distance-based outliers in large datasets*, In *Proc. 24th Int. Conf. Very Large Data Bases (VLDB)*, 392-403, 24-27, 1998.
- [7] S.D. Bay and M. Schwabacher, *Mining distance-based outliers in near linear time with randomization and a simple pruning rule*, In *Proc. of the ninth ACM-SIGKDD Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, 2003.
- [8] R. Ng and J. Han, *Efficient and Effective Clustering Methods for Spatial Data Mining*, In *Proceedings of Very Large Data Bases Conference*, 144-155, 1994.
- [9] E. Acuna and C. A. Rodriguez, *Meta analysis study of outlier detection methods in classification*, Technical paper, Department of Mathematics, University of Puerto Rico at Mayaguez, In *proceedings IPSI 2004*, Venice, 2004.
- [10] S. Shekhar, C. T. Lu and P. Zhang, *Detecting Graph-Based Spatial Outlier*, *Intelligent Data Analysis: An International Journal*, 6(5), (2002), 451-468.
- [11] V. Menon and S. Kalyani, *Structured and Unstructured Outlier Identification for Robust PCA: A Non iterative, Parameter free Algorithm*, arXiv:1809.04445v1
- [12] T. Cai, J. Fan, and T. Jiang, *Distributions of angles in random packing on spheres*, *J. Mach. Learning Res.*, **14**(1), (2013), 1837-1864.
- [13] M. Short, *Improved Inequalities for the Poisson and Binomial Distribution and Upper Tail Quantile Functions*, *ISRN Probability and Statistics*, Volume **2013**, Article ID 412958, 6 pages.
- [14] W. Hoeffding, *Probability Inequalities for Sums of Bounded Random Variables*, *Journal of the American Statistical Association*, **58**(301), (1963), 13-30.