A Novel Outlier Detection Method for Multivariate Data

Yahya Almardeny, Noureddine Boujnah, and Frances Cleary

Abstract—Detecting anomalous objects from given data has a broad range of real-world applications. Although there is a rich number of outlier detection algorithms, most of them involve hidden assumptions and restrictions. This paper proposes a novel, yet effective outlier learning algorithm that is based on decomposing the full attributes space into different combinations of subspaces, in which the 3D-vectors, representing the data points per 3D-subspace, are rotated about the geometric median, using Rodrigues rotation formula, to construct the overall outlying score. The proposed approach is parameter-free, requires no distribution assumptions and easy to implement. Extensive experimental study and comparison are conducted on both synthetic and real-world datasets with six popular outlier detection algorithms, each from different category. The comparison is evaluated based on the precision *@s*, average precision, rank power, AUC ROC and time complexity metrics. The results show that the performance of the proposed method is competitive and promising.

Index Terms—Rotation Based Outliers, Outlier Detection, Data Mining, Multivariate Data.

1 Introduction

With the vast development in emerging technologies such as artificial intelligence, medical field advancements and IoT, more data has been available in the market. The latter led to embrace more data-driven decisions for drawing accurate conclusions in major industries. Hence, it is of great interest in a variety of real-world applications to recognize and isolate data that has abnormal or exceptional behavior which often manifests interesting facts, such as in fraud discovery, image processing, signal analysis, network intrusion, measurement errors detection in data derived from sensors, and machine learning modeling, to name a few [1], [2], [3], [4].

Barnett and Lewis defined an outlier as "an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data" [5]. In the statistics literature and data mining, an outlier is also referred to as abnormality, anomaly, discordant, or deviant [6].

Outlier detection can be defined as the process of identifying rare and suspicious observations that differ significantly from the majority of data [6].

Technically, the procedure of detecting outliers consists of two main steps: 1) giving outlying score to the data points and 2) determining outliers by ranking them based on some metrics.

The most common outlier detection models in the literature can be approximately classified into statistic-based, distance-based, density-based, clustering-based, deviation-based, high-dimensional approaches and recently the machine learning-based type of methods [1], [4], [6], [7].

Several classical outlier detectors are based on the sample mean and covariance matrix in general, but they do not always yield better results, as they themselves are affected by outliers [8].

In this work, we propose a novel, yet effective learning algorithm for outlier detection in multivariate data where the number of attributes is greater than or equal 3. The core work of the proposed algorithm is that the full attributes space is decomposed into different combinations of subspaces in which the 3D-vectors, representing the data points per 3D-subspace, are rotated about the geometric median two times counterclockwise using Rodrigues rotation formula. The results of the rotation are parallelepipeds where their volumes are mathematically analyzed as cost functions and used to calculate the Median Absolute Deviations to obtain the outlying scores for each 3D-subdimension. Subsequently, the outlying scores of the full space are reconstructed by taking the average of the outlying scores of all 3D-subspaces. Finally, all observations are ranked in a descending order according to their scores and the top s observations with highest scores are considered as promising candidates of outliers. It should

be pointed out that our proposed Rotation-based Outlier Detection approach – ROD, is parameter-free, requires no statistical distribution assumptions and is intuitive in three-dimensional space.

This paper is organized as follows: In section 2 we briefly review existing literature and research work. Section 3 demonstrates the proposed algorithm geometrically in 3D, analyses it mathematically and suggests an optimization approach. Section 4 extends the method into higher dimensions. Section 5 provides a discussion on efficiency after performing the experimental study on synthetic and real-world datasets, and Section 6 concludes this paper and suggests future work.

2 Related Work

Over the years, many outlier detection methods have been introduced in different research communities. In this section, we briefly talk about the latest works since plenty of good survey papers have been published in this field (see [6], [7]).

There are different modes of outlier detection techniques that can be roughly categorized into supervised, semi-supervised and unsupervised approaches [6]. Supervised outlier detection requires pre-labelled data, while semi-supervised needs pre-classified data but only learns data marked normal. For the unsupervised methods, the class label of data is not required which makes it the most popular.

Outlier detection approaches can be roughly classified into statistic-based, distance-based, density-based, ensemble-based, machine learning-based (a.k.a deep learning-based), the high-dimensional approaches, and so on [1], [4], [6], [7].

The statistical-based algorithms assume that an outlier is a point that has a low generation probability by some distribution in the dataset. However, this distribution assumption does not always hold true. Besides, it lacks robustness because mean and standard deviation are sensitive to extreme values [9], [10]. Efforts have been put to minimize the influence of outliers on the methodology such as the Minimum Covariance Determinant (MCD) estimator which is one of the first affine equivariant and highly robust estimators of multivariate data [11].

The general model of the distance-based methods works by taking the k-Nearest Neighbor distance of a point as an outlier score. This approach is distribution-free but assumes that normal data objects have a certain dense of neighborhood. [12].

The density-based methods work by comparing the relative density around a point with the density around its local neighbors. Local Outlier Factor (LOF) is a very popular density-based technique that computes the ratio between local density of a point and the local density of its nearest neighbors. A point is considered as an outlier if its LOF value is high [13]. This type is more effective than distance-based methods, but is not very effective in high dimensions due to the degradation of the accuracy of the density estimation process [1], [6].

The general idea of the ensemble-based approaches is to use meta-algorithms in which their outputs are combined and used for outlier analysis. Many techniques have been proposed, including Feature bagging [14] and Isolation Forest [15]. The Feature Bagging framework consolidates results from several outlier detection algorithms where each detector randomly selects a subset of the original features. Whereas the Isolation Forest (iForest) builds an ensemble of trees and identifies an outlier as an instance that has short path length on the trees (i.e. easily partitioned). However, iForest fails to detect local outliers when several clusters of normal instances exist in the dataset; that is because normal clusters of similar density mask local outliers so they become less susceptible to isolation [16].

The Angle-Based Outlier Degree (ABOD) is a popular, robust and parameter-free algorithm for high-dimensional data [17]. It measures the variance of the angle spectrum of the data points weighted by the corresponding distances. Yet, it is not distribution-free and it has a very high computational cost $\mathcal{O}(n^3)$. For a better performance, Pham and Pagh in 2012 proposed a fast approximation algorithm called FastVOA [18], that has a near-linear complexity and based on random sampling for mining top s outliers.

Moreover, since outliers may be visible only in subspaces of the original data space, Kriegel in 2009 created a Subspace Outlier Degree (SOD) model which assumes that a set of nearest neighbors of an outlier has a lower-dimensional projection with small variance [19]. Like most of kNN-based algorithms, choosing appropriate number k is a very important factor that affects SOD performance [20].

Into the bargain, several deep learning-based approaches have been proposed to solve outlier detection problems [21]. Fully connected AutoEncoder is a neural network that works by varying on the connectivity architecture randomly where it uses reconstruction error as an outlier score [22]. AutoEncoder is capable of avoiding overfitting and achieving robustness because of its ensemble-centric approach. However, it suffers from a high time complexity that is found in neural networks in general [22].

3 Outlier Detection with Vector Rotation

3.1 Motivation

In view of the fact that the volume of a parallelepiped is the product of the base area and the height, three non-collinear vectors $\vec{a}$, $\vec{b}$ and $\vec{c} \in \mathbb{R}^3$ can form parallelepiped in which $\vec{a}$ and $\vec{b}$ constitute the base. Hence the volume can be defined as:

$$\begin{aligned} volume &= |(\vec{a} \times \vec{b}) \cdot \vec{c}| \\ &= (\|\vec{a}\|\|\vec{b}\| \sin\gamma) \cdot \|\vec{c}\| \, |\cos\theta| \end{aligned} \tag{1}$$

where $\gamma = \angle(\vec{a}, \vec{b})$ and θ is the angle between $\vec{c}$ and the height $\vec{h}$ as shown in Fig. 1.

If we draw a diagonal line, from the origin of the three vectors to the upper-right corner as shown in Fig. 1. One can imagine that vectors $\vec{b}$ and $\vec{c}$ were resulted from rotating $\vec{a}$ two times counterclockwise about that diagonal line by some angles $\theta'_1 \neq \theta'_2 > 0$. Intuitively, the volume of the parallelepiped will be proportional to $\|\vec{a}\|$ and the angle between $\vec{a}$ and the rotation-axis.

Furthermore, it is well-known that the geometric median, which can be found to generalize the median by using the appropriate L1 estimator, is insensitive to outliers since it is not skewed so much by extremely large or small values [23].

Although computing the geometric median is a computationally challenging task when $d > 2$ [23], Vardi and Zhang [24] published a modified version of Weiszfeld algorithm to find the geometric median y of a set of points $S = \{x_1, \ldots, x_m\} \in \mathbb{R}^{\mathrm{d}}$, that is *extremely simple* to program and has *very quick* convergence:

$$y \to T(y) = \Big(1 - \frac{\eta(y)}{r(y)}\Big)^{+} \widetilde{T}(y) + min\,\Big(1, \frac{\eta(y)}{r(y)}\Big) y \tag{2}$$

where:

$$\widetilde{T}(y) = \Big\{ \sum_{x_i \neq y} \frac{\eta_i}{\|y - x_i\|} \Big\}^{-1} \sum_{x_i \neq y} \frac{\eta_i x_i}{\|y - x_i\|}$$

$$r(y) = \|\widetilde{R}(y)\|, \quad \widetilde{R}(y) = \sum_{x_i \neq y} \eta_i \frac{x_i - y}{\|x_i - y\|}$$

$$\eta(y) = \begin{cases} \eta(k) & if \;\; y = x_k,\, k = 1, ..., m, \\ 0 & otherwise \end{cases}$$

In words, $y \in \mathbb{R}^{\mathrm{d}}$ is the geometric median only if it is a fixed point and $r(y) \leq \eta(y)$, where $\eta(y)$ is a weight variable at y that equals to: either 1) $\eta(k)$: the number of data vectors in S that are found to have zero Euclidean distance to y during the iterative process, then $T(y)$ is a weighted average of S; or 2) zero if $\eta(k) = 0$, then $T(y) = \widetilde{T}(y)$ as in Weiszfeld algorithm. It should be pointed out that y is unique whenever the points are not collinear and it is equivariant for Euclidean similarity transformations, including translation, rotation and reflection.

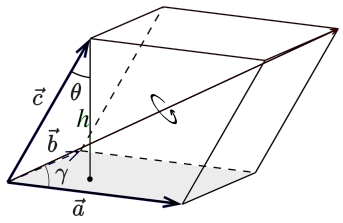


Fig. 1: Example of parallelepiped, generated by three vectors $\vec{a}$, $\vec{b}$ and $\vec{c}$

3.2 Rotation-based Outlier Detection (ROD)

Theorem 1. Let $D = \{\vec{v_1}, \vec{v_2}, ..., \vec{v_n}\} \in \mathbb{R}^3$ be a collection of vectors representing the data points of a three-dimensional dataset. If $\vec{m} \in \mathbb{R}^3$ is the unit vector of the geometric median of D, that is describing an axis of rotation; it can be proved that $\forall \vec{v} \in D$ independent from $\vec{m}$, the $signed\ volume$ of the parallelepiped formed by rotating $\vec{v}$ two times around $\vec{m}$, according to the right hand rule, by two consecutive angles $\theta_1 < \theta_2 \in (0, 2\pi)$, using Rodrigues rotation formula, can be approximated to (*hence correlated to*) a cost function, given by (*see appendix A*):

$$f(\vec{v}, \gamma) = \|\vec{v}\|^3 (\cos\gamma \sin\gamma^2) \tag{3}$$

Eq. 3 describes the differences among the vectors in the dataset with regard to their magnitudes and the angle $\gamma = \angle(\vec{v}, \vec{m})$ that reflects the degree of deviation from $\vec{m}$. We will denote $(\cos\gamma \sin\gamma^2)$ as $\overset{\star}{f}(\gamma)$, and $f(\vec{v}, \gamma)$ as $rod(v)$ interchangeably from now onwards.

From $\overset{\star}{f}(\gamma)$ and Fig. 2, one can observe the following properties:

1) Since $\overset{\star}{f}(\gamma)$ is periodic over $[0, 2\pi]$, it is more convenient to confine the study over $[0, \pi]$ because the trigonometric function calculates only for the smallest angle between $\vec{v}$ and $\vec{m}$.
2) $\overset{\star}{f}(\gamma) = 0$ whenever $\gamma \in \{0, \frac{\pi}{2}, \pi\}$; and here we have three cases:

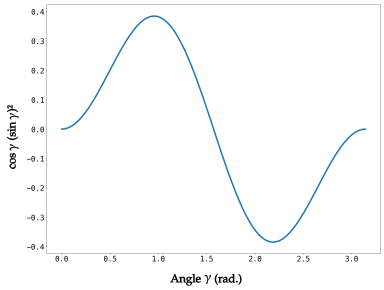


Fig. 2: $\overset{\star}{f}(\gamma)$ Graph

- $\gamma = \overset{\star}{f}(\gamma) = 0$ only if $\vec{v} = \vec{m}$,
- $\gamma = \frac{\pi}{2}$, $\overset{\star}{f}(\gamma) = 0$ only if $\vec{v} \perp \vec{m}$,
- and $\gamma = \pi$, $\overset{\star}{f}(\gamma) = 0$ only if $\vec{v} \parallel \vec{m}$.

3) It reaches its maximum value ≈ 0.385 and minimum value ≈ -0.385 at the turning points of the graph. We will name these angles the *threshold angles* α_1 and α_2 respectively (*see appendix B*).
4) $\overset{\star}{f}(\gamma \in [0, \frac{\pi}{2}]) = -\overset{\star}{f}(\gamma \in [\frac{\pi}{2}, \pi])$ which provides uniqueness to the values by applying more emphasis on the points that are not in the same hyperplane with $\vec{m}$ (i.e. the orientation of the parallelepiped).

3.3 ROD Optimization

In this section, we propose an optimization approach for the proposed algorithm.

3.3.1 Deviation Proportionality

Lemma 1. Since $\overset{\star}{f}(\gamma) \not\propto \gamma$ over $(\alpha_1, \frac{\pi}{2}] \cup (\alpha_2, \pi]$. Scaling down the angles to $(0, \alpha_1] \cup (\frac{\pi}{2}, \alpha_2]$ shall reserve a proper deviation proportionality of the data vectors. (*see appendix C).*

3.3.2 Vectors Linearity

Lemma 2. Subtracting the geometric median from every vector in the dataset shall unmask the rotation cost of any vector with suspicious small magnitude, that is also collinear with another vector that has normal to high magnitude, without ruining the relative positional relations among the data points. (*see appendix D*).

3.4 Rotation Outlier Score

In order to isolate ROD *costs* that appear to be inconsistent with the remainder of other costs, we chose the Median Absolute Deviation about the median for its robustness and efficiency properties [25]:

$$MAD = median\{|x_i - \tilde{x}|\}$$
$$M_i = \frac{0.6745(x_i - \tilde{x})}{MAD} \quad (4)$$

where $\tilde{x}$ is the median of the dataset and M_i is the ith outlierness score of each point. In order to make the estimator consistent, the constant 0.6745, introduced by Iglewicz and Hoaglin in [26], is needed since for any normal-like distribution, M_i would converge to 1 as $E(MAD) = 0.6745\sigma$ for large dataset.

4 ROD in Higher Dimensions

4.1 Motive

Some of the important motives for finding outliers in subspaces of the original features space can be summarized under the following points:

- In complex manifolds, complex outliers that are hidden in subspaces would be missed [27], [28].
- Data might be generated by different mechanisms per dimension (or subset of dimensions), thus different views of data can reveal outliers that were not seen in the full attributes space [29].
- In high dimensional space, true outliers might be masked due to the curse of dimensionality, where data points spread too thin as the dimensions increase, making data extremely noisy [30]. Besides, the concept of neighbourhood becomes meaningless [31]; and approaches that are based on finding the relative contrast between distances of the data points become unreliable [32].

4.2 Methodology

Proposition 1. Given a dataset $D \subseteq \mathbb{R}^{n \times d}$ (n samples and $d > 3$ features), ROD can be utilized in the 3D-subspaces $\{U_{ij}, ..., U_{IJ}\}$ that are resulted from decomposing the full attributes space V into sets of different combinations of subspaces $\{s_{i \in I} \mid s_i = \{U_j \mid j \in [J]\}\}$, then for every data sample, the overall outlying score of V is constructed by combining the ROD scores per 3D-subspace. (*see appendix D*)

Algorithm 1 shows the detailed steps of applying ROD for multivariate dataset where the number of features ≥ 3.

Remark 1: Dataset features with high magnitude would dominate the algorithm since the ROD cost is

Algorithm 1 ROD

Input: The data collection $D \subseteq \mathbb{R}^{n \times d}$, and the number of outlier candidates s ;
Output: The s desired outliers ;
procedure
  Scale D using Eq. 5;
  Initialize A as a list to contain all 3D-scores;
  for each $U_j \in \binom{d}{3}$ subsets of attributes, $U_j \subseteq \mathbb{R}^{n \times 3}$ **do**
    Initialize Γ as a list to contain all angles;
    Find $\vec{m}$ = geometric median using Eq. 2;
    for each observation as a vector $\vec{v} \in U_j$ **do**
      Subtract $\vec{v}$ from $\vec{m}$ to obtain $\vec{v'}$;
      Calculate the magnitude of $\vec{v'}$;
      Find $\gamma = \angle(\vec{v'}, \vec{m})$ and add it to Γ;
    Scale all angles in Γ according to 3.3.1;
    Solve for Eq. 3 using scaled angles in Γ;
    Find the score per sample using Eq. 4;
    Scale outlying scores using Eq. 6;
    Add scaled scores of current U_j to A;
  Find A = $average(A^T)$ per row (at axis 0);
  Sort scores of A in descending order and return top s observations as desired outliers;

correlated to the vector magnitude per sample. Consequently, MAD_i would vary noticeably, masking some of the 3D-subspace scores. To solve this, we scale the data according to the quantile range which is robust to outliers:

$$x_{i_{new}} = \frac{x_i - Q_1(x)}{Q_3(x) - Q_1(x)} \tag{5}$$

where x_i is the ith sample and $Q_1(x)$ and $Q_3(x)$ are the first and third quantiles of the dataset.

Remark 2: Similar to the first remark, limiting MAD_i over a predefined numerical range corrects false weights, assigned by the average function, to each 3D-score. For example, suppose we have two samples, each has two 3D-scores as follows: $s_1 = \{2, 4\}$, $s_2 = \{3, 1\} \implies average(s_1) > average(s_2)$, yet both have similar number of outliers per their 3D-subspaces as described in Lemma 7. To solve this, we squish the results over $[0, 1]$ using:

$$x_{ij_{new}} = \frac{1}{1 + e^{-x_{ij}}} \tag{6}$$

where x_{ij} is the MAD_i of the ith sample and jth 3D-subspace.

TABLE 1: Brief Description of Experimental Data

Dataset	Observations	Attributes	Outliers
SMTP	95156	3	30
Banknote	1372	4	610
Thyroid	7200	6	534
Diabetes	768	8	268
Shuttle	49097	9	3511
Seismic	2584	11	170
Digits	6870	16	156
Cardio	1655	21	176
WBC	378	30	21

5 Experimental Study

This section reports the experimental details and its results that have been conducted on the proposed algorithm to evaluate its efficiency and effectiveness.

5.1 Experimental Settings

5.1.1 Experimental Data

Two series of comparisons were carried out on two artificial datasets and 9 popular real-world datasets in which all of them have been used previously in the literature of outlier detection. Table 1 summarizes an overview about the datasets information including the name of dataset, the number of observations, the number of attributes and the number of true outliers. All datasets have been downloaded from ODDS[1], which can be found on our repository along with the code used in this paper[2].

One can see from this table that datasets vary in their sizes, dimensions and the quantity of outliers, which covers a wide range of cases for this study.

Furthermore, all datasets represent classification problems in which ODDS considers the rare class as outlier denoted as "1", whereas the remaining classes as normal observations denoted as "0".

5.1.2 Evaluation Metrics

To make comprehensive comparison, we embraced five different performance evaluation metrics, namely: Precision @s [4], Average Precision [33], Rank Power [6], Area under the ROC Curve [34], and Time Complexity.

Precision @s is extensively used in learning algorithm evaluation. It is the ratio of correctly predicted observations as outlier to the total number of outlier candidates:

$$Pr = \frac{k}{s} \tag{7}$$

where k is the number of true outliers found within s outliers candidates.

1. http://odds.cs.stonybrook.edu/
2. https://codeocean.com/capsule/2686787/tree

Average Precision (AP), on the other hand, summarizes a precision-recall curve as the weighted mean of precisions achieved at all possible thresholds:

$$AP = \sum_{n} (R_n - R_{n-1}) Pr_n \tag{8}$$

where Pr_n and R_n are the precision and recall at the nth threshold, respectively.

Rank Power is an effective metric to evaluate how algorithm is ranking the outliers within the candidate ones. An outlier detection algorithm is considered more competent if it ranks true outliers in the top of the list of outlier candidates:

$$RP = \frac{k(k+1)}{2 * \sum_{i=1}^{k} R(x_i)} \tag{9}$$

where k is the number of true outliers found within s outliers candidates, and $R(x_i)$ is the rank of the ith true outlier x_i.

The AUC ROC tells how much an outlier detector is capable of distinguishing between outliers and inliers. ROC is a probability curve and AUC represents a summary statistic of the ROC curve, thus the higher the AUC the better the algorithm on average.

Time complexity reflects the computational complexity and describes the time an algorithm takes to finish execution. It shows the efficiency of the outlier detector on big data.

5.1.3 Comparing Algorithms

To make a challenging contest, ROD was compared with six popular outliers detection algorithms, each from different category. Moreover, we even increased the challenge and *tuned* the parameters of the other algorithms on each dataset, to make them achieve the highest precision possible, against the favor of ROD. However, in real-life scenarios, the default parameters values suggested in the literature are often used. The outlier detection algorithms are: the Statistic-based MCD [11], the Density-based LOF [13], the Angle-based ABOD [17], the Ensembles-based IForest [15], the Subspace-based SOD [19], and the Neural-Network-based AutoEncoder [22].

The experiments were conducted under the popular PyOD framework [35], which implements state-of-the-art outliers detection algorithms. The experiments were carried out on Intel Core i7-8750H with CPU clock rate 2.20 GHz and 16 GB RAM.

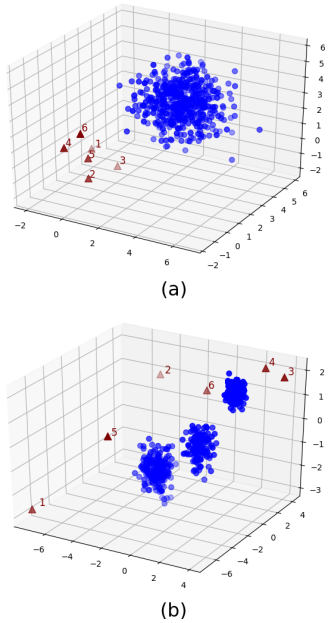


Fig. 3: (a) Synthetic data 1. (b) Synthetic data 2.

5.2 Experimental Results and Discussion

5.2.1 Synthetic Data

Two synthetic datasets were used to test the capability of ROD algorithm in different scenarios. Each dataset consists of 500 observations in which 6 of them are outliers. The first dataset involves two different statistical distributions where inliers come from Gaussian distribution and outliers come from Uniform distribution. On the other hand, the second dataset consists of 3 clusters that differ in size and density; this dataset involves the low density pattern problem and global outliers, which are considered as difficult tasks for outliers detection algorithms [36]. This data is generated using the appropriate off-the-shelf data generation functions provided by PyOD.

After applying the proposed method, all outliers were identified by ROD and ranked on the top of the list. The results are provided in Fig. 3 where the triangles refer to the outliers and the accompanied numbers indicate their ranks in the list.

5.2.2 Real-World Data

In this comparison, we first tuned the parameters of the *other algorithms* and used those that gave the highest precision on each dataset. Table 2 reports the precision (%) of outliers identification accomplished by the comparing algorithms @ top s (s=20, 50 and 100 respectively), in addition to the Average Precision (AP). Since the competition is intense, we considered the top two algorithms indicated by the bold type.

TABLE 2: Precision(%) of the Outliers Detection Algorithms on the Experimental Data

Dataset	Top s	MCD	LOF	ABOD	IForest	SOD	AutoEnc	ROD
SMTP	20	0.0	**65.0**	0.0	0.0	15.0	**65.0**	**65.0**
	50	0.0	**40.0**	0.0	0.0	6.0	**40.0**	34.0
	100	0.0	**20.0**	0.0	0.0	*3.0*	**20.0**	**20.0**
	AP	0.50	33.63	0.07	0.42	7.19	**35.54**	**51.04**
Banknote	20	**100.0**	35.0	0.0	85.0	20.0	**100.0**	**100.0**
	50	**90.0**	54.0	18.0	76.0	32.0	86.0	**94.0**
	100	72.0	54.0	27.0	77.0	32.0	**81.0**	**82.0**
	AP	56.57	47.97	38.68	51.90	42.98	**57.34**	**60.72**
Thyroid	20	**95.0**	**75.0**	40.0	60.0	25.0	60.0	**75.0**
	50	**86.0**	58.0	72.0	58.0	30.0	52.0	**74.0**
	100	**81.0**	47.0	**75.0**	60.0	33.0	44.0	65.0
	AP	**50.46**	25.20	18.16	31.57	21.20	19.16	**35.17**
Diabetes	20	**60.0**	40.0	**60.0**	**65.0**	45.0	45.0	55.0
	50	56.0	34.0	**58.0**	52.0	48.0	54.0	**62.0**
	100	46.0	28.0	51.0	**57.0**	47.0	50.0	**62.0**
	AP	48.79	34.31	45.14	**50.88**	42.22	44.23	**51.47**
Shuttle	20	60.0	50.0	30.0	**100.0**	55.0	45.0	**95.0**
	50	50.0	68.0	46.0	**98.0**	50.0	54.0	**98.0**
	100	53.0	56.0	53.0	**98.0**	48.0	56.0	**99.0**
	AP	84.11	11.64	17.55	**97.89**	11.55	91.54	**96.04**
Seismic	20	10.0	5.0	20.0	**30.0**	**30.0**	5.0	15.0
	50	12.0	8.0	18.0	**22.0**	20.0	12.0	**24.0**
	100	19.0	11.0	**21.0**	18.0	17.0	16.0	**25.0**
	AP	12.68	8.18	**15.49**	14.74	13.45	12.68	**16.8**
Digits	20	0.0	20.0	30.0	**50.0**	35.0	10.0	**75.0**
	50	12.0	12.0	16.0	**42.0**	18.0	34.0	**52.0**
	100	10.0	8.0	9.0	**45.0**	9.0	36.0	**45.0**
	AP	6.92	4.56	5.53	**32.88**	7.42	22.12	**34.74**
Cardio	20	40.0	20.0	50.0	**100.0**	65.0	90.0	**100.0**
	50	58.0	26.0	48.0	**84.0**	56.0	68.0	**72.0**
	100	51.0	23.0	36.0	57.0	44.0	**60.0**	**62.0**
	AP	35.68	13.29	21.50	58.36	29.51	**62.24**	**63.38**
WBC	20	45.0	20.0	30.0	**60.0**	45.0	50.0	**55.0**
	50	28.0	16.0	**32.0**	**34.0**	**32.0**	28.0	**32.0**
	100	18.0	12.0	**21.0**	**21.0**	18.0	17.0	17.0
	AP	44.07	15.96	32.84	**59.03**	42.13	50.12	**55.32**

From the experimental results, one can observe that ROD has predominant precision. Our method achieved the best performance in both AP and @s on six datasets and ranked the second best algorithm on the three remaining datasets.

It should be pointed out that the other 6 comparing algorithms showed inconsistency in the AP results. The AP results varied noticeably except for ROD which always ranked among the best two. This reflects the *overall* capability of ROD not to label as true outlier a sample that is negative and to find the true outlying samples. In other words, the ROD was a better model in ordering the predictions without considering any specific decision threshold, which is useful when true outliers ratio is imbalanced (i.e. too high or too low).

Table 3 shows the rank power of the top $s = 50$ achieved by the outliers detection algorithms on the experimental data using Eq. 9. Value 1.0 denotes the best possible result and means the method ranked all outliers on top of the list. One can observe that like the criterion of precision, ROD outperformed most of the other comparing algorithms on all datasets except Seismic and WBC where ROD ranked the third, yet very close to the top two.

For more comprehensive comparison and in order to average the performance of the outlier detection methods, one should take into account the True Positive Rate and False Positive Rate at different thresholds, which what AUC-ROC metric does.

Fig. 4 presents the varieties of the area under the curve of the comparing algorithms on the datasets. One can see that ROD had the largest area under the curve on three datasets and the second largest on the remaining datasets except for SMTP where the performance was relatively poor due to the features insignificance. For e.g., ROD ranked the first on Digits, with AUC = 0.95, and the second, but very

TABLE 3: Rank Power ($s = 50$) of the Outliers Detection Algorithms on the Experimental Data

Dataset	MCD	LOF	ABOD	IForest	SOD	AutoEnc	ROD
SMTP	0.0	0.50	0.0	0.0	0.42	**0.57**	**0.53**
Banknote	**0.97**	0.46	0.13	0.82	0.26	0.95	**0.99**
Thyroid	**0.91**	0.65	0.60	0.63	0.28	0.54	**0.75**
Diabetes	**0.57**	0.39	0.54	0.56	0.49	0.52	**0.60**
Shuttle	0.53	0.60	0.40	**0.99**	0.48	0.52	**0.96**
Seismic	0.11	0.07	0.21	**0.22**	**0.28**	0.11	0.21
Digits	0.10	0.19	0.29	**0.45**	0.36	0.25	**0.70**
Cardio	0.50	0.26	0.52	**0.91**	0.66	0.77	**0.86**
WBC	0.48	0.20	0.31	**0.54**	0.39	**0.54**	0.50

close to the first, with AUC = 0.99, on Shuttle. In addition to ROD, IForest, MCD and AutoEncoder were competitive algorithms; for e.g. AutoEncoder did well on many datasets, that is because of the reconstruction technique which applies error measures, resulting in good representation on whole data. One the other hand, one can see the influence of the curse of dimensionality on LOF where the concept of neighbourhood becomes meaningless. The LOF performance decreased as the dimensions of the datasets increased. *On average*, ROD showed high sensitivity to outliers compared to the others.

Table 4 records the elapsed time the comparing outlier detectors took to fit the data. One can observe that ROD ranked the second fastest on Banknote dataset, and the third fastest on three datasets,

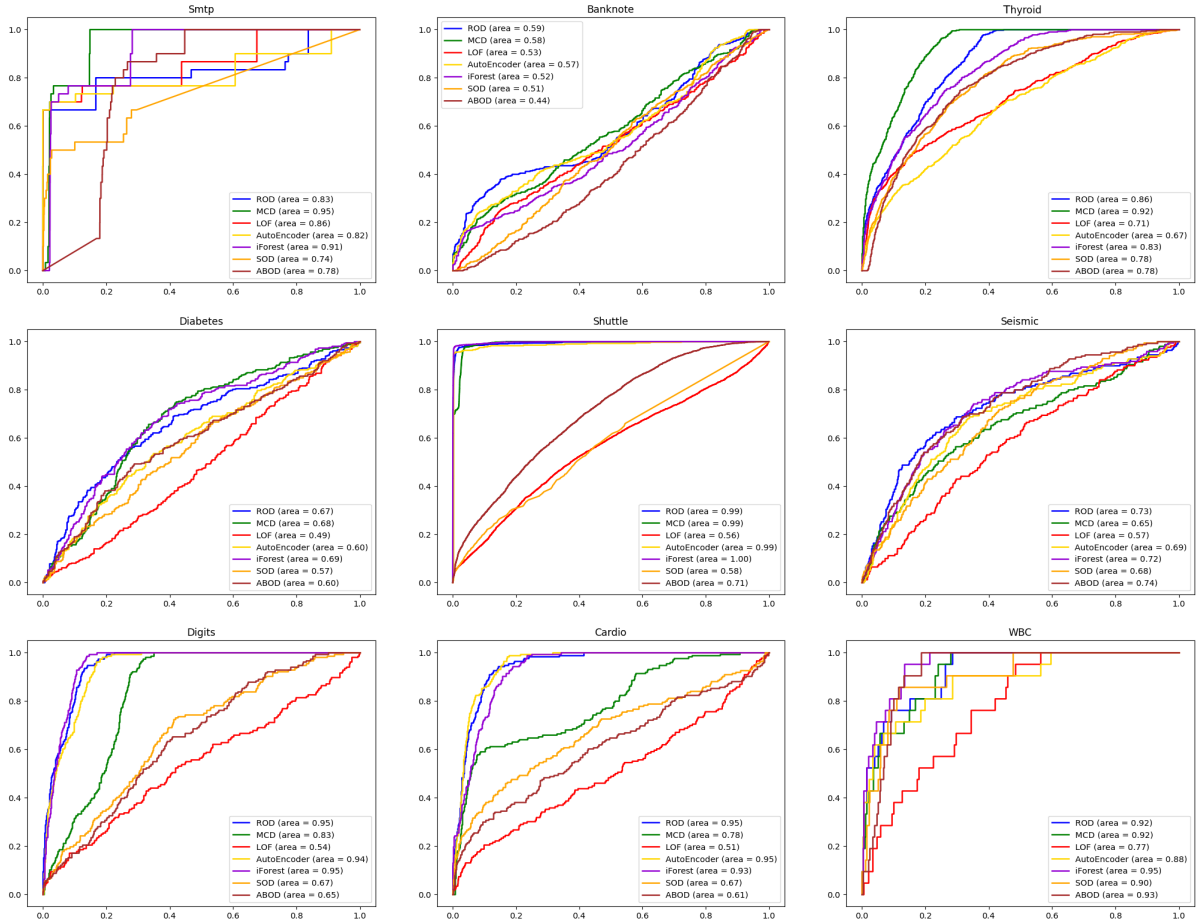


Fig. 4: AUC of the outlier detection algorithms on datasets

TABLE 4: Time Cost (in seconds) of the Outlier Detection Algorithms on the Experimental Data

Dataset	MCD	LOF	ABOD	IForest	SOD	AutoEnc	ROD
SMTP	22.38	**2.97**	12.37	**2.04**	4215.06	220.83	4.20
Banknote	0.46	**0.005**	8.73	0.12	0.98	3.98	**0.11**
Thyroid	1.32	**0.20**	0.96	**0.46**	21.63	16.76	1.52
Diabetes	0.44	*0.004*	0.61	**0.14**	0.33	2.91	0.59
Shuttle	16.33	**3.91**	171.61	**1.24**	1117.6	109.02	9.02
Seismic	0.37	**0.03**	0.39	**0.14**	2.90	7.18	3.89
Digits	5.05	**0.33**	1.10	**0.48**	27.12	16.37	64.67
Cardio	0.99	**0.15**	2.00	**0.21**	1.81	4.83	22.22
WBC	0.11	**0.01**	0.15	0.10	**0.09**	2.23	42.93

namely: SMTP, Thyroid and Shuttle. However, ROD time complexity is more sensitive to the number of features than the number of samples. ROD shows relatively poor performance as the dimensions increase due to its time complexity $\mathcal{O}(NC)$, where $N = number\ of\ samples$, and $C = \frac{d(d-1)(d-2)}{6} : d = number\ of\ attributes$. On the other hand, the time complexity of algorithms like SOD, AutoEncoder and ABOD are much more sensitive to the number of samples as shown in the table, e.g. SMTP and Shuttle. It should be pointed out that ROD running time is parameter-independent, unlike ABOD, SOD, and AutoEncoder which ranked the worst on big datasets. Finally, selecting limited number of features randomly might sound appropriate way to overcome ROD high complexity in very high dimensions. However, this would break the symmetry explained in Lemma 7 and might lead to lower precision. One solution, for future work, would be selecting certain combinations of subspaces of interest that most provide information about the outlying objects in 3D-dimensions.

6 Conclusion

This paper proposes a new outlier learning method for multivariate data, called ROD, that is parameter-free and has no distribution assumptions. The core work of ROD is to decompose the full attributes space into different combinations of subspaces, then the 3D-vectors, representing data points per 3D-subspace, are rotated about the geometric median two times counterclockwise using Rodrigues rotation formula. The results of the rotations are parallelepipeds where their volumes are mathematically analyzed as cost functions. Subsequently, ROD costs are used to calculate the MAD to obtain outlierness scores. Consequently, the 3D-scores are combined to construct the overall full-space outlying scores. The observations with high scores are promising candidates of outliers.

We performed a comprehensive comparison with six popular outlier detection methods, each from different category, on both synthetic and real-world datasets. The experimental results show that the proposed approach is promising, and its performance ranked the first at many aspects. Since ROD studies $\binom{d}{3}$ 3D-subspaces, the full attributes space is very well explored but at the cost of time complexity. In our future work, we will attempt to utilize ROD on limited number of 3D-subspaces of interest to speed up its running time without affecting its precision.

Appendix A
Proof of Theorem 1

Lemma 3. Starting with Rodrigues rotation formula for a vector $\vec{v} \in \mathbb{R}^3$ around a rotation-axis m, the rotated vector $\vec{v}_{rot}$ is given by (vector notations omitted):

$$v_{rot} = v\cos\theta + (m \times v)\sin\theta + m(m \cdot v)(1 - \cos\theta) \tag{10}$$

where θ is the angle between v and m.
Let: $v_{rot} = Rv$:

$$R = I + (\sin\theta)M + (1 - \cos\theta)M^2 \tag{11}$$

where I is the 3×3 identity matrix and $\mathbf{M}$ is the *cross-product matrix* of m:

$$\begin{bmatrix} 0 & -m_z & m_y \\ m_z & 0 & -m_x \\ -m_y & m_x & 0 \end{bmatrix}$$

Proof 1. Let $v \in D$ be a non-zero vector independent from m. Also, let v' and v'' be the two rotated vectors of v around m by $\theta_1 < \theta_2 \in (0, 2\pi)$ respectively according to the right hand rule, that is:

$$v' = R_1 v$$
$$v'' = R_2 v$$

Because v is independent from m; vectors v, v' and v'' will always form a three-dimensional parallelepiped which its signed volume is given by:

$$volume = v \cdot (v' \times v'') = v \cdot (R_1 v \times R_2 v) \quad (12)$$

Using 11 in 12:

$$\begin{aligned} volume = v \cdot [\,&(v + (\sin\theta_1)Mv \\ &+ (1-\cos\theta_1)M^2 v) \\ &\times (v + (\sin\theta_2)Mv \\ &+ (1-\cos\theta_2)M^2 v)\,] \end{aligned}$$

Applying the distributive property of cross-product over addition and Ptolemy's identities:

$$\begin{aligned} volume = \\ & v \cdot \Big[v \times v + v \times (\sin\theta_2)Mv \\ &+ v \times (1-\cos\theta_2)M^2 v + (\sin\theta_1)Mv \\ &\times v + (\sin\theta_1)Mv \times (\sin\theta_2)Mv \\ &+ (\sin\theta_1)Mv \times (1-\cos\theta_2)M^2 v \\ &+ (1-\cos\theta_1)M^2 v \times v \\ &+ (1-\cos\theta_1)M^2 v \times (\sin\theta_2)Mv \\ &+ (1-\cos\theta_1)M^2 v \times (1-\cos\theta_2)M^2 v \Big] \\ = v \cdot & \Big[(\sin\theta_2 - \sin\theta_1) v \times Mv \\ &+ (\cos\theta_1 - \cos\theta_2) v \times M^2 v \\ &+ [\sin\theta_1 - \sin\theta_2 + (\cos\theta_1 \sin\theta_2) \\ &- (\sin\theta_1 \cos\theta_2)] Mv \times M^2 v \Big] \\ = v \cdot & \Big[(\sin\theta_2 - \sin\theta_1) v \times (Mv) \\ &+ (\cos\theta_1 - \cos\theta_2) v \times (M^2 v) \\ &+ [\sin\theta_1 - \sin\theta_2 + \sin(\theta_2 - \theta_1)] \\ &(Mv) \times (M^2 v) \Big] \end{aligned}$$

$$\begin{aligned} volume = {} & [\sin\theta_1 - \sin\theta_2 + \sin(\theta_2 - \theta_1)] \\ & v \cdot [(Mv) \times (M^2 v)] \end{aligned}$$

Since $M = m \times v$, using the vector triple product property:

$$\begin{aligned} volume = {} & [\sin\theta_1 - \sin\theta_2 + \sin(\theta_2 - \theta_1)] \\ & v \cdot [(m \times v) \times ((m \cdot v)m - v)] \\ = {} & [\sin\theta_1 - \sin\theta_2 + \sin(\theta_2 - \theta_1)] \\ & v \cdot [\,(m \cdot v)[(m \times v) \times m] \\ & - (m \times v) \times v\,] \end{aligned}$$

Since the scalar triple product is unchanged under a circular shift of its three operands:

$$\begin{aligned} volume = {} & [\sin\theta_1 - \sin\theta_2 + \sin(\theta_2 - \theta_1)] \\ & (m \cdot v)\|m \times v\|^2 \end{aligned}$$

Applying Lagrange's identity:

$$\begin{aligned} volume = {} & [\sin\theta_1 - \sin\theta_2 + \sin(\theta_2 - \theta_1)] \\ & (m \cdot v)(\|v\|^2 - (m \cdot v)^2) \end{aligned} \quad (13)$$

Let $\gamma = \angle(\vec{m}, \vec{v})$, we know that:

$$\cos(\gamma) = \frac{\vec{m} \cdot \vec{v}}{\|\vec{m}\|\|\vec{v}\|} \implies \vec{m} \cdot \vec{v} = \cos(\gamma) \cdot \|\vec{m}\|\|\vec{v}\|$$

$$\begin{aligned} \implies volume = {} & [\sin\theta_1 - \sin\theta_2 + \sin(\theta_2 - \theta_1)] \\ & \|v\|^3(\cos\gamma - \cos\gamma^3) \end{aligned}$$

By ignoring the *constant* $[\sin\theta_1 - \sin\theta_2 + \sin(\theta_2 - \theta_1)] \neq 0$, the resulted volume can be approximated to (*and hence correlated to*) a cost function, given by:

$$\begin{aligned} volume \approx f(\vec{v}, \gamma) &= \|v\|^3(\cos\gamma - \cos\gamma^3) \\ &= \|v\|^3(\cos\gamma \sin\gamma^2) \end{aligned}$$

APPENDIX B
CALCULATION OF THRESHOLD ANGLES

$$\begin{aligned} 0 &= \frac{\delta \overset{\star}{f}}{\delta\gamma} \\ &= -\sin\gamma^3 + 2\sin\gamma\cos\gamma^2 \\ &= \sin\gamma(2\cos\gamma^2 - \sin\gamma^2) \\ \sin\gamma^2 = 2\cos\gamma^2 &\implies \tan\gamma^2 = 2 \\ \implies \gamma &= \arctan(\pm 1.4142) \\ &= \begin{cases} max.\ at\ \ 0.955\,rad. \\ min.\ at\ \ 2.186\,rad. \end{cases} \end{aligned}$$

APPENDIX C
DEVIATION PROPORTIONALITY

Let $\vec{v_1} = [1,2,3]$, $\vec{v_2} = [1,3,2]$, $\vec{v_3} = [-3,2,1] \in \mathbb{R}^3$, and $\vec{m}$ is the geometric median. Let $\gamma_1, \gamma_2, \gamma_3$ be $\angle(\vec{v_1}, \vec{m}) = 0.265$, $\angle(\vec{v_2}, \vec{m}) = 0.157$, $\angle(\vec{v_3}, \vec{m}) = 1.13\ rad.$ respectively.

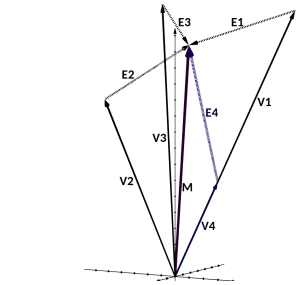
Although $\|\vec{v_1}\| = \|\vec{v_2}\| = \|\vec{v_3}\| = 3.741$ and $\overset{\star}{f}(\gamma_3) < \overset{\star}{f}(\gamma_2) < \overset{\star}{f}(\gamma_1)$, yet it can be easily noted that $\vec{v_3}$ deviates the most.

That is because $\frac{\pi}{2} \geq \gamma_3 > \alpha_1$ where $\overset{\star}{f}(\gamma) \not\propto \gamma$ over $(\alpha_1, \frac{\pi}{2}] \cup (\alpha_2, \pi]$.

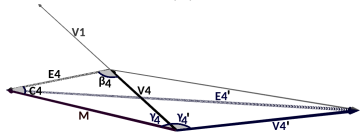
One solution to reserve a proper deviation proportionality, is to scale angles into the following intervals:

- $(0, \alpha_1]$ if $0 \leq \gamma \leq \frac{\pi}{2} : \implies \overset{\star}{f}(\gamma) \propto \gamma$
- $(\frac{\pi}{2}, \alpha_2]$ if $\frac{\pi}{2} < \gamma \leq \pi : \implies |\overset{\star}{f}(\gamma)| \propto \gamma$

APPENDIX D
VECTORS LINEARITY



(a)



(b)

Fig. 5: (a) Vectors Linearity use-case . (b) Parallelogram formed after centering $\vec{v_4}$ around $\vec{m}$

Consider Fig. 5(a) where we have four vectors: $\vec{v_1} = [1, 2, 3]$, $\vec{v_2} = [3, 1, 2]$, $\vec{v_3} = [2, 1, 3]$, $\vec{v_4} = [0.4, 0.8, 1.2] \in \mathbb{R}^3$; $\vec{m}$ is the geometric median, and $\|\vec{e_1}\| \neq \|\vec{e_2}\| \neq \|\vec{e_3}\| \neq \|\vec{e_4}\| > 0$ represent the Euclidean distances to $\vec{m}$ respectively.

Let $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ be $\angle(\vec{v_1}, \vec{m}) = 0.322, \angle(\vec{v_2}, \vec{m}) = 0.374, \angle(\vec{v_3}, \vec{m}) = 0.087, \angle(\vec{v_4}, \vec{m}) = 0.322\ rad.$ respectively. Although $\|\vec{V_4}\| = 1.49 < \|\vec{V_1}\| = \|\vec{V_2}\| = \|\vec{V_3}\| = 3.741$, the ROD cost of $\vec{v_4}$ is *masked*. That is because $\vec{v_1}$ and $\vec{v_4}$ are collinear, and therefore share same angle. This is also applicable for any vector with suspicious small magnitude and relatively normal deviation from $\vec{m}$.

Furthermore, in spite of the fact that $\|\vec{v_4}\|$ is the shortest, its Euclidean distance to $\vec{m}$ is conspicuously the longest: $||\vec{e_4}|| = 2.0 > ||\vec{e_2}|| > ||\vec{e_1}|| > ||\vec{e_3}||$.
Geometrically speaking, since $\vec{m}$ is affine equivariant, subtracting $\vec{m}$ from every vector shall not ruin the relative positional relations among the data points nor with $\vec{m}$.
Consider Fig. 5(b) where we focused from a different viewing angle on the previous example to show the change in the relationship with $\vec{m}$ after centering $\vec{v_4}$ around $\vec{m}$.

Let c_4 be $\angle(\vec{e_4}, \vec{m})$, β_4 be $\angle(\vec{e_4}, \vec{v_4})$, $\vec{v_4}'$, $\vec{e_4}'$ and ${\gamma_4}'$ be the new vector, the new vector that represents the Euclidean distance to $\vec{m}$ and the new angle $\angle(\vec{v_4}', \vec{m})$, after centering $\vec{v_4}$ around $\vec{m}$ respectively. Based on the facts that $c_4 \propto \|\vec{V_4}\|$ always holds true, and the diagonals of parallelogram divide it into four triangles of equal area, we have:
${\gamma_4}' = \gamma_4 + \beta_4 \implies c_4 = \pi - (\gamma_4 + \beta_4) = \pi - {\gamma_4}'$
$: c_4 \in [0, \frac{\pi}{2}] \implies {\gamma_4}' \in [\frac{\pi}{2}, \pi] \implies {\gamma_4}' \propto c_4^{-1}$
Using the law of cosines, and for any vector in the dataset, one can write: $c = \arccos\zeta$, where:

$$\zeta = \frac{\|\vec{m}\| - \|\vec{v}\| \cos\gamma}{\sqrt{\|\vec{v}\|^2 + \|\vec{m}\|^2 - 2\|\vec{v}\|\|\vec{m}\| \cos\gamma}}$$

And by exploiting the relationships between trigonometric functions and their inverse, we can find the relationship between $\overset{\star}{f}(\gamma')$ and the original $\|\vec{v}\|$ as follows:

$$\begin{aligned}
\overset{\star}{f}(\gamma') &= \cos(\pi - \arccos(\zeta)) \sin(\pi - \arccos(\zeta))^2 \\
&= -\cos(\arccos(\zeta)) \sin(\arccos(\zeta))^2 \\
&= -\zeta(\sqrt{1-\zeta^2})^2 = \zeta^3 - \zeta \\
&\implies \overset{\star}{f}(\gamma') < 0 \ \ whenever\ 1 > \zeta > 0 \\
&\implies \|\vec{m}\| - \|\vec{v}\| \cos\gamma > 0 \implies \frac{\|\vec{m}\|}{\cos\gamma} > \|\vec{v}\|
\end{aligned} \tag{14}$$

We conclude from Eq. 14 that the sign of $\overset{\star}{f}(\gamma')$ is related to the ratio between $\|\vec{m}\|$ and $\|\vec{v}\|$, normalized by $\cos\gamma$. As a result, centering the vectors around $\vec{m}$ unmasks vectors with suspicious small magnitudes by sorting them into two groups – Negatives and Positives. The above analysis can be generalized mathematically as follows:
$\forall\, \vec{v} \in D : \vec{v'} = \vec{v} - \vec{m} \implies$

$$\begin{cases} \|\vec{v'}\| = \|\vec{e}\|, \\ \gamma' \propto \|\vec{v}\|^{-1} \implies \begin{cases} \overset{\star}{f}(\gamma') < 0 \ if\ \|\vec{v}\| << \|\vec{m}\|, \\ \overset{\star}{f}(\gamma') > 0 \ if\ \|\vec{v}\| >> \|\vec{m}\| \\ \qquad\qquad and\ \cos\gamma > 0, \\ \overset{\star}{f}(\gamma') > 0 \ if\ \cos\gamma < 0 \end{cases} \end{cases}$$

APPENDIX E
PROOF OF PROPOSITION 1

Lemma 4. Let $U = \{\vec{u}_1, ...\vec{u}_m\}$ be a finite vector subspace. If $\vec{u}_i = [a_1, ..., a_d \mid d > 3] \in U : supp(\vec{u}_i) = \{j \in [d] \mid a_j \neq 0\} = 3$, then $rod(\vec{u}_i) = rod(\vec{u}_i^\star) : u_i^\star = [a_j \mid j \in [3], a_j \neq 0]$.

Proof: from vectors theory, a basis of *non-zero* vector subspace U is a linearly independent subset of U

that spans U. Accordingly, the zero basis vector does not exist for U, as it does not satisfy the linear independence and the spanning properties, making $dim(U) = 3$.

Lemma 5. If $rod(\vec{u})$ is anomalous, $\vec{u} = [a_1, a_2, a_3] \neq \vec{0}$, then $\vec{u}$ is anomalous in at least one of its three dimensions.

Proof: One can see from Eq. 13 that since $\vec{m}$ is at fixed point, $\implies$ $(\vec{m} \cdot \vec{v})$, $\|\vec{v}\|^2$ are correlated to $a_i \; \forall \; i \in [3]$.

Lemma 6. A finite vector space V over $\mathbb{R}^{\mathrm{d}} \mid d > 3$ can be decomposed into different combinations of subspaces where each decomposition set $s_i \subset S$ contains different combination of subspaces U_{ij} that describes V uniquely. Hence, there exist $\frac{d(d-1)(d-2)}{6}$ subspaces of $dim(U_{ij}) = 3$.

Proof: By the definition of direct sum:
$V = \bigoplus_{j \in J} U_j \iff V = U_1 + ... + U_j : U_1 \cap ... \cap U_j = \{0\}$; and since $dim(U_j) \in [d-1]$, $\exists \; S = \{s_{i \in I} \mid s_i = \{U_1 \oplus ... \oplus U_j\}\} \rightarrow \exists \binom{d}{3}$ subspaces $\vdash U_{ij} = \mathbb{R}^3$.

Lemma 7. The number of occurrences of each dimension in the resulted 3D-subspace components of the full-dimensional space is equiprobable and has balanced weight (i.e. importance) over all dimensions.

Proof: By symmetry, choosing combinations of 3 dimensions from d, each chosen dimension will have the same probability of occurrence = $\frac{3}{n}\binom{n}{3}$ and the importance of the selection of each dimension is evenly balanced. Given a dataset:

$$D = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & \boldsymbol{a_{22}} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & \boldsymbol{a_{34}} \end{pmatrix} \subseteq \mathbb{R}^{3 \times 4}$$

as a toy example where $\vec{v}_1$, $\vec{v}_2$, $\vec{v}_3$ are the first, second and third rows in D respectively and the bold element refers to an outlier at that dimension in the given vector. By lemmas 5 and 6, one can construct the following 3D views and weight ratios:

3D Views	$\vec{v}_1$	$\vec{v}_2$	$\vec{v}_3$	Weights
View 1	$\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix}$	$\begin{bmatrix} a_{21} \\ \boldsymbol{a_{22}} \\ a_{23} \end{bmatrix}$	$\begin{bmatrix} a_{31} \\ a_{32} \\ a_{33} \end{bmatrix}$	0:1:0
View 2	$\begin{bmatrix} a_{11} \\ a_{12} \\ a_{14} \end{bmatrix}$	$\begin{bmatrix} a_{21} \\ \boldsymbol{a_{22}} \\ a_{24} \end{bmatrix}$	$\begin{bmatrix} a_{31} \\ a_{32} \\ \boldsymbol{a_{34}} \end{bmatrix}$	0:1:1
View 3	$\begin{bmatrix} a_{11} \\ a_{13} \\ a_{14} \end{bmatrix}$	$\begin{bmatrix} a_{21} \\ a_{23} \\ a_{24} \end{bmatrix}$	$\begin{bmatrix} a_{31} \\ a_{33} \\ \boldsymbol{a_{34}} \end{bmatrix}$	0:0:1
View 4	$\begin{bmatrix} a_{12} \\ a_{13} \\ a_{14} \end{bmatrix}$	$\begin{bmatrix} \boldsymbol{a_{22}} \\ a_{23} \\ a_{24} \end{bmatrix}$	$\begin{bmatrix} a_{32} \\ a_{33} \\ \boldsymbol{a_{34}} \end{bmatrix}$	0:1:1
4D View				**0:3:3**

One can observe that the weights are also proportional to the number of outliers in all dimensions per data observation. However, this is not the case for arbitrary (i.e. naive) random selection, where the outlier per some dimension would gain more views leading to unbalanced weights overall.

Based on the Motive in 4.1 and the Lemmas 4, 5, 6 and 7, considering remarks in 4.2; we extend ROD into higher dimensions by decomposing the full attributes space into different combinations of subspaces, then applying ROD only on the collected 3D subspaces. Subsequently, we construct an overall outlying score by averaging the ROD 3D-subspaces scores per data sample.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees and the associate editor for their valuable comments and constructive suggestions, which have improved this paper greatly.

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, p. 15, 2009.

[2] V. K. Pang-Ning Tan, Michael Steinbach, *Introduction to Data Mining*. USA: Pearson Addison-Wesley, 2006.

[3] A. K. Deepa Verma, Rakesh Kumar, "Survey paper on outlier detection using fuzzy logic based method," *International Journal on Cybernetics and Informatics*, vol. 6, pp. 29–35, 04 2017.

[4] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, pp. 363–387, 2012.

[5] T. L. Vic Barnett, *Outliers in Statistical Data*, 3rd ed. USA: Wiley, 1994.

[6] C. C. Aggarwal, *Outlier Analysis*, 2nd ed. IBM T.J. Watson Research Center, Yorktown Heights, New York, USA: Springer, Cham, 2016.
[7] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, pp. 85–126, 10 2004.
[8] S. Kaliyaperumal and M. Kuppusamy, "Outlier detection in multivariate data," *Applied Mathematical Sciences*, vol. 9, pp. 2317–2324, 01 2015.
[9] P. B. O. Krishna Modi, "Outlier analysis approaches in data mining," *IJIRT*, vol. 3, pp. 6–12, 12 2016.
[10] E. Eskin, "Anomaly detection over noisy data using learned probability distributions," in *Proceedings of the Seventeenth International Conference on Machine Learning*, San Francisco, CA, USA, 2000, pp. 255–262.
[11] M. D. Mia Hubert, "Minimum covariance determinant," *John Wiley and Sons, Inc.*, vol. 2, pp. 36–43, 01 2010.
[12] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *SIGMOD Rec.*, vol. 29, pp. 427–438, 05 2000.
[13] M. M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander, "Lof: Identifying density-based local outliers," *ACM Sigmod Record*, vol. 29, pp. 93–104, 06 2000.
[14] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, pp. 157–166.
[15] F. T. Liu, K. Ming Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, 12 2008, pp. 413–422.
[16] S. Aryal, K. M. Ting, J. R. Wells, and T. Washio, "Improving iforest with relative mass," in *Advances in Knowledge Discovery and Data Mining*, Cham, 2014, pp. 510–521.
[17] M. S. Hans-Peter Kriegel and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *ACM SIGKDD KDD'08*, Las Vegas, USA, 2008, pp. 444–452.
[18] R. P. Ninh Pham, "A near-linear time approximation algorithm forangle-based outlier detection in high-dimensional data," in *ACM KDD'12*, Beijing, China, 08 2012.
[19] E. S. Hans-Peter Kriegel, Peer Kroger and A. Zimek, "Outlier detection in axis-parallel subspaces of high dimensional data," in *PAKDD 2009*, Bangkok, Thailand, 2009.
[20] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognition*, vol. 74, pp. 406–421, 2018.
[21] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *CoRR*, vol. abs/1901.03407, 02 2019.
[22] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proceedings of the 2017 SIAM International Conference on Data Mining*, 2017, pp. 90–98.
[23] H. X. Shashi Shekhar and X. Zhou, *Encyclopedia of GIS*, 2nd ed. USA: Springer International Publishing, 2017.
[24] Y. Vardi and C.-H. Zhang, "The multivariate l1-median and associated data depth," *Proceedings of the National Academy of Sciences*, vol. 97, no. 4, pp. 1423–1426, 2000.
[25] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *Journal of Experimental Social Psychology*, vol. 49, no. 4, pp. 764 – 766, 2013.
[26] B. Iglewicz and D. C. Hoaglin, *How to detect and handle outliers*. Asq Press, 1993, vol. 16.
[27] E. Müller, I. Assent, P. Iglesias, Y. Mülle, and K. Böhm, "Outlier ranking via subspace analysis in multiple views of the data," in *2012 IEEE 12th international conference on data mining*. IEEE, 2012, pp. 529–538.
[28] C. C. Aggarwal, "High-dimensional outlier detection: The subspace method," in *Outlier Analysis*. Springer, 2017, pp. 149–184.
[29] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Outlier detection in arbitrarily oriented subspaces," in *2012 IEEE 12th international conference on data mining*. IEEE, 2012, pp. 379–388.
[30] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *Acm Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.
[31] K. Beyer and J. Goldstein, "R. ramakrishnan, u. shaft. when is nearest neighbors meaningful," in *Proc of ICDT*, vol. 99, 1999.
[32] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *International conference on database theory*. Springer, 2001, pp. 420–434.
[33] M. Zhu, "Recall, precision and average precision," 2004.
[34] J. Ha, S. Seok, and J.-S. Lee, "A precise ranking method for outlier detection," *Information Sciences*, vol. 324, pp. 88–107, 2015.
[35] Y. Zhao, Z. Nasrullah, and Z. Li, "Pyod: A python toolbox for scalable outlier detection," *arXiv preprint arXiv:1901.01588*, 2019.
[36] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2002, pp. 535–548.

Yahya Almardeny is a Machine Learning Engineer, Researcher and Data Analyst in Telecommunications Software and Systems Group (TSSG) and PT lecturer in Waterford Institute of Technology (WIT). He is currently pursuing master's degrees in Artificial Intelligence in University Limerick (UL). Yahya received a certificate in AI from UL in 2019 and a BSc in Software Systems Development from WIT in 2018.

Noureddine Boujnah is an assistant professor at Gabes university, he received his PhD from Polytechnic of Turin Italy in satellite communication in 2011, matser of science in signal processing from Higher school of communication of Tunis (Sup'Com-) and Engineering degree in telecommunication from Sup'com too.

Frances Cleary is a Research Unit Manager in TSSG, WIT and actively leads a Division that has expertise relating to VR, Pervasive sensing, Embedded Systems and Bio-Nano Communications. She currently manages a Research Division with approximately 20 researchers active in cutting edge research in the above thematic research areas.