Open Journal of Statistics, 2015, 5, 618-630
Published Online October 2015 in SciRes. http://www.scirp.org/journal/ojs
http://dx.doi.org/10.4236/ojs.2015.56063

Random Subspace Learning Approach to High-Dimensional Outliers Detection

Bohan Liu, Ernest Fokoué

School of Mathematical Sciences, Rochester Institute of Technology, Rochester, NY, USA
Email: bl3267@rit.edu, epfeqa@rit.edu

Received 16 September 2015; accepted 27 October 2015; published 30 October 2015

Abstract

We introduce and develop a novel approach to outlier detection based on adaptation of random subspace learning. Our proposed method handles both high-dimension low-sample size and traditional low-dimensional high-sample size datasets. Essentially, we avoid the computational bottleneck of techniques like Minimum Covariance Determinant (MCD) by computing the needed determinants and associated measures in much lower dimensional subspaces. Both theoretical and computational development of our approach reveal that it is computationally more efficient than the regularized methods in high-dimensional low-sample size, and often competes favorably with existing methods as far as the percentage of correct outlier detection are concerned.

Keywords

High-Dimensional, Robust, Outlier Detection, Contamination, Large *p* Small *n*, Random Subspace Method, Minimum Covariance Determinant

1. Introduction

We are given a dataset $\mathcal{D}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$, where $\boldsymbol{x}_i=(x_{i1},\cdots,x_{ip})^\top\in\mathcal{X}\subset\mathbb{R}^{1\times p}$, under the special scenario in which $n\lll p$ refers to as high dimensional low sample size (HDLSS) setting. It is assumed that the basic distribution of the X_i'sis multivariate Gaussian, so that the density of X is given by $\phi_p(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$, with

$$\phi_p(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})=\frac{1}{\sqrt{(2\pi)^p|\boldsymbol{\Sigma}|}}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}. \quad (1)$$

It is also further assumed that the data set $\mathcal{D}$ is contaminated, with a proportion $\varepsilon\in(0,\tau)$ where $\tau<\mathrm{e}^{-1}$, of observations that are outliers, so that under ε-contamination regime, the probability density function of X

How to cite this paper: Liu, B.H. and Fokoué, E. (2015) Random Subspace Learning Approach to High-Dimensional Outliers Detection. *Open Journal of Statistics*, **5**, 618-630. http://dx.doi.org/10.4236/ojs.2015.56063

is given by

$$p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \varepsilon, \eta, \gamma) = (1-\varepsilon)\phi_p(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \varepsilon\phi_p(\boldsymbol{x}; \boldsymbol{\mu}+\eta, \gamma\boldsymbol{\Sigma}), \tag{2}$$

where η represents the contamination of the location parameter $\boldsymbol{\mu}$, while γ captures the level of contamination of the scatter matrix $\boldsymbol{\Sigma}$. Given a dataset with the above characteristics, the goal of all outlier detection techniques and methods is to *select and isolate as many outliers as possible so as to perform robust statistical procedures non-aversely affected by those outliers*. In such scenarios, where the multivariate Gaussian is the assumed basic underlying distribution, the classical Mahalanobis distance is the default measure of the proximity of the observations, namely

$$d^2_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{x}_i) = (\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}). \tag{3}$$

And experimenters of often addressing and tackling the outlier detection task in such situations using either the so-called Minimum Covariance Determinant (MCD) algorithm [1] or some extensions or adaptations thereof. The MCD is described as followed:

Minimum Covariance Determinant (*MCD*)

***Step* 1.** Select *h* observations, and form the dataset $\mathcal{D}_H$, $H \subset \{1, \cdots, n\}$;

***Step* 2.** Compute the empirical covariance $\hat{\boldsymbol{\Sigma}}_H$ and mean $\hat{\boldsymbol{\mu}}_H$;

***Step* 3.** Compute the Mahalanobis distances $d^2_{\hat{\boldsymbol{\mu}}_H, \hat{\boldsymbol{\Sigma}}_H}(\boldsymbol{x}_i), i = 1, \cdots, n$;

***Step* 4.** Select the *h* observations having the smallest Mahalanobis distance;

***Step* 5.** Update $\mathcal{D}_H$ and repeat **Steps 2** to **5** until $\det\left(\hat{\boldsymbol{\Sigma}}_H\right)$ no longer decreases;

The MCD algorithm can be formulated as an optimization problem.

$$\left(\hat{H}, \hat{\boldsymbol{\mu}}_H, \hat{\boldsymbol{\Sigma}}_H\right) = \underset{\boldsymbol{\mu},\boldsymbol{\Sigma},H}{\operatorname{argmin}} \left\{\mathcal{E}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, H)\right\}. \tag{4}$$

The MCD algorithm can be formulated as an optimization problem. The seminal MCD algorithm proposed by [1], which is turned out to be rather slow and did not scale well as a function of the sample size *n*. That limitation of MCD leads its author to the creation of the so-called FAST-MCD [2], focused on solving the outlier detection problem in a more computationally efficient way. Since the algorithm only needs to select a limited number *h* of observations for each loop, its complexity can be reduced when sample size *n* is large, since only a small fraction of the data is used. However, it must be noted that the bulk of the computations in MCD has to do with the estimation of determinants and the Mahalanobis distances, both requiring a complexity of $O\left(p^3\right)$ where *p* is the dimensionality of the input space as defined earlier. Therefore, it becomes crucial to find out how MCD fares when *n* is large and *p* is also large, even the now quite ubiquitous scenario where *n* is small but *p* is very larger, and indeed much larger than *n*. This *p* larger than *n* scenario, referred to as high dimension low sample size (HDLSS) is very common nowadays in application domains such as gene expression datasets from RNA-sequencing and microarray, audio processing, image processing, just to name a few. As noted before, with the MCD algorithm, *h* observations have to be selected to compute the robust estimator. Unfortunately, when $n \lll p$, neither the inverse nor the determinant of covariance matrix can be computed. As we'll show later, the $O\left(p^3\right)$ complexity of matrix inversion and determinant computation renders MCD untenable for *p* as moderate as 500. Therefore, it is natural, in the presence of HDLSS datasets, to contemplate at least some intermediate dimensionality reduction step prior to performing the outlier detection task. Several algorithms have been proposed, among which PCOut by [3], Regularized MCD (R-MCD) by [4] and other ideas by [5]-[8]. When instability in the data makes the computation of $\boldsymbol{\Sigma}$ problematic in *p* dimension, regularized MCD may be used with objective function

$$\mathcal{E}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, H, \lambda) = \mathcal{E}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, H) + \lambda \operatorname{trace}\left(\boldsymbol{\Sigma}^{-1}\right), \tag{5}$$

where λ is the so-called regularizer or tuning parameter, chosen to stabilize the procedure. However, it turns out that even the above Regularized MCD cannot be contemplated when $p \ggg n$, since $\det\left(\hat{\boldsymbol{\Sigma}}\right)$ is always zero in such cases. The solution to that added difficulty is addressed by solving

$$\left(\hat{H}, \hat{\boldsymbol{\mu}}_H, \hat{\boldsymbol{\Sigma}}_H\right) = \operatorname{argmax}\left\{\log\left\{\det\left(\tilde{\boldsymbol{\Sigma}}\right)\right\} + \frac{1}{h}\sum_{i \in H}(\boldsymbol{x}-\boldsymbol{\mu})^\top \tilde{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) + \lambda \operatorname{trace}\left(\tilde{\boldsymbol{\Sigma}}^{-1}\right)\right\}, \tag{6}$$

where the regularized covariance matrix $\boldsymbol{\Sigma}$ is given by

$$\tilde{\boldsymbol{\Sigma}}(\alpha)=(1-\alpha)\hat{\boldsymbol{\Sigma}}+\frac{\alpha}{p}\operatorname{trace}\left(\hat{\boldsymbol{\Sigma}}\right)\boldsymbol{I}_p. \tag{7}$$

With $\alpha\in(0,1)$, for many HDLSS datasets however, the dimensionality p of the input space is often large, with numbers like $p\geq 10^3$ or even $p\geq 10^4$ rather very common. As a result, even the above direct regularization is computationally intractable, because when p is large, the $O\left(p^3\right)$ complexity of the needed matrix inversion and determinant calculation makes the problem computationally untenable. The fastest matrix inversion algorithms like [9] [10] are theoretically around $O\left(p^{2.376}\right)$ and $O\left(p^{2.373}\right)$, and so complicated that there are virtually no useful implementation of any of them. In short, the regularization approach to MCD like algorithms is impractical and unusable for HDLSS datasets even for values of p around a few hundreds. Another approach to outlier detection in the HDLSS context has revolved around extensions and adaptations of principle component analysis (PCA). Classical PCA seeks to project high dimensional vectors onto a lower dimensional orthogonal space while maximizing the variance. By reducing the dimensionality of the original data, one seeks to create a new data representation that evades the curse of dimensionality. However, PCA, in its generic form, is not robust, for the obvious reason that it is built by a series of transformations of means and covariance matrices whose generic estimators are notoriously non robust. It is therefore of interest to seek to perform PCA in a way that does not suffer from the presence of outliers in the data, and thereby identify the outlying observations as a byproduct of such a PCA. Many authors have worked on the robustification of PCA, and among them [11] whose proposed ROBPCA, a robust PCA method, which essentially robustifies PCA by combining MCD with the famous *projection pursuit* technique ([12] [13]). Interestingly, if instead of reducing the dimensionality based on robust estimators, one can first apply PCA to the whole data, then outliers may surprisingly lie on several directions where they are then exposed more clearly and distinctly. Such an insight appears to have motivated the creation of the so-called PCOut algorithm proposed by [3]. PCOut uses PCA as part of its preprocessing step after the original data has been scaled by Median Absolute Deviation (MAD). In fact, in PCOut, each attribute is transformed as follows

$$\boldsymbol{x}_j^*=\frac{\boldsymbol{x}_j-\tilde{\boldsymbol{x}}_j}{\operatorname{MAD}\left(\boldsymbol{x}_j\right)},\ j=1,\cdots,p, \tag{8}$$

where $\boldsymbol{x}_j=\left(x_{1j},\cdots,x_{nj}\right)\subset\mathbb{R}^{n\times 1}$ and $\boldsymbol{x}_j$ is the median of $\boldsymbol{x}_j$. Then $\boldsymbol{X}^*=\left[\boldsymbol{x}_1^*,\boldsymbol{x}_2^*,\cdots,\boldsymbol{x}_p^*\right]$, PCA can be performed, namely

$$\boldsymbol{X}^{*\top}\boldsymbol{X}^*=\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\top}. \tag{9}$$

From which the principal component scores $\boldsymbol{Z}=\boldsymbol{X}^*\cdot\boldsymbol{V}$ may then be used for the purpose of outlier detection. In fact, it also turns out that the principal component scores $\boldsymbol{Z}$ may be re-scaled to achieve a much lower dimension with 99% variance retained. Unlike MCD, PCA based re-scaled method is not only practical but also performs better with high dimensional datasets. 99% of simulated outliers are detected when $n=2000$, $p=2000$. A higher false positive rate is reported in low dimensional cases, and less than half of the outliers were identified in scenarios with $n=2000$, $p=50$. It is clear by now that with HDLSS datasets, some form of dimensionality reduction is needed prior to performing outlier detection. Unlike the authors just mentioned who all resorted to some extension or adaptation of principal component analysis wherein dimensionality reduction is based on transformational projection, we herein propose an approach where dimensionality reduction is not only stochastic but also selection-based rather than projection-based. The rest of this paper is organized as follows: in Section 2, we present a detailed description of our proposed approach, along with all the needed theoretical and conceptual justifications. In the interest of completeness, we close this section with the general description of a nonparametric machine learning kernel method for novelty detection known as the one-class support vector machine, which under suitable conditions is an alternative to the outlier detection approach proposed in this paper. Section 3 contains our extensive computational demonstrations on various scenarios. We specifically present the comparisons of the predictive/detection performances between our RSSL based approach and the PCA based methods discussed earlier. We mainly used simulated data here, with simulations seeking to assess the impact of various aspects of the data such as the dimensionality p of the input space, the contamination rate ε and other aspects like the magnitude γ of the contamination of the scatter matrix. We conclude with Section 4,

in which we provide a thorough discussion of our results along with various pointers to our current and future work on this rather compelling theme of outlier detection.

2. Random Subspace Learning Approach to Outlier Detection

2.1. Rationale for Random Subspace Learning

We herein propose a technique that combines the concept underlying Random Subspace Learning (RSSL) by [14] with some of the key ideas behind minimum covariance determinant (MCD) to achieve a computational efficient, scalable, intuitive appealing and highly accurate outlier detection method for both HDLSS and LDHSS datasets. With our proposed method, the computation of the robust estimators of both location and scatter matrix can be achieved by tracing the optimal subspaces directly. Besides, we demonstrate via practical examples that our RSSL based method is computationally very efficient, specifically because it turns out that, unlike the other methods mentioned earlier, our method does not require the computationally expensive calculations of determinants and Mahalanobis distances at each step. Moreover, whenever such calculations are needed, they are all performed in very low dimensional spaces, further emphasizing the computational strength of our approach. The original MCD algorithm formulates the outlier detection problem as the problem of finding the smallest determinants of covariance computed from a sequence $\mathcal{D}_h^{(k)}, k=1,\cdots,m$ of different subsets of the original data set $\mathcal{D}$. Each subset contains h observations. More precisely, if $\mathcal{D}_{\text{optimal}}$ is the subset of $\mathcal{D}$ whose observations yield the estimated covariance matrix with the smallest (minimum) determinant out of all the m subsets considered, then we must have follows

$$\det\left(\hat{\boldsymbol{\Sigma}}\left(\mathcal{D}_{\text{optimal}}\right)\right)=\min\left\{\det\left(\hat{\boldsymbol{\Sigma}}\left(\mathcal{D}_h^{(1)}\right)\right),\det\left(\hat{\boldsymbol{\Sigma}}\left(\mathcal{D}_h^{(2)}\right)\right),\cdots,\det\left(\hat{\boldsymbol{\Sigma}}\left(\mathcal{D}_h^{(m)}\right)\right)\right\}, \tag{10}$$

where m is the number of iterations needed for the MCD algorithm to converge. $\mathcal{D}_{\text{optimal}}$ is the subset of $\mathcal{D}$ that produces the estimated covariance matrix with the smallest determinant. The MCD estimates of the location vector and scatter matrix parameters are given by follows

$$\hat{\boldsymbol{\mu}}_{\text{MCD}}=\hat{\boldsymbol{\mu}}\left(\mathcal{D}_{\text{optimal}}\right) \quad \text{and} \quad \hat{\boldsymbol{\Sigma}}_{\text{MCD}}=\hat{\boldsymbol{\Sigma}}\left(\mathcal{D}_{\text{optimal}}\right). \tag{11}$$

The number h of observations in each subset is required to be $n/2 \le h < n$. It turns out that $h=\left[\left(n+p+1\right)/2\right]$ reaches its highest possible breakdown value according to [15]. It is obvious that with $h=\left[\left(n+p+1\right)/2\right]$ being the highest breakdown point, $n/2 \le h < n$ cannot be achieved in the HDLSS context, since in such a context $p \ggg n$. It is therefore intuitively appealing to contemplate a subspace of the input space $\mathcal{X}$, and define/construct such a subspace in such a way that its dimensionality $d < p$ is also such that $d < n$ to allow the seamless computation of the needed distances.

2.2. Description Random Subspace Learning for Outlier Detection

Random Subspace Learning in its generic form is designed for precisely this kind of procedure. In a nutshell, RSSL combines instance-bagging (bootstrap *i.e.* sampling observations with replacement) with attribute-bagging (sampling indices of attributes without replacement), to allow efficient ensemble learning in high dimensional spaces. Random Subspace Learning (Attribute Bagging) proceeds very much like traditional bagging, with the added crucial step consisting of selecting a subset of the variables from the input space for training rather than building each base learners using all the p original variables.

Random Subspace Learning (*RSSL*): *Attribute-bagging step*

***Step* 1.** Randomly draw the number $d < p$ of variables to consider;

***Step* 2.** Draw without replacement the indices of d variables of the origina p variables;

***Step* 3.** Perform learning/estimation in the d-dimensional subspace.

This attribute-bagging step is the main ingredient of our outlier detection approach in high dimensional spaces.

Random Subspace Outlier

***Step* 1.** Draw with replacement $\left\{i_1^{(b)},\cdots,i_n^{(b)}\right\}$, from $\{1,2,\cdots,n\}$ to form the bootstrap sample $\mathcal{D}^{(b)}$;

***Step* 2.** Start *for* $b=1$ to *B do*:

Draw without replacement from $\{1,2,\cdots,p\}$ a subset $\left\{j_1^{(b)},\cdots,j_d^{(b)}\right\}$ of d variables

Drop unselected variables from $\mathcal{D}^{(b)}$ o that $\mathcal{D}_{sub}^{(b)}$ is d dimensional

Build the b th determinant of covariance $\det\left(\hat{\boldsymbol{\Sigma}}\left(\mathcal{D}_{sub}^{(b)}\right)\right)$

End *for*

***Step* 3.** Sort the ensemble $\left\{\det\left(\hat{\boldsymbol{\Sigma}}\left(\mathcal{D}_{sub}^{(b)}\right)\right), b=1,\cdots,B\right\}$;

***Step* 4.** Form $\mathcal{D}^*:\det\left(\mathcal{D}^*\right)=\operatorname{argmin}\left\{\det\left(\hat{\boldsymbol{\Sigma}}\left(\mathcal{D}_{sub}^{(b)}\right)\right), b=1,\cdots,B\right\}$;

***Step* 5.** Compute $\hat{\boldsymbol{\mu}}^*$ and $\hat{\boldsymbol{\Sigma}}^*$ base on $\mathcal{D}^*$.

We can build the robust distance by

$$\widehat{\delta^*}(\boldsymbol{x})=\left(\boldsymbol{x}-\hat{\boldsymbol{\mu}}^*\right)^{\top}\hat{\boldsymbol{\Sigma}}^{*-1}\left(\boldsymbol{x}-\hat{\boldsymbol{\mu}}^*\right). \tag{12}$$

The RSSL outlier detection algorithm computes a determinant of covariance for each subsample, with each subsample residing in a subspace spanned by the d randomly selected variables, where d is usually selected to be $\min\left(n/5,\sqrt{p}\right)$. A total of B subsets are generated, and their low dimensional covariance matrices are formed along with the corresponding determinants. Then the best subsample, meaning the one with the smallest covariance determinant is singled. It turns out that in the LDHSS context $\left(n\ggg p\right)$, our RSSL outlier detection algorithm always robustly yields the robust estimators $\hat{\boldsymbol{\mu}}^*$ and $\hat{\boldsymbol{\Sigma}}^*$ needed to compute the Mahalanobis distance for all the observations. Then the outliers can be selected using the typical cut-off built on classical $\chi^2_{p,5\%}$. In HDLSS context, in order to handle the curse of dimensionality, we need to involve a new variable selection procedure to adjust our framework and concurrently stabilize the detection. The modified version of our RSSL outlier detection algorithm in HDLSS is then given by

Random Subspace Learning for Outlier Detection when $n\lll p$

***Step* 1.** Draw with replacement $\left\{i_1^{(b)},\cdots,i_n^{(b)}\right\}$, from $\{1,2,\cdots,n\}$ to form the bootstrap sample $\mathcal{D}^{(b)}$;

***Step* 2.** Start *for* $b=1$ to B *do*:

Draw without replacement from $\{1,2,\cdots,p\}$ a subset $\left\{j_1^{(b)},\cdots,j_d^{(b)}\right\}$ of d variables

Drop unselected variables from $\mathcal{D}^{(b)}$ so that $\mathcal{D}_{sub}^{(b)}$ is d dimensional

Build the b th determinant of covariance $\det\left(\hat{\boldsymbol{\Sigma}}\left(\mathcal{D}_{sub}^{(b)}\right)\right)$

End *for*

***Step* 3.** Sort the ensemble $\left\{\det\left(\hat{\boldsymbol{\Sigma}}\left(\mathcal{D}_{sub}^{(b)}\right)\right), b=1,\cdots,B\right\}$;

***Step* 4.** Keep the k smallest samples based on elbow to form $\mathcal{D}^{(\eta)}$, where $\eta=1,\cdots,k$ and $k<B$;

***Step* 5.** Start *for* $j=2$ to d *do*:

Select $v=j$ most frequent variables left in $\mathcal{D}^{(\eta)}$ to compute $\det\left(\hat{\boldsymbol{\Sigma}}\left(\mathcal{D}_{sub=j}^{(\eta=1)}\right)\right)$

End *for*

***Step* 6.** Form $\mathcal{D}^*:\det\left(\mathcal{D}^*\right)=\operatorname{argmin}\left\{\det\left(\hat{\boldsymbol{\Sigma}}\left(\mathcal{D}_{sub=j}^{(\eta=1)}\right)\right), j=2,\cdots,d\right\}$;

***Step* 7.** Compute $\hat{\boldsymbol{\mu}}^*$ and $\hat{\boldsymbol{\Sigma}}^*$ base on $\mathcal{D}^*$.

We can build the robust distance by the same way:

$$\widehat{\delta^*}(\boldsymbol{x})=\left(\boldsymbol{x}-\hat{\boldsymbol{\mu}}^*\right)^{\top}\hat{\boldsymbol{\Sigma}}^{*-1}\left(\boldsymbol{x}-\hat{\boldsymbol{\mu}}^*\right).$$

Without selecting the smallest determinant of covariance, we choose to select a certain number of subsamples to achieve the variable selection through a sort of voting process. The portion of the most frequently appearing variables are elected to build an optimal space that allow us to compute our robust estimators. The simulation results and other details will be discussed later.

2.3. Justification Random Subspace Learning for Outlier Detection

Conjecture 1. *Let* $\mathcal{D}$ *be the dataset under consideration. Assume that a proportion* ε *of the observations in* $\mathcal{D}$ *are outliers. If* $\varepsilon < e^{-1}$, *then with high probability, the proposed RSSL outlier detection algorithm will efficiently correctly identify a set of data that contains very few of the outliers.*

Sketch 1. Let $\boldsymbol{x}_i \in \mathcal{D}$ be a random observation in the original dataset $\mathcal{D}$. Let $\mathcal{D}^{(b)}$ denote the b th bootstrapped sample from $\mathcal{D}$. Let $\Pr\left[\boldsymbol{x}_i \in \mathcal{D}^{(b)}\right]$ represent the proportion of observations that are in $\mathcal{D}$ but also present in $\mathcal{D}^{(b)}$. It is easy to prove

$$\Pr\left[\boldsymbol{x}_i \in \mathcal{D}^{(b)}\right] = 1 - \left(1 - \frac{1}{n}\right)^n. \tag{13}$$

In other words, if $\Pr\left[\boldsymbol{x}_i \notin \mathcal{D}^{(b)}\right] = \Pr\left[O_n\right]$ denotes the observations from $\mathcal{D}$ not present in $\mathcal{D}^{(b)}$, we must have

$$\Pr\left[\boldsymbol{x}_i \notin \mathcal{D}^{(b)}\right] = \left(1 - \frac{1}{n}\right)^n = \Pr\left[O_n\right]. \tag{14}$$

Since $\Pr\left[O_n\right]$ is known to converge to e^{-1} as n goes to infinity. Therefore for each given bootstrapped sample $\mathcal{D}^{(b)}$, there is a probability close to e^{-1} that any given outlier will not corrupt the estimation of location vector and scatter matrix parameters. Since the outliers as well as all other observations have an asymptotic probability of e^{-1} of not affecting the bootstrapped estimator that we build. Therefore over a large enough re-sampling process (large B), there will be many bootstrapped samples $\mathcal{D}^{(b)}$ with very few outliers leading to a sequence of small covariance determinants as desired, if $\varepsilon < e^{-1}$. It is therefore reasonable to deduce that by averaging this exclusion of outliers over many replications, robust estimators will naturally be generated by the RSSL algorithm.

2.4. Alternatives to Parametric Outlier Detection Methods

The assumption of multivariate Gaussianity of the $\boldsymbol{x}_i$'s is obviously limiting as it could happen that the data does not follow a Gaussian distribution. Outside of the realm where location and scatter matrix play a central role, other methods have been proposed, especially in the field of machine learning, and specifically with similarity measures known as kernels. One such method is known as One-Class Support Vector Machine (OCSVM) proposed by [16] to solve the so-called novelty detection problem. It is important to emphasize right away that novelty detection although similar in spirit to outlier detection, can be quite different when it comes to the way the algorithms are trained. OCSVM approach to novelty detection is interesting to mention here because despite some conceptual differences from the covariance methods explored earlier, it is formidable at handling HDLSS data thanks to the power of kernels. Let $\boldsymbol{x}_i \Phi : \mathcal{X} \to \mathcal{F}$. The one-class SVM novelty detection solves

$$\underset{\boldsymbol{w}\in\mathcal{F},\boldsymbol{\xi}\in\mathbb{R}^n,\rho\in\mathbb{R}}{\operatorname{argmin}} \left\{\frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho\right\}, \tag{15}$$

subject to

$$\left\langle \boldsymbol{w}, \Phi\left(\boldsymbol{x}_i\right)\right\rangle > \rho - \xi_i,\ \xi_i \geq 0,\ i = 1,\cdots,n, \tag{16}$$

using $\mathcal{K}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \left\langle \Phi\left(\boldsymbol{x}_i\right), \Phi\left(\boldsymbol{x}_j\right)\right\rangle = \Phi\left(\boldsymbol{x}_i\right)^{\top}\Phi\left(\boldsymbol{x}_j\right)$ we get

$$\hat{f}\left(\boldsymbol{x}_i\right) = \operatorname{sign}\left(\sum_{j=1}^{n}\hat{\alpha}_j\mathcal{K}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) - \hat{\rho}\right), \tag{17}$$

so that any $\boldsymbol{x}_i$ with $\hat{f}\left(\boldsymbol{x}_i\right) < 0$ is declared an outlier. The $\hat{\alpha}_j$'s and $\hat{\rho}$ are determined by solving the quadratic programming problem formulated above. The parameter ν controls the proportion of outliers detected. One of the most common kernel is the so-called RBF kernel defined by

$$\mathcal{K}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \exp\left\{-\frac{1}{2\sigma^2}\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2\right\}. \tag{18}$$

OCSVM has been extensively studied and applied by many researchers among which [17]-[19], and later enhanced by [20]. OCSVM is often applied to semi-supervised learning tasks where training focuses on all the positive examples (non-outliers) and then the detection of anomalies is performed by searching points that fall geometrically outside of the estimated/learned decision boundary of the good (non-outlying trained instances). It is a concrete and quite popular algorithm for solving one-class problems in fields like digital recognition and documentation categorization. However, it is crucial to note that OCSVM cannot be used with many other real life datasets for which outliers are not well-defined and/or for which there are no clearly identified all-positive training examples available such as gene expression mentioned before.

3. Computational Demonstrations

3.1. Setup of Computational Demonstration and Initial Results

In this section, we conduct a simulation study to assess the performance of our algorithm based on various important aspects of the data, and we also provide a comparison of the predictive/detection performance of our method against existing approaches. All our simulated data are generated according to the ε-contaminated multivariate Gaussian introduced via Equation (1) and Equation (2). In order to assess the effect the covariance between the attributes, we use an AR-type covariance matrix of the following form

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & \rho & \cdots & \rho \\ \rho & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \rho \\ \rho & \cdots & \rho & 1 \end{pmatrix} = \left[(1-\rho)\boldsymbol{I}_p + \rho \mathbf{1}_p \mathbf{1}_p^{\top} \right]. \tag{19}$$

where $\boldsymbol{I}_p$ is the p-dimensional identity matrix, while $\mathbf{1}_p$ is p-dimensional vector of ones. For the remaining parameters, we consider 3 different levels of contamination $\varepsilon \in \{0.05, 0.1, 0.15\}$, namely mild contamination to strong contamination. The dimensionality p will increase in low-dimensional case as $\{30, 40, 50, 60, 70\}$ and high dimensional case as $\{1000, 2000, 3000, 4000, 5000\}$ and the number of observations are fixed at 1500 and 100. We compare our algorithm to existing PCA based algorithms PCOut and PCDist, both of which are available in Rwithin the package called rrcovHD.

As can be seen in **Figure 1**, the overwhelming majority of samples lead to determinants that are small as evidenced by the heavy right skewness with concentration around zero. This further confirms our conjecture that as long as $\varepsilon < e^{-1}$ which is a rather reasonable and easily realized assumption, we should isolate samples with few or no outliers.

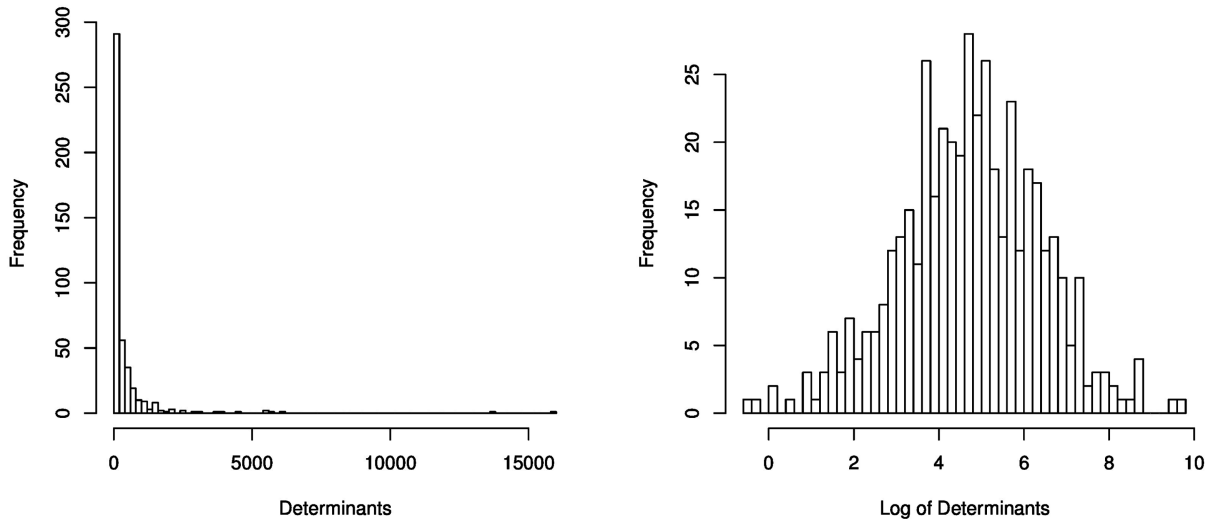

Figure 1. (left) Histogram of the distribution of the determinants from $\mathcal{D}_{sub}^{(b)}$ when n = 100, p = 3000; (right) Histogram of log determinants for all the bootstrap samples.

Since each bootstrapped sample selected has a small chance of being affected by the outliers, we can select the dimensionality that maximize this benefits. In our HDLSS simulations, determinants are computed based on all the randomly selected subspaces, and are ruled by predominantly small values, which imply the robustness of the classifier. **Figure 1** patently shows the dominance of small values of determinants, which in this case are the determinants of all bootstrapped samples based on our simulated data. A distinguishable elbow is presented in **Figure 2**. The next crucial step lies in selecting a certain number of bootstrap samples, say *k*, to build an optimal subspace. Since most of the determinants are close to each other, it is a non-trivial problem, which means that *k* needs to be carefully chosen to avoid going beyond the elbow. However, it is important to notice if *k* is too small then the variable selection in later steps of the algorithm will become a random pick, because there is no opportunity for each variable to appear in the ensemble. Here, we choose *k* to be the number of roughly the first 30% to 80% of *B* bootstrap samples $\mathcal{D}^{(\eta)}$ according to their ascending order of the determinants. This choice is based on our empirical experimentations. It is not too difficult to infer the asymptotic normal distribution of the frequencies of all variables in $\mathcal{D}^{(\eta)}$ as we can observe in **Figure 2**. Thus, the most frequently appearing variables located on the left tail can be adopted/kept to build our robust estimator. Once the selection of *k* is made, the frequencies of variables appearing in this ensemble can be obtained/computed for variable selection. The 2 to *m* most frequently appearing variables are included to compute the determinants in **Figure 2**. *m* is usually small, since we assume from the start that the true dimensionality of the data is indeed small. Here for instance, we choose 20 for the purposes of our computational demonstration. A sharp maximum indicates the number of dimension ν from that sorted ensemble that we need to choose. Thus, with the bootstrapped observations having the smallest determinant with the subspace that generates the largest determinant, we can successfully compute $\mathcal{D}^* = \mathcal{D}_{sub=\nu}^{(\eta=1)}$. Then the robust estimators can be formed by $\hat{\boldsymbol{\mu}}^*$ and $\hat{\boldsymbol{\Sigma}}^*$. Theoretically then we are in a presence of a minimax formulation of our outlier detection problem, namely

$$\left\{\mathcal{D}^{(*)}, \mathcal{V}^{(*)}\right\} = \underset{\mathcal{V}^{(b)}}{\operatorname{argmax}} \left\{ \underset{\mathcal{D}^{(b)}}{\operatorname{argmin}} \left\{ \det\left(\operatorname{cov}\left(\hat{\boldsymbol{\Sigma}}\left(\mathcal{D}^{(b)}\left(\mathcal{V}^{(b)} \right) \right) \right) \right) \right\} \right\}. \tag{20}$$

By Equation (20), it should be understood that we need to isolate the precious subsample $\mathcal{D}^{(*)}$ that achieves the smallest overall covariance determinant, but then concurrently identify along with $\mathcal{D}^{(*)}$ the subspace $\mathcal{V}^{(*)}$ that yields the highest value of that covariance determinant among all the possible subspaces considered.

3.2. Further Results and Computational Comparisons

As indicated in our introductory section, we use the Mahalanobis distance as our measure of proximity. As since we are operating under the assumption of multivariate normality, we use the traditional distribution quantiles $\chi^2_{d,\alpha/2}$ as our cut-off with the typical $\alpha = 10\%$ and $\alpha = 10\%$. As usual, all observations with distances larger

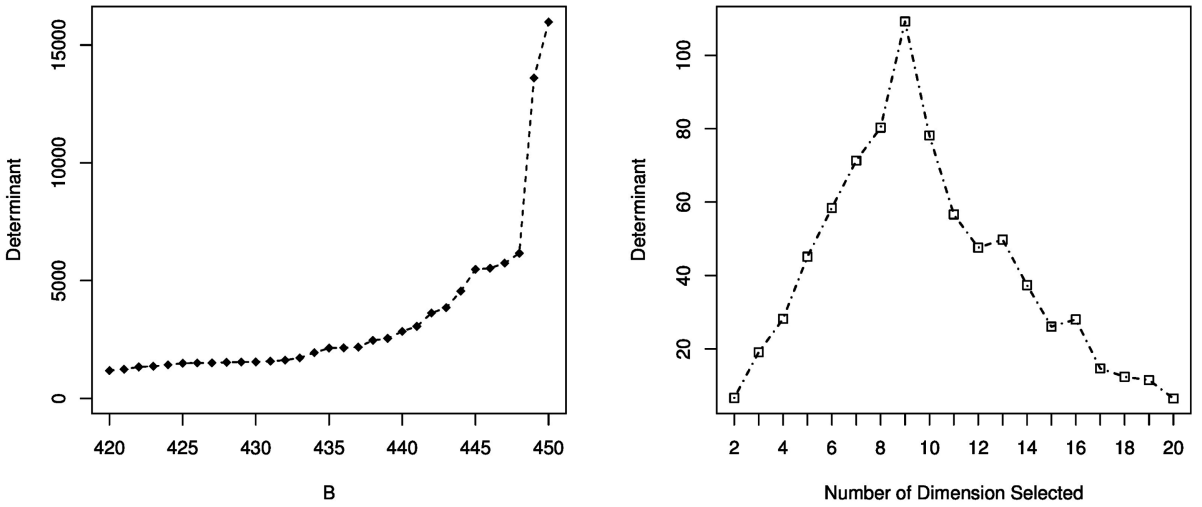

Figure 2. (left) Tail of sorted determinants in high dimensional $\mathcal{D}_{sub}^{(b)}$, where *B* = 450. *k* can be selected before reaching the elbow; (right) The concave shape can be observed by computing determinants of covariance from 2 to m dimension.

than $\chi^2_{d,\alpha/2}$ are classified as outliers. The data for simulation study are generated with $\eta,\kappa\in\{2,5\}$ representing both easy and hard situation for RSSL algorithm to detect the outliers, and ε, p as the rate of contamination. Throughout, we use $R=200$ replications for each combination of parameters for each algorithm, and we use the average test error AVE as our measure of predictive/detection performance. Specifically,

$$\mathrm{AVE}\left(\hat{f}\right)=\frac{1}{R}\sum_{r=1}^{R}\left\{\frac{1}{m}\sum_{i=1}^{m}\ell\left(\boldsymbol{y}_i^{(r)},\hat{f}_r\left(\boldsymbol{x}_i^{(r)}\right)\right)\right\}, \tag{21}$$

where $\hat{f}_r\left(\boldsymbol{x}_i^{(r)}\right)$ is the predicted label of the test set observation i yielded by f in the r-th replication. The loss function used here is the basic zero-one loss defined by

$$\ell\left(\boldsymbol{y}_i^{(r)},\hat{f}_r\left(\boldsymbol{x}_i^{(r)}\right)\right)=1_{\left\{\boldsymbol{y}_i^{(r)}\neq\hat{f}_r\left(\boldsymbol{x}_i^{(r)}\right)\right\}}=\begin{cases}1 & \boldsymbol{y}_i^{(r)}\neq\hat{f}_r\left(\boldsymbol{x}_i^{(r)}\right)\\ 0 & \text{otherwise.}\end{cases} \tag{22}$$

It will be seen later that our proposed method produces predictive accurate outlier detection results, typically competing favorably against other techniques, and usually outperforming them. Firstly however, we show in **Figure 3** the detection performance of our algorithm based on two randomly selected subspaces. The outliers detected by our algorithm are identified by red triangles and contained in the red contour, while the black circles are the normal data.

The improvement of our random subspace learning algorithm in low dimensional data with dimensionality such that $p\in\{30,40,50,60,70\}$ and relative large sample size $n=1500$, is demonstrated in **Figure 4** in comparison to PCOut and PCDist. Given a relatively easy task, namely with $\kappa,\eta=5$, the outliers are scattered widely and shifted far from normal, the RSSL with $1-\alpha$ equals 95% and 90% perform consistently very well, typically outperforming the competition. When the rate of contamination is increasing in this scenario, almost 100% accuracy can be achieved with RSSL based algorithm. When the outliers are spread more narrowly and closer to the mean with $\kappa,\eta=2$, the predictive accuracy of our random subspace based algorithm is slightly less powerful but still very strong, namely with $p\in\{1000,2000,3000,4000,5000\}$ a predictive detection rate close to 96% to 99%. In high dimensional settings, namely with and low sample size $n=100$, RSSL is also performs reasonably well as shown in **Figure 5**. With $1-\alpha=95\%$ chi-squared cut-off, when $\kappa,\eta=5$, 96% to 98% of outliers can be detected constantly among all simulated high dimensions. Under more difficult conditions, as with $\kappa,\eta=2$, a decent amount of outliers can be detected with accuracy around 92% to 96%. Based on the properties of robust PCA based algorithms, the situation that we define as "easy" for RSSL algorithms is actually "harder" for PCOut and PCDist. The principle component space is selected based on the visibility of outliers, and especially for PCOut, the components with nonzero robust kurtosis are assigned higher weights by

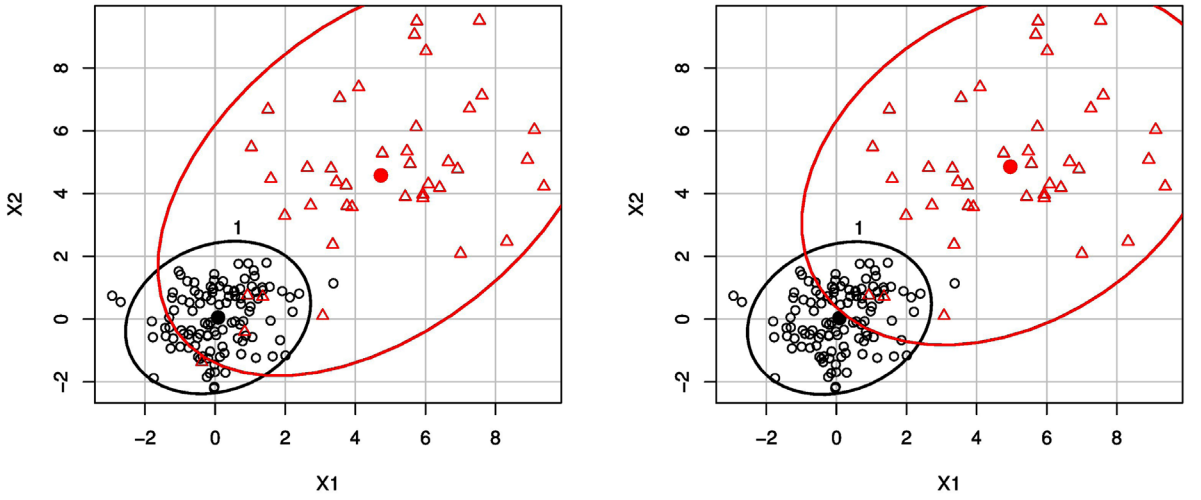

Figure 3. (left) The outliers detected in a two dimensional subspace are marked as red triangles. Selection is based on $\chi^2_{df=d,\alpha=5\%}$.

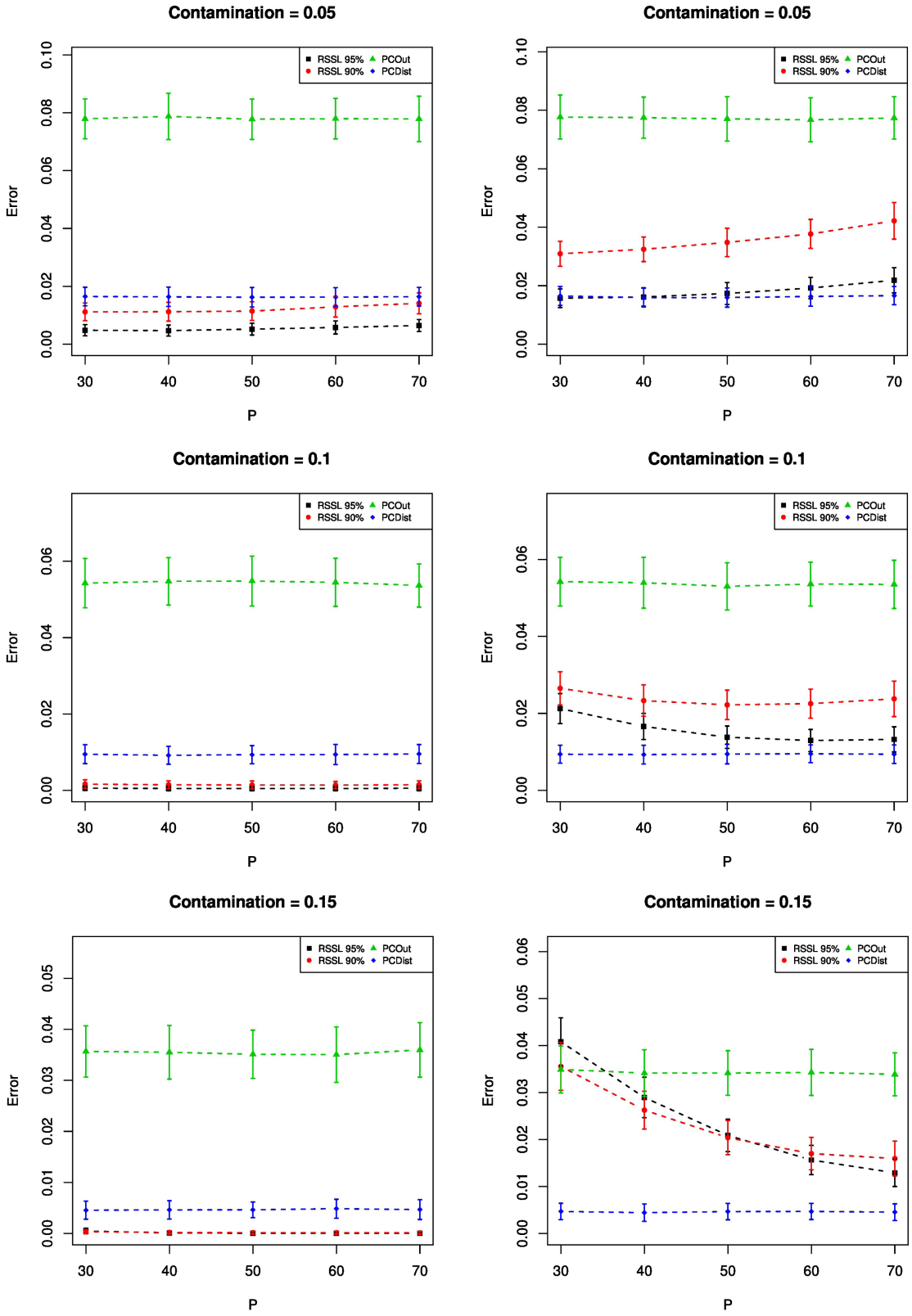

Figure 4. The average error and standard deviation in low dimensional simulation with $\kappa, \eta = 5$ (left column) and $\kappa, \eta = 2$ (right column).

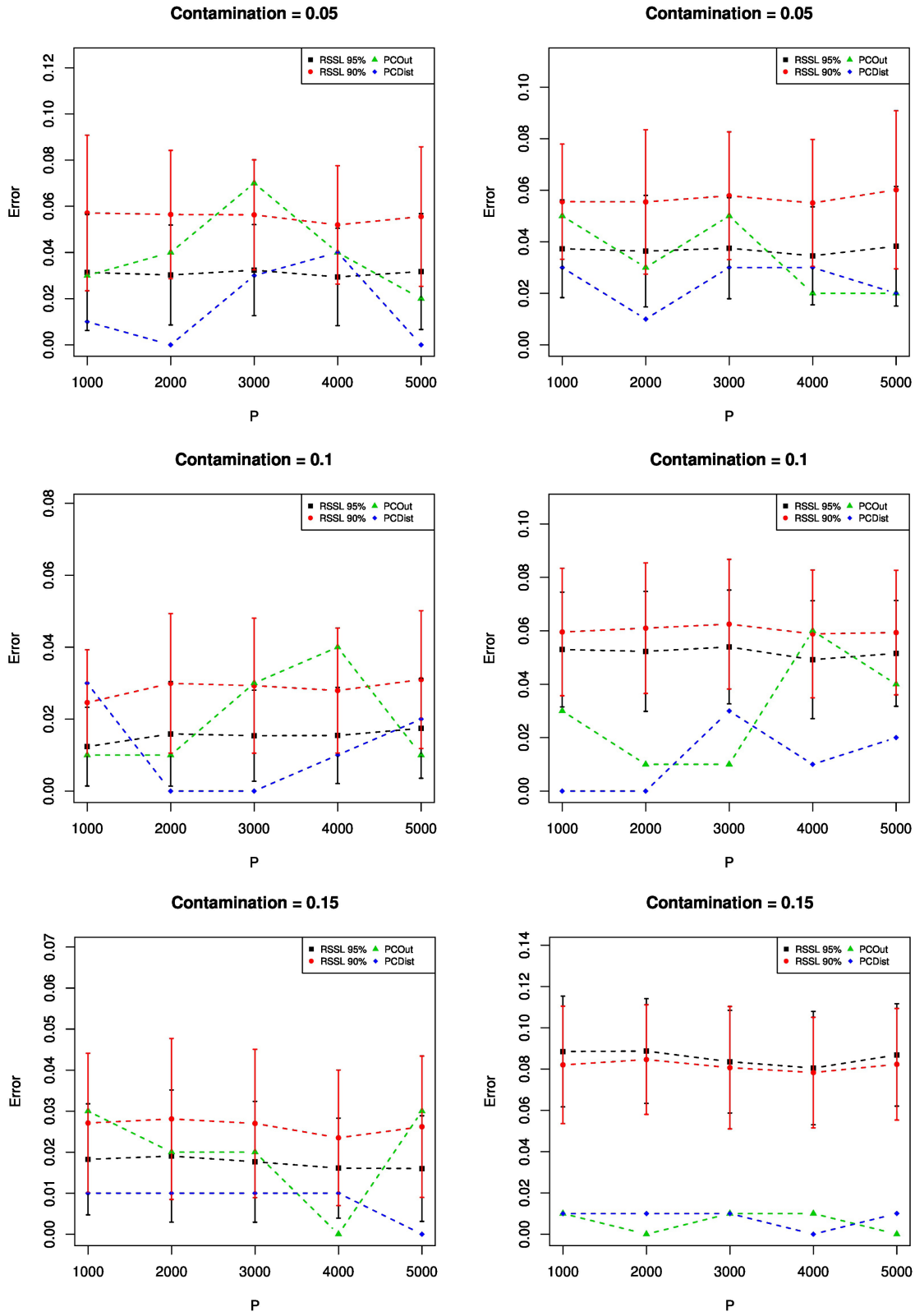

Figure 5. The average error and standard deviation in high dimensional simulation with $\kappa,\eta=5$ (left column) and $\kappa,\eta=2$ (right column).

the absolute value of their kurtosis coefficients. This method is shown to yield good performances when dealing with small shift of mean and scatter of the covariance matrix. However, if the outliers lied on larger η and κ where excessive choices can be made then, it is more difficult for PCA to find the dimensionality to make the outliers "stick out". Reversely, with a small values of κ and η, the most obvious directions are emphasized by PCA but less chance for algorithms like RSSL to obtain the most sensible subspace to build robust estimators. So in **Figure 5**, when $\kappa, \eta = 2$ the accuracy reduced to around 92% but in all other high-dimensional settings the performance of RSSL is consistent with PCOut and identically stable.

4. Conclusion

We have presented what we can rightfully claim to be a computational efficient, scalable, intuitive appealing and highly predictively accurate outlier detection method for both HDLSS and LDHSS datasets. As an adaptation of both random subspace learning and minimum covariance determinant, our proposed approach can be readily used on vast number of real life examples where both its component building blocks have been successfully applied. The particular appeal of the random subspace learning aspect of our method comes in handy for many outlier detection tasks on high dimension low sample size datasets like DNA Microarray Gene Expression datasets for which the MCD approach is proved to be computational untenable. As our computational demonstrations section above reveal, our proposed approach competes favorably with other existing methods, sometimes outperforming them predictively despite its straightforwardness and relatively simple implementation. Specifically, our proposed method is shown to be very competitive for both low dimensional space and high dimensional space outlier detection and is computationally very efficient. We are currently seeking out interesting real life datasets on which to apply our method. We also plan to extend our method beyond settings where the underlying distribution is Gaussian.

Acknowledgements

Ernest Fokoué wishes to express his heartfelt gratitude and infinite thanks to our lady of perpetual help for her ever-present support and guidance, especially for the uninterrupted flow of inspiration received through her most powerful intercession.

References

[1] Rousseeuw, P.J. (1984) Least Median of Squares Regression. *Journal of the American Statistical Association*, **79**, 871-880. http://dx.doi.org/10.1080/01621459.1984.10477105

[2] Rousseeuw, P. and Van Driessen, K. (1999) A Fast Algorithm for the Minimum Covariance Determinant Estimator. *Technometrics*, **41**, 212-223. http://dx.doi.org/10.1080/00401706.1999.10485670

[3] Filzmoser, P., Maronna, R. and Werner, M. (2008) Outlier Identification in High Dimensions. *Computational Statistics & Data Analysis*, **52**, 1694-1711. http://dx.doi.org/10.1016/j.csda.2007.05.018

[4] Fritsch, V., Varoquaux, G., Thyreau, B., Poline, J.-B. and Thirion, B. (2011) Detecting Outlying Subjects in High-Dimensional Neuroimaging Datasets with Regularized Minimum Covariance Determinan 225. In: Fichtinger, G., Martel, A. and Peters, T., Eds., *Medical Image Computing and Computer-Assisted Intervention MICCAI* 2011, Springer, Berlin Heidelberg, 264-271.

[5] Angiulli, F. and Pizzuti, C. (2002) Fast Outlier Detection in High Dimensional Spaces. In: Tapio, E., Heikki, M. and Hannu, T., Eds., *Principles of Data Mining and* 230 *Knowledge Discovery*, Springer, Rende, 15-27. http://dx.doi.org/10.1007/3-540-45681-3_2

[6] Aggarwal, C. and Yu, S. (2005) An Effective and Efficient Algorithm for High-Dimensional Outlier Detection. *The VLDB Journal*, **14**, 211-221. http://dx.doi.org/10.1007/s00778-004-0125-5

[7] Ghoting, A., Parthasarathy, S. and Otey, M.E. (2008) Fast Mining of Distance-Based Outliers in High-Dimensional 235 Datasets. *Data Mining and Knowledge Discovery*, **16**, 349-364. http://dx.doi.org/10.1007/s10618-008-0093-2

[8] Kriegel, H.-P., Kröger, P., Schubert, E. and Zimek, A. (2009) Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data. In: Editor, Ed., *Advances in Knowledge Discovery and Data Mining*, Springer, München, 831-838. http://dx.doi.org/10.1007/978-3-642-01307-2_86

[9] Coppersmith, D. and Winograd, S. (1990) Matrix Multiplication via Arithmetic Progressions. *Journal of Symbolic Computation*, **9**, 251-280. http://dx.doi.org/10.1016/S0747-7171(08)80013-2

[10] Le Gall, F. (2014) Powers of Tensors and Fast Matrix Multiplication. *Proceedings of the* 39*th International Symposium*

on Symbolic and Algebraic Computation, New York, 23-25 July 2014. http://dx.doi.org/10.1145/2608628.2608664

[11] Hubert, M. and Engelen, S. (2004) Robust PCA and Classification in Biosciences. *Bioinformatics*, **20**, 1728-1736. http://dx.doi.org/10.1093/bioinformatics/bth158

[12] Croux, C. and Ruiz-Gazen, A. (1996) A Fast Algorithm for Robust Principal Components Based on Projection Pursuit. In: Prat, A., Ed., *COMPSTAT*, Springer, Heidelberg, 211-216. http://dx.doi.org/10.1007/978-3-642-46992-3_22

[13] Li, G.Y. and Chen, Z.L. (1985) Projection-Pursuit Approach to Robust Dispersion Matrices and Principal Components: Primary Theory. *Journal of the American Statistical Association*, **80**, 759-766. http://dx.doi.org/10.1080/01621459.1985.10478181

[14] Ho, T.K. (1998) The Random Subspace Method for Constructing Decision Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 832-844. http://dx.doi.org/10.1109/34.709601

[15] Lopuhaa, H.P. and Rousseeuw, P.J. (1991) Breakdown Points of Affine Equivariant Estimators of Multivariate Location and Covariance. *The Annals of Statistics*, **19**, 229-248. http://dx.doi.org/10.1214/aos/1176347978

[16] Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J. and Williamson, R.C. (1999) Estimating the Support of a High-Dimensional Distribution. *Neural Computation*, **13**, 1443-1471.

[17] Hubert, M., Rousseeuw, P.J. and VandenBranden, K. (2005) Robpca: A New Approach to Robust Principal Component Analysis. *Technometrics*, **47**, 64-79. http://dx.doi.org/10.1198/004017004000000563

[18] Manevitz, L.M. and Yousef, M. (2002) One-Class SVMs for Document Classification. *The Journal of Machine Learning Research*, **2**, 139-154.

[19] Zhang, R., Zhang, S., Muthuraman, S. and Jiang, J. (2007) One Class Support Vector Machine for Anomaly Detection in the Communication. *Proceedings of the* 5*th Conference on Applied Electromagnetics*, *Wireless and Optical Communications*, *ELECTROSCIENCE*'07, 14-16 December 2007, Tenerife, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, 31-37.

[20] Amer, M., Goldstein, M. and Abdennadher, S. (2013) Enhancing One-Class Support Vector Machines for Unsupervised Anomaly Detection. *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, *ODD*'13, ACM, New York, 2013, 8-15. http://dx.doi.org/10.1145/2500853.2500857