5-2016

Random Subspace Learning on Outlier Detection and Classification with Minimum Covariance Determinant Estimator

Bohan Liu
bl3267@rit.edu

Random Subspace Learning on Outlier Detection and Classification with Minimum Covariance Determinant Estimator

by

Bohan Liu

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science
in Applied Statistics
School of Mathematical Sciences
College of Science

Supervised by

Professor Dr. Ernest Fokoué
Department of Mathematical Sciences
School of Mathematical Sciences
College of Science
Rochester Institute of Technology
Rochester, New York
May 2016

Approved by:

Dr. Ernest Fokoué, Professor
Thesis Advisor, School of Mathematical Sciences

Dr. Steven LaLonde, Associate Professor
Committee Member, School of Mathematical Sciences

Dr. Joseph Voelkel, Professor
Committee Member, School of Mathematical Sciences

Thesis Release Permission Form

Rochester Institute of Technology
School of Mathematical Sciences

Title:

Random Subspace Learning on Outlier Detection and Classification with Minimum Covariance Determinant Estimator

I, Bohan Liu, hereby grant permission to the Wallace Memorial Library to reproduce my thesis in whole or part.

Bohan Liu

Date

To Jiapei and my parents

Acknowledgments

I am unspeakably grateful for my thesis advisor Dr. Ernest Fokoué for his generosity of sharing all his deep and profound knowledge with me. I deeply appreciate the effort he put through to help many of the students, including me, to chase their dreams. His kindness and enthusiasm are carved into my mind forever.

I wish to thank Dr. Steven LaLonde and Dr. Joseph Voelkel for their kindly acceptance of being on my thesis committee and their precious advises to my research thesis.

I wish to sincerely thank Xingchen Yu and Zichen Ma for encouraging and helping me during my studies. I will treasure our friendship for my life.

I would give my deepest thanks to my parents and my wife, who are always unconditionally supporting me in accomplishing this master's degree and my Ph.D. application.

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

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

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

Abstract

Random Subspace Learning on Outlier Detection and Classification with Minimum Covariance Determinant Estimator

Bohan Liu

Supervising Professor: Dr. Ernest Fokoué

The questions brought by high dimensional data is interesting and challenging. Our study is targeting on the particular type of data in this situation that namely “large p, small n”. Since the dimensionality is massively larger than the number of observations in the data, any measurement of covariance and its inverse will be miserably affected. The definition of high dimension in statistics has been changed throughout decades. Modern datasets with over thousands of dimensions are demanding the ability to gain deeper understanding but hindered by the curse of dimensionality. We decide to review and explore further to negotiate with the curse and extend previous studies to pave a new way for estimating robustness then apply it to outlier detection and classification.

We explored the random subspace learning and expand other classification and outlier detection algorithms to adapt its framework. Our proposed methods can handle 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.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Difficulties

The beginning point of our work is mapped from our curiosity of robust estimators in modern data-driven decisions. One can naturally connect this topic to outlier detection and location estimation. Many studies in this field had brought various "thick-skinned" properties to attention since Box[19] used the word "robustness" to describe the insensitive of violation of normality in Bartlett's[10] version of Neyman-Pearson[111] likelihood ratio test. Modern datasets brought us huge amount of challenges that not only because they consume massive computational resource due to their exponentially increasing scales but also expand themselves to extreme structures such as "short-fat", so called **High Dimensional Low Sample Size** (HDLSS). Early studies seem to address this issue rarely but enormous attentions have been drawn throughout this decade. So we ask ourselves this question:*Can we build robust estimators that can adapt to high dimensional data?* With this question in mind, we notice there are several points can not be ignored. First of all, we are not solely focusing on certain applications but interested in using statistical machine learning to build estimators that can also be applied in various needs. The results of the algorithm can be used and adjusted in tasks like outlier detection and classification. Second, our target will not only be those extreme scenarios but also other typical statistical situations like Boston Housing and Iris datasets. The estimation method can have roots in lower dimensional space then expand its stems and leaves to survive in harsher environment. Finally, the computational complexity should also be considered as a major factor especially with high-dimensional problem. If covariance matrices or their inverse with massive amount of variables are encountered, the time used in estimation will increase cubically with the dimensionality then the task will simply be impractical.

To follow the direction that we mentioned earlier, there are vast number of routes to explore. A very intuitive procedure can start from techniques like dimensionality reduction or feature selection to uncover the structure of the meaningful proportion. Then from this seemingly more rational base we may construct our new estimators or models to travel through the patterns hidden in the data. Thus, the core problems that have to be solved to guide us can be summarized by:

- What are the techniques used to reduce the dimensionality?
- What are the techniques have been applied on subspace to achieve different goals?
- How we can extract the essence of previous studies to hit our targets?
- How is the performance if we eventually applied our estimations to different applications?

Unfolding these four questions can clarify the goal of our research: *Reduce the dimensionality efficiently then extend or combine previous algorithms to build robust estimators for both high and low dimensional data*. After we acquired our estimators we can compare their performances to other algorithms in applications such as outlier detection and classification.

1.2 Background: The Curse of Dimensionality

1.2.1 Where Comes the Curse

Introduced by Bellman[12] in 1957, the term curse in machine learning is mainly used to describe the explosively increasing complexity with each variable added in higher dimension. Given a value of smooth function defined in a high dimensional space, it is very likely the convergence rate of the estimator will be inevitably slow. Although the term is often related to the poor performance of classical algorithms especially for non-parametric ones like nearest neighbors and Gaussian kernel, the true difficulties come from its deep uncertainty within. It is like someone dropped a key when he was walking through a narrow alley that all he needs to do to find the key is just to walk in an opposite direction. But if the key was lost on a golf course it will be almost impossible to retrieve it. Some properties of high dimensional space has been demonstrated in previous studies[9] [35] [89] and many of them are speechlessly counter-intuitive. Human beings are deaf and blind in the universe, not only because the range of frequency or spectrum we can hear and see but also because we can never possibly imagine adding even one more dimension to our existing world.

Unfortunately, among all these cruel situations there are some extremists can be easily encountered frequently. These are the datasets we mentioned before as HDLSS and often being labeled as "large p, small n". To be more explicitly, given the data $\mathscr{D} = \left[\mathbf{x}_1^\top, \mathbf{x}_2^\top, \cdots, \mathbf{x}_n^\top\right]$, where $\mathbf{x}_i = \left(x_{i1}, \cdots, x_{ip}\right)^\top \in \mathbb{R}^{1\times p}$:

$$\boldsymbol{X} = \begin{pmatrix} \mathbf{x}_{11} & \mathbf{x}_{12} & \cdots & \cdots & \mathbf{x}_{1p} \\ \mathbf{x}_{21} & \mathbf{x}_{22} & \cdots & \cdots & \mathbf{x}_{2p} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{x}_{n1} & \mathbf{x}_{n2} & \cdots & \cdots & \mathbf{x}_{np} \end{pmatrix} \tag{1.1}$$

with $p \gg n$. Typical examples can be found in many fields especially for computational biology and computer vision where plenty of datasets are considered as benchmarks. In bioinformatics, such instances are considered as daily basis in so called micro-array gene expression. Micro-array is a technology that using silicon bio-chip with tens of thousands of preselected gene spots to collect and measure gene expression from biological samples. Later the data is cleaned and normalized through some database search algorithms that huge number of variables are generated after this pre-processing step. Famous datasets including "leukemia", "lymphoma" and "colon-cancer" mainly come from previous cancer studies[3] [4] [58] where researchers were trying to find statistical patterns that can classify different tumors. Among these datasets, a relative small number of variables are around 2000 (colon-cancer dataset). But the dataset contains only 62 observations and makes the ratio between the number of observation and dimensionality n/p equals 0.031. One of the noticeable examples in computer vision and image processing is ICDAR2013, a competition of gender prediction from handwriting posted on Kaggle (a data science competition platform). There were only 475 observations were provided but the number of features extracted from four documents went up to 28000. Traditional methods usually fail sorrowfully in these cases due to multiple properties of the curse. Later in this thesis, we will often use p to refer the number of dimensions or variables and n to refer the number of observations or examples in the data.

1.2.2 What Comes With the Curse

Classical approaches in finite dimensional space fail in different ways. Any statistical method that needs to compute the inverse of covariance matrix fails immediately. Some of the attempts[125] [159] of approximating the inverse, though indeed reduced the computational expenses, are still in development and have not been practically implemented. Several noticeable properties of the curse can be summarized below:

- The available sample points are going to be inevitably sparse in high dimensional space.
- Most of the data are severely pushed far from the centre in high dimensional space.
- Distance functions may largely lose their meaning in high dimensional space.
- The major proportion of the data is very likely to be noisy or highly correlated in high dimensional space.
- The number of models is growing ruthlessly in high dimensional space.

The sparsity maybe the most intuitive problem that one can think of. Imagine if 10 points need to be sampled from an line interval from 0 to 1 in 1-dimensional coordinate system. This means with each dimension add to the original space that 10^p points need to be sampled from p dimensional space. In the colon-cancer dataset we mentioned before, it may seem to be very crowded if all 62 observations lined on a one dimensional line interval. With the exponential increasing in volume caused by adding other 1999 dimensions, majority of the sample points may isolated from each other.

The skewness of the data from centre is often demonstrated by the ratio between an hypersphere and a hypercube. The side length of the hypercube equals the diameter of the hypersphere. It is relatively fair to imagine that in two or three dimensional space that data may equally spread in both of the shapes if the points are “randomly” distributed. However, increasing the dimensionality of both of them to a slightly larger number, say 10, the volume of the hypersphere collapses sharply towards 0 as in figure (1.1). Then majority of the data are squeezed to the edges of the hypercube where far from the centre. Formally, the volume of a p-ball:

$$V_p = \frac{r^p \pi^{p/2}}{\Gamma(p/2+1)} \longrightarrow 0, \quad as \quad p \longrightarrow \infty \tag{1.2}$$

where r is the radius of the hypersphere. To make this even more counter-intuitive, the volume of a unit hypercube remains 1 as the the dimensionality goes to infinity. If the length of the side is less than 1 the volume approaches 0 and if length of the side is larger than 1 the volume turns out to be infinity. The shape of the hypercube is commonly visualized as a sea urchin where majority of the data are located on its "spikes" as in (1.2).

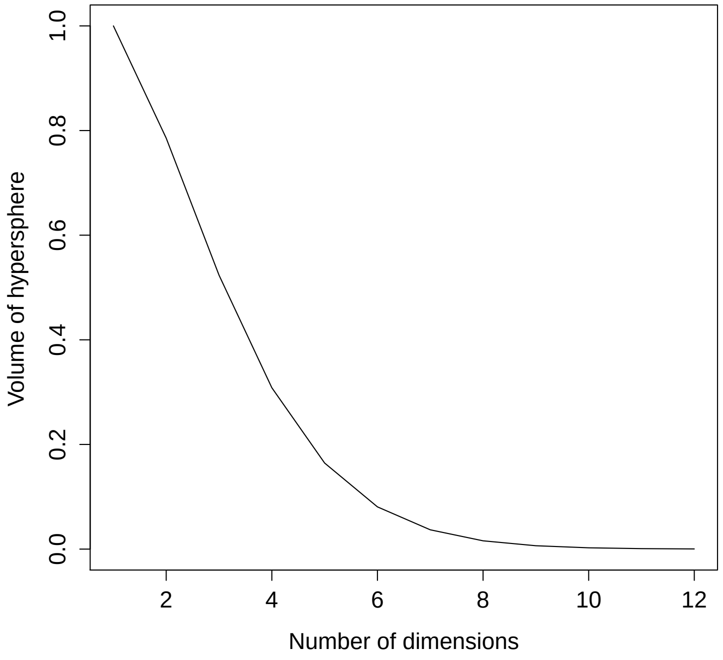

Figure 1.1: The volume of hypersphere with diameter equals 1 decreases sharply as the number of dimensions increases.

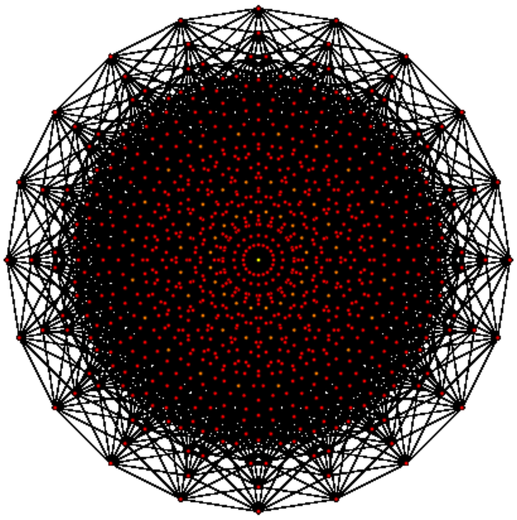

Figure 1.2: Orthogonal projection of a 10 dimensional hypercube

Another consequential effect is many of the distance measures in machine learning start to drop their meaningfulness. As Beyer *et al.*[15] showed under certain conditions, the ratio between the variance of the distance measure of any given data point and the variance of the mean distance measure of the distance is converging to zero as dimensionality goes to infinity. So a little bit more formally, we have:

$$\lim_{p\to\infty} var\left(\frac{(f_p(X_i))^d}{\mathbb{E}\left((f_p(X_i))^d\right)}\right) = 0, \tag{1.3}$$

Then for every $\epsilon > 0$,

$$\lim_{p\to\infty} Pr\left[max\left(dist_p(X_i)\right) \leqslant (1+\epsilon)min\left(dist_p(X_i)\right)\right] = 1, \tag{1.4}$$

where d is a constant that $d \in (0, \infty)$. Given dimensionality p, f_p is a function of data X that inputs a data point X_i where $i = 1, 2, \cdots, n$ from both query and data domain that output a non-negative real number. As a result, if the dimensionality inflates to infinity, the proportion of the difference between the maximum distance and the minimum distance from the centroid collapse to zero. Thus, the meaning of many distance measures becomes in doubt. Vast number of machine learning algorithms which relied on the distances like Mahalanobis distance, Manhattan distance *etc.* may generate invalid results. Imaginably, the dimensionality can also affect likelihood compute from Gaussian and make it skew towards to higher dimensions. One of the famous examples can be found is the outlier detection algorithm based on the Nearest Neighbors proposed by Ramaswamy *et al.*[124] in 2000. Later the lack of contract phenomena of distances for any given data point was shown in relevant research of Zimek *et al.*[163] in 2013 by asymptotically computing and comparing the minimum and maximum distances of simulated Gaussian and uniform data from lower to higher dimensions. A figure of maximum distance divided by the minimum distance for multivariate Gaussian can be found in fig (1.3).

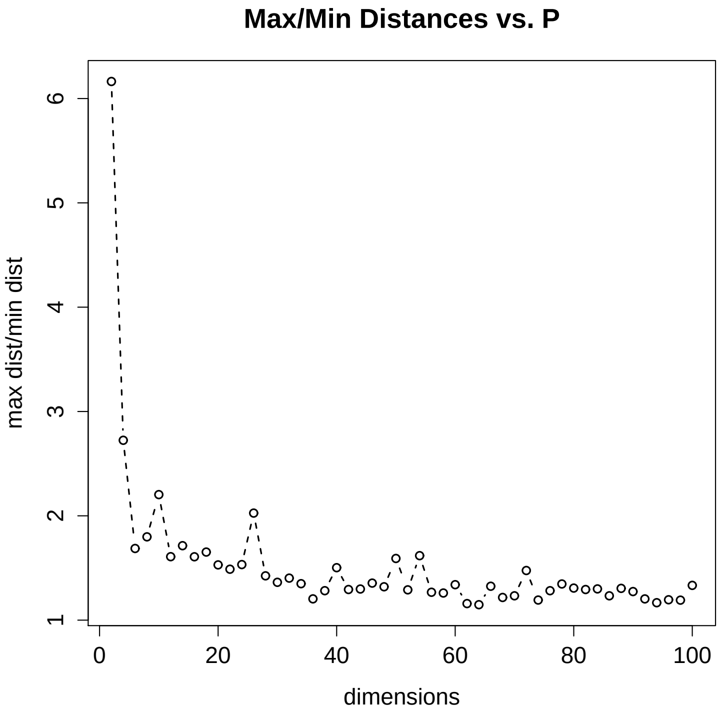

Figure 1.3: The ratio between max-distance and min-distance vs. dimensionality from multivariate Gaussian when $n = 10$

For a finite lower dimensional dataset, students are taught to watch the multicollinearity while learning multiple linear regression by plotting the correlation matrix to pairwisely check variables that are highly correlated with each other. However, this is not the first time common sense has been greatly challenged by simply introduce an ultra-high dimensionality. The probability of multicollinearity can be largely amplified with p increasing. Fan and Lv[40] showed the maximum sample correlation and multiple correlation are frequently occurred even samples are drawn from independent Gaussian variables in higher dimensions (p equals 10^3 and 10^4). This implies that the noisy variables can be very deceive in high dimensional space especially when $p \gg n$. Our truly relevant variables sometimes may be represented by the combination of or, replaced by noise and then associate with responses. Thus the fitted model looks like putting the earth's land surface on a single rope string, theoretically, if nobody moves. Then the world can be destroyed with a slightest breath just because of its massively inflated variance.

A simple example can be illustrated in multiple linear regression model:

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \mathcal{E}, \quad where\, \mathcal{E} \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{\sigma}^2 \boldsymbol{I}), \tag{1.5}$$

the variance of an individual prediction $\hat{\mathbf{y}}_i$ given a new observation $\mathbf{x}_i$ that $\mathbf{x}_i = (x_{i1}, \cdots, x_{ip})^\top \in \mathbb{R}^{1\times p}$ can be expressed as:

$$var\left(\hat{\mathbf{y}}_i \mid \mathbf{x}_i\right) = \mathbf{x}_i^\top var\left(\hat{\boldsymbol{\beta}}\right)\mathbf{x}_i + var\left(\mathcal{E} \mid \mathbf{x}_i\right) \tag{1.6}$$

$$= \hat{\sigma}^2\left(\mathbf{x}_i^\top (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\mathbf{x}_i + 1\right) \tag{1.7}$$

If p increases drastically, more and more eigenvalues of $\boldsymbol{X}^\top \boldsymbol{X}$ starts to reach zero and its inverse is lack of boundaries. Thus, as a consequence $var\left(\hat{\mathbf{y}}_i \mid \mathbf{x}_i\right)$ becomes infinity.

Even if somehow all of the problems mentioned above do not hold, the number of models to estimate and parameters to be selected is devastating. For regression, parsimony is usually one of the first lessons in model selection but the simplest model can still knock people off by checking how many significance test they need to compute. Fokoué *et al.*[25] in his book described the explosive increasing number of models to estimate by simply using polynomial regression in order of two. There are already 63 models under such condition with only two variables, not to mention how many models need to be built with other polynomial regressions when the order is slightly larger than two.

1.2.3 Three Attitudes

Countless of researchers achieved remarkable results through decades. Many of the studies patiently sit down and talk to the crazily enlarged dimensionality and try to dig out its real thoughts. Though the inspiration of principle component analysis (PCA) can trace back all the way to Pearson's so called “closest fit” of data points[118], truly thanks to the amazing results from applied linear algebra like singular value decomposition (SVD) and eigenvalue decomposition (EVD) that this powerful tool and its derivative improvements are still popular today and habitually applied in various fields. It is completely possible that one finally lost his patience with this complicated discussion that forcefully runs down the topics to end the conversation. Proposed by Tikhonov[145] in 1943, regularization introduces a parameter-wised penalty to solve the ill-conditioned inverse problems especially in regression and classification. Theoretical results and applications that focused some of the implements such as ridge[72], LASSO[144] and elastic net[164] constantly draw attention every year. Instead of reasoning out the whole conversation, one can bring up some small issues at a time but having meetings much more frequently. Popularized by researchers like Brieman[22] and Ho[70] that ensemble learning is undoubtedly one of the strongest work in terms of performance nowadays. In later chapters we will address and discuss these significant works in detail.

1.3 Outline of the Thesis

The thesis is divided into five chapters to deliver a relative thorough study of our project. Despite the introductory chapter, the more detailed review starts from chapter 2 and chapter 3. Our two relative studies and results are demonstrated in chapter 4 and 5 then we summarize the topic in the end of chapter 5.

In chapter 2 and 3 some commonly used modern techniques that deals with high dimensional data are reviewed. We are focused on the mechanisms of some PCA based algorithms and its derivatives. Some of their similarities and relationships are explored with examples. Also, their limitations and improvements are addressed. Then a general introduction of ensemble learning can be reached in chapter 3. We will discuss three most popular ensemble learning algorithms in modern statistics.

Since the application may involve outlier detection, Chapter 4 contains a more detailed review of recent outlier detection algorithms with pros and cons. More importantly, current methods dealing with high dimensional data will be emphasized in this chapter.

Later in chapter 4 we implement our extensive studies on current methods. Also, we can talk about some of results in outlier detection and classification and compare to the performances to some of the current algorithms in chapter 5. Both simulations and an example of the benchmark dataset are involved.

In the end, we summarize the reason of pluses and minuses in terms methodology and performances of our proposed algorithms. Our future directions and works may be raised in the final paragraphs.

Chapter 2

Common Techniques in High Dimensions

2.1 Principle Component Analysis

2.1.1 Brief Review and Recent Developments

The content of Principle Component Analysis (PCA) can be written into several books. Although the most famous derivations from Pearson[118] was done in 1901, not too many works were published until Hotelling[73] after 32 years later. It is unbelievable that after a hundred years later that from 2001 to 2002, there are still over thousands of paper published that related to PCA within a single year. Another distinct point to mention is Eckart and Young's[38] illustration of the connection between principle components and singular value decomposition (SVD) derived by Beltrami[13] and Jordan[82]. It turned out that the ℓ-2 low rank approximation of the data can be obtained by the diagonal matrix with larger elements decomposed from SVD. The method still stands for the most powerful decomposition today. In fact, eigenvalue decomposition of a low-dimensional covariance matrix can be largely simplified by just computing the SVD of the original data matrix.

Later influential studies were mainly targeted on its two infamous limitations. First, principle components are assumed to be linearly separable. Commonly this convenience does not hold in many fields especially social science. Thus plenty of approaches had been proposed by researchers to solve this problem throughout decades and they are generously categorized as non-linear principle component analysis (NLPCA).

Majority of the attentions had been drawn by two studies during 90's. The two most famous methods either creating non-linear functions to map the original spaces to reduced spaces or reshape the data to a higher dimensional space to compromise the linearity. Kramer[90] (1991) simply trained a two sigmoid layer neural network (NN) that maps the input space to low-dimensional feature space and then de-maps the outputs back to data space. At last, the first mapping layer of the trained NN can be separated and used as NLPCA to reduce the dimensionality of the data. Schölkopf *et al.*[140] (1998) adopted the kernel function to project input space into high-dimensional feature space and then perform the regular PCA on that space. The curse of dimensionality vanished to the number of observations by the inner product of the kernel functions unless n is too large.

Even the data is linearly separable, the second issue lies in the nature of PCA that it is always searching the largest variances. If the data does not scale well or there are some contamination, say an outlier, that can drive the entries low-rank approximation far away. Research directions that involved in this type of problems are categorized as robust principle component analysis (Robust PCA or RPCA). Due to the application in image processing and computer vision, studies of how to inject the robustness into PCA based algorithms are still intensely explored. Candès *et al.*[24] (2009) proposed a penalizing term on the small perturbation matrix beside the low-rank approximation. The method is so called principle component pursuit (PCP) that in sense of solving convex optimization. Netrapalli *et al.*[109] (2014) presented an alternative non-convex approach that greatly challenged the convex low-rank approximation in terms of computational efficiency. Later in this chapter we will discuss a little bit more about some PCA or RPCA based algorithms and in chapter 5 and we compare our algorithm and a PCA based algorithm in terms of accuracy.

2.1.2 Some General Deductions

The classical PCA problem can be summarized as finding the best representation or basis for the data space. It is very intuitive to rebuild the coordinates or project the data to new arrangements according to its variance. Thus, assume our data $\boldsymbol{X}$ has already centered to 0, in convex fashion we can define the problem as:

$$\ell\left(\boldsymbol{W},\boldsymbol{Z}\right)=\arg\min_{\boldsymbol{W},\boldsymbol{Z}}\left\|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{Z}\right\|_F^2=\arg\min\sum_{i=1}^{n}\left\|\mathbf{x}_i-\hat{\mathbf{x}}_i\right\|^2 \tag{2.1}$$

This is so called the *reconstruction error* in PCA. Here the data $\boldsymbol{X}$ is presented as a $p \times n$ matrix. $\boldsymbol{W}$ is an orthonormal $p \times d$ matrix that representing directions having largest variance. $\boldsymbol{Z}$ is a $n \times d$ matrix, where $d < p$, that actually builds from d eigenvectors associated with ranked eigenvalues from largest to smallest. Thus each row of $\boldsymbol{Z}$: $\mathbf{z}_i = \boldsymbol{W}^\top\mathbf{x}_i$, where $i = 1, 2, \cdots, n$, denotes the encoding of original data into our new d-dimensional column space. Naturally, the reconstruction or decoding process is expressed by $\hat{\mathbf{x}}_i = \boldsymbol{W}\mathbf{z}_i$ and then we can minimize this error to obtain our estimation $\hat{\boldsymbol{W}}$. In addition, $\|\boldsymbol{X}\|_F$ denotes the Frobenius norm of $\boldsymbol{X}$:

$$\|\boldsymbol{X}\|_F=\sqrt{tr\left(\boldsymbol{X}^\top\boldsymbol{X}\right)} \tag{2.2}$$

If we step back to examine a larger picture of the variance of data, with the reconstruction in 2.1 above, we have:

$$\mathbb{E}\left[\|\boldsymbol{X}\|^2\right]=\mathbb{E}\left[\left\|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{X}\right\|^2\right]+\mathbb{E}\left[\left\|\boldsymbol{W}^\top\boldsymbol{X}\right\|^2\right] \tag{2.3}$$

where $\mathbb{E}\left[\|\boldsymbol{X}\|^2\right]$ is the total variance of original data. When we subtract the reconstruction error $\mathbb{E}\left[\left\|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{X}\right\|^2\right]$ we have $\mathbb{E}\left[\left\|\boldsymbol{W}^\top\boldsymbol{X}\right\|^2\right]$. This is usually referred as the actual amount of *variance captured* in PCA. It is very intuitive to think that minimizing the reconstruction error equals maximizing our variance captured. Now to show the connection between these two parts, for convenient purpose, we are only looking for 1-dimensional solution that $d = 1$ and assume that this principle component vector with unit length:

$$\begin{aligned}\ell\left(\boldsymbol{w}, \boldsymbol{Z}\right) &= \arg\max_{\boldsymbol{w}} \mathbb{E}\left[\left\|\boldsymbol{w}^{\top}\boldsymbol{X}\right\|^{2}\right], \quad s.t.\ \|\boldsymbol{w}\| = 1 \\ &= \arg\max_{\boldsymbol{w}} \frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{w}^{\top}\mathbf{x}_{i}\right\|^{2} \\ &= \arg\max_{\boldsymbol{w}} \boldsymbol{w}^{\top}\boldsymbol{X}\boldsymbol{X}^{\top}\boldsymbol{w} \\ &= \max\lambda\left(\boldsymbol{X}\right)\end{aligned} \tag{2.4}$$

where $\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^{\top} = \boldsymbol{C}$ is just straightly equal to the empirical covariance matrix of X. Since we set $d = 1$ and $\boldsymbol{w}$ is orthonormal, $\boldsymbol{w}^{\top}\boldsymbol{w}$ vanished to 1. It turned out the solution is just the maximum eigenvalue of our covariance matrix. Similarly, remain the same setting above, we minimize the reconstruction error:

$$\begin{aligned}\ell\left(\boldsymbol{w}\right) &= \arg\min_{\boldsymbol{w}} \sum_{i=1}^{n}\left\|\mathbf{x}_{i} - \boldsymbol{w}\boldsymbol{w}^{\top}\mathbf{x}_{i}\right\|^{2} \\ &= \arg\min_{\boldsymbol{w}} \sum_{i=1}^{n}\left(\|\mathbf{x}_{i}\|^{2} - \left(\boldsymbol{w}^{\top}\mathbf{x}_{i}\right)^{2}\right) \\ &= \arg\min_{\boldsymbol{w}}\left(tr\left(\boldsymbol{X}\boldsymbol{X}^{\top}\right) - \left\|\boldsymbol{w}^{\top}\boldsymbol{X}\right\|^{2}\right)\end{aligned} \tag{2.5}$$

where $tr\left(\boldsymbol{X}\boldsymbol{X}^{\top}\right)$ is just a constant. Thus, minimizing the reconstruction error is just the opposite of equation of maximizing the variance captured. Similar results can be showed for principle components $d > 1$, a demonstration of first two principle components of Iris dataset can be found in Figure (2.1).

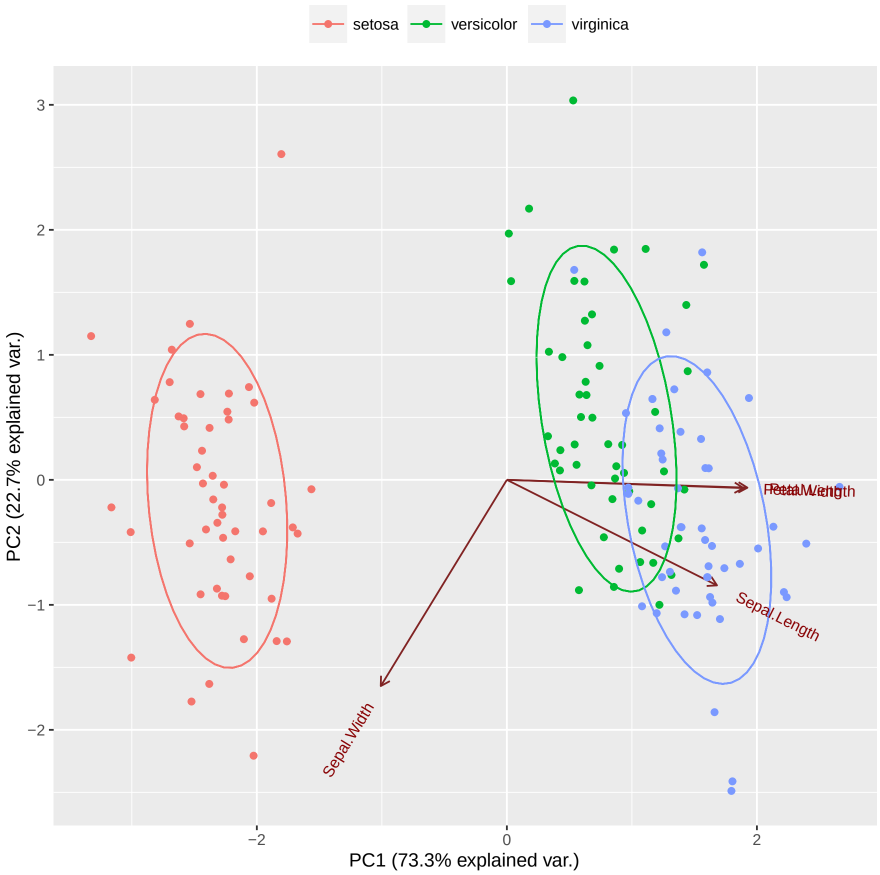

Figure 2.1: projection of Iris dataset on first two principle components

Another important point of view is the relationship between SVD and PCA. As we mentioned before, the principle components are eigenvectors of the covariance matrix of $\boldsymbol{X}$, here on the purpose of clarity, we still denote the dimensionality of $\boldsymbol{X}$ as $p \times n$. Thus if we apply the eigenvalue decomposition on our covariance matrix $\boldsymbol{\mathcal{C}}$ we have:

$$\boldsymbol{\mathcal{C}} = \boldsymbol{X}\boldsymbol{X}^{\top} = \boldsymbol{U}\Lambda\boldsymbol{U}^{\top} \tag{2.6}$$

Now $\boldsymbol{U}$ is the matrix contains eigenvectors of $\boldsymbol{C}$ in each column. Thus the principle components can just be represented by $\boldsymbol{U}^\top \boldsymbol{X}$ similar to previous equations. Λ is a diagonal matrix filled with ranked eigenvalues $\lambda_i, i = 1, 2, \cdots, d$ of $\boldsymbol{A}$. Since in SVD we have:

$$\boldsymbol{X} = \boldsymbol{U}\Sigma\boldsymbol{V}^\top \tag{2.7}$$

where $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}$ and $\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}$, Σ is also a diagonal matrix with all singular values then we can present our covariance matrix as:

$$\begin{aligned} \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\top &= \frac{1}{n}\boldsymbol{U}\Sigma\boldsymbol{V}^\top\boldsymbol{V}\Sigma\boldsymbol{U}^\top \qquad (2.8) \\ &= \boldsymbol{U}\left(\frac{1}{n}\Sigma^2\right)\boldsymbol{U}^\top \end{aligned}$$

Thus, if we denote the entries of Σ as $s_i, i = 1, 2, \cdots, d$, then the eigenvalues $\lambda_i = (1/n)\, s_i^2$ are just the scaled square of the singular values of the covariance matrix. So, the principle components are the columns of the left singular matrix $\boldsymbol{U}$. Computationally, using the SVD to perform PCA is generally preferred. d of singular vectors will only require $\mathcal{O}(npd)$ which is much more cheap than computing the covariance matrix with expense of $\mathcal{O}\left(np^2\right)$. Figure (2.2) illustrates an example of dimensionality reduction of an image.

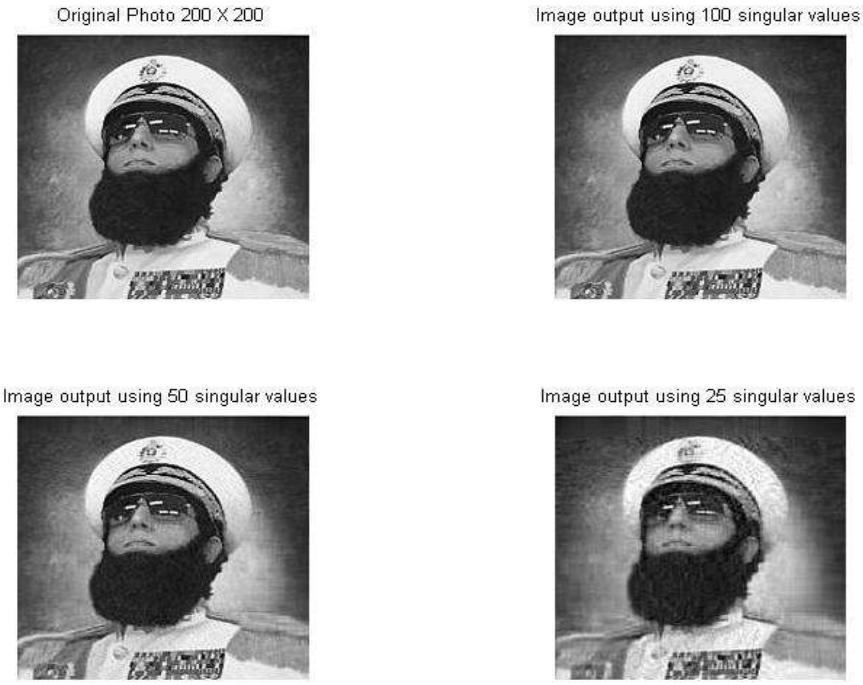
Original Photo 200 X 200
Image output using 100 singular values
Image output using 50 singular values
Image output using 25 singular values

Figure 2.2: SVD of an image with different choices of largest singular values

2.1.3 Notes on NLPCA

As many of the problems in machine learning, PCA can also be solved as convex optimization with constraint just like we mentioned earlier. The process of projecting $\boldsymbol{X}$ back and forth: $\boldsymbol{W}\boldsymbol{W}^{\top}\boldsymbol{X}$ is considered as a common analogy of encoding and decoding. Oja's[113] work established the connection between PCA and neural networks in 1982 that a modified Hebbian's learning was adopted fit the PCA into linear neurons. Later several studies including Kramer's[90] autoassociative principle component network are all in this encoding-decoding trend:

$$\ell\left(\boldsymbol{F},\boldsymbol{G}\right) = \arg\min_{\boldsymbol{F},\mathcal{G}_{\boldsymbol{F}}} \sum_{i=1}^{n} \left\|\mathbf{x}_i - \boldsymbol{F}\left(\boldsymbol{G}_{\boldsymbol{F}}\left(\mathbf{x}_i\right)\right)\right\|^2 \tag{2.9}$$

where $\boldsymbol{F}$ and $\boldsymbol{G}_{\boldsymbol{F}}$ are non-linear functions. The function $\boldsymbol{G}_{\boldsymbol{F}} : \mathbb{R}^p \rightarrow \mathbb{R}^1$ while

$\boldsymbol{F} : \mathbb{R}^1 \rightarrow \mathbb{R}^p$, a more specific presentation is shown below in figure (2.3). One crucial issue with Kramer's method that raised by Malthouse[102] in 1998 is that the pre-defined continuity of the function $\boldsymbol{G_F}$. Since it is possible that some principle components' projection index are discontinuous, the ambiguity can mislead the index to map points to undesirable places.

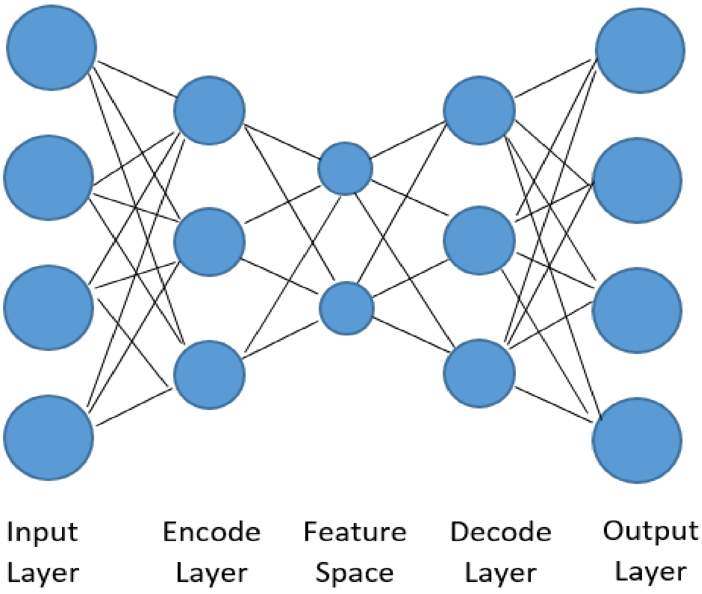

Figure 2.3: The 5-layer neural network map the input to $\mathbb{R}^d$ and de-map to $\mathbb{R}^p$

Instead of mapping data to a non-linear lower-dimensional space, application of kernel trick that allow us to project the points to non-linear higher dimensional feature space. But we jump into kernel, another key factor to mention is to look back at PCA in sense of using Lagrange multiplier λ to solve the optimization with a slightly different constraint. Assume we still searching for the 1-dimensional principle component:

$$\ell\left(\boldsymbol{w}, \lambda\right) = \arg\max_{\boldsymbol{w},\lambda} \left\|\boldsymbol{w}^\top \boldsymbol{X}\right\|^2 + \lambda\left(\|\boldsymbol{w}\| - 1\right), \quad s.t.\|\boldsymbol{w}\|^2 \leqslant 1 \tag{2.10}$$

Take the derivative with respect to $\|\boldsymbol{w}\|$ we have:

$$\nabla \ell \left(\boldsymbol{w}, \lambda\right) = 2\left(\boldsymbol{X}\boldsymbol{X}^\top\right)\boldsymbol{w} - 2\lambda \boldsymbol{w}$$

Let the gradient equals to 0, then the problem reduced to an eigenvalue equation:

$$\left(\boldsymbol{X}\boldsymbol{X}^\top\right)\boldsymbol{w} = \lambda \boldsymbol{w} \tag{2.11}$$

This basic form is greatly linked kernel PCA by Schölkopf[137] and his generalized version of *Representor Theorem*. Just like the normal vectors perpendicular to the decision plane in Support Vector Machine, we can imagine if our principle components like $\boldsymbol{w}$ can be decomposed as:

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i = \boldsymbol{X}\boldsymbol{\alpha} \tag{2.12}$$

The essential of PCA is no more than inner product, if we replace the vector $\boldsymbol{w}$ during maximizing our variance captured: $\boldsymbol{w}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{w}$. Then the form turns out to be $\boldsymbol{\alpha}^\top\left(\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{\alpha} = \boldsymbol{\alpha}^\top \boldsymbol{D}^2\boldsymbol{\alpha}$ with constraint $\boldsymbol{\alpha}^\top \boldsymbol{D}\boldsymbol{\alpha} = 1$ resembling $\|\boldsymbol{w}\| = 1$. Thus, we just loop back to solve the same sort of eigenvalue problem like $\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\alpha} = \tilde{\lambda}\boldsymbol{\alpha}$. Vital point here is computing the inner product does not require the actual access of both vectors. This property functioning like a “black box” no matter how we move our data points as long as they still remain in the form of the inner product. But first, we define the kernel function as:

$$\boldsymbol{\Phi} : \mathbb{R}^p \rightarrow \boldsymbol{\mathcal{F}}, \quad \textit{where } \boldsymbol{\mathcal{F}} \textit{ is a Hilbert space} \tag{2.13}$$

$\mathcal{F}$ is our feature space that can be arbitrarily large without a bound. Some common examples include: Gaussian RBF kernel $\boldsymbol{\phi}(\mathbf{x},\mathbf{y}) = \boldsymbol{e}^{-\|\mathbf{x}-\mathbf{y}\|^2}$, polynomial kernel $\boldsymbol{\phi}(\mathbf{x},\mathbf{y}) = \left(1+\mathbf{x}^\top\mathbf{y}\right)^2$ *etc.*. Thanks to Mercer's theorem, for a finite set $\{x_i\}, i = 1, 2, \cdots, n$ in $\boldsymbol{X} \in \mathbb{R}^n$ and countable set of non-negative eigenvalues $\{\lambda_i\}, i = 1, 2, \cdots, \infty$, the continuous kernel function of pair $\boldsymbol{\mathcal{K}}(x, z)$ on $\boldsymbol{X} \times \boldsymbol{X}$ can be decomposed to $\sum_{t=1}^{\infty} \lambda_t \phi_t (x) \phi_t (z)$. By substituting just in the fashion of 2.12, we can repost our object as:

$$\frac{1}{n}\boldsymbol{K}\boldsymbol{\alpha} = \tilde{\lambda}\boldsymbol{\alpha} \tag{2.14}$$

where $\boldsymbol{K}$ is our $n \times n$ kernel matrix that $\boldsymbol{K}_{ij}$ is defined by $\boldsymbol{\phi}\left(\mathbf{x}_i\right)^\top \boldsymbol{\phi}\left(\mathbf{x}_j\right)$. Thus, this just reduced to a common situation just like any other eigenvalue problems. Figure (2.4) shows projection of the spam and iris dataset on two kernel principal components by using RBF kernel.

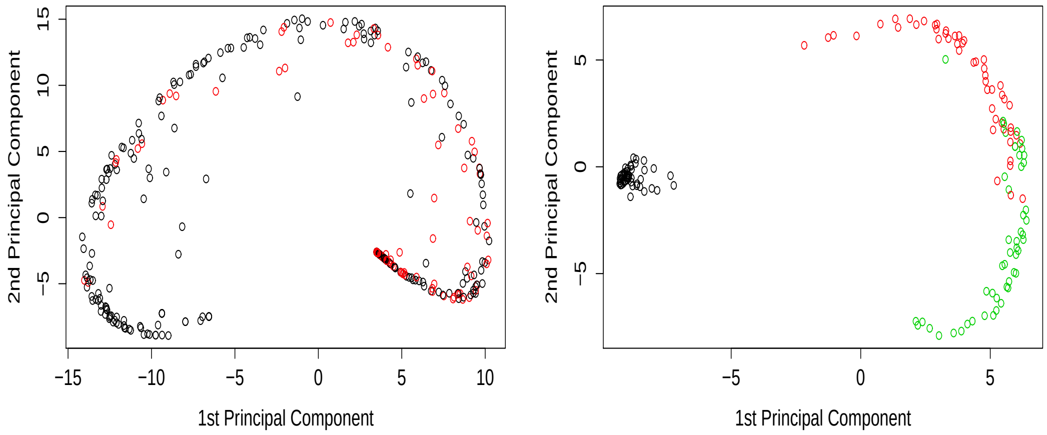

Figure 2.4: (left) projection of the spam dataset on 2 principal components by RBF kernel. (right) projection of the iris dataset on 2 principal components by RBF kernel.

2.1.4 Notes on RPCA

Beginning from the basic assumption of the intrinsic lower dimensionality given a large data matrix, Candès[24] repost the PCA problem as decomposition the data matrix itself:

$$\boldsymbol{X} = \boldsymbol{L}_0 + \boldsymbol{S}_0 \tag{2.15}$$

So the PCA problem can be described as optimizing $\|\boldsymbol{X} - \boldsymbol{L}\|$ subject to $rank\,(L) \leqslant d$, where L is the approximation of low rank matrix $\boldsymbol{L}_0$ and the support of $\boldsymbol{S}_0$ is assumed to be sparse. If $\boldsymbol{X}$ is heavily contaminated, the noise reside in $\boldsymbol{S}_0$ can be largely amplified. Thus, the authors proposed Principle Component Pursuit (PCP) by separating the low rank approximation and the sparse component:

$$(\boldsymbol{L}, \boldsymbol{S}) = \arg\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1\,, \quad s.t.\ \boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S} \tag{2.16}$$

where the $\|\boldsymbol{A}\|_*$ is an nuclear norm of matrix $\boldsymbol{A}$. The meaningful disentangling and recovery of $\boldsymbol{L}_0$ requires $\left\|\boldsymbol{U}\boldsymbol{V}^\top\right\|_\infty < \mu\sqrt{r}/n$, where r is the rank of $\boldsymbol{L}$ and μ is its level of incoherence. The paper illustrated one way to solve this convex optimization by applying the augmented Lagrange multiplier:

$$\ell\,(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{Z}) = \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \boldsymbol{Z}^\top (\boldsymbol{X} - \boldsymbol{L} - \boldsymbol{S}) + \frac{\mu}{2} \|\boldsymbol{X} - \boldsymbol{L} - \boldsymbol{S}\|_F^2$$

Thus, the problem can be solved by sequentially updating $\boldsymbol{L}$, $\boldsymbol{S}$ and $\boldsymbol{Z}$ until the approximations: $\|\boldsymbol{X} - \boldsymbol{L} - \boldsymbol{S}\|_F \leqslant \xi \|\boldsymbol{X}\|_F$ where ξ is enough small. For example, we start with selecting $\mu > 0$ and setting $\boldsymbol{S}_0, \boldsymbol{Z}_0$ equals to 0 then updating $\boldsymbol{L}_1$ with the solution:

$$\boldsymbol{L}_{t+1} = \boldsymbol{\mathcal{D}}_{\mu^{-1}}\left(\boldsymbol{X} - \boldsymbol{S}_t + \mu^{-1}\boldsymbol{Z}_t\right)$$

where $\boldsymbol{\mathcal{D}}_{\mu^{-1}}\left(\boldsymbol{X}\right)$ is a function recovers $\boldsymbol{X}$ from its SVD having only singular values that larger than μ^{-1}. It is defined by:

$$\boldsymbol{\mathcal{D}}_{\tau}\left(\boldsymbol{X}\right) = \boldsymbol{U}\boldsymbol{\mathcal{S}}_{\tau}\left(\Sigma\right)\boldsymbol{V}^{\top}, \quad where\ \boldsymbol{\mathcal{S}}_{\tau}\left(x\right) = I\left(\max\left(|x| - \tau\right)\right)$$

$\boldsymbol{\mathcal{S}}_{\tau}$ is so called the shrinkage operator. Then $\boldsymbol{S}$ can be updated by:

$$\boldsymbol{\mathcal{S}}_{\lambda\mu^{-1}}\left(\boldsymbol{X} - \boldsymbol{L}_{t+1} + \mu^{-1}\boldsymbol{Z}_t\right)$$

Then we can update $\boldsymbol{Z}$ by a further step to complete the loop. The algorithm needs to find the eigenvalues for each step so sometimes it could be computationally expensive and the choices of μ and ξ are vital. Later in outlier detection we will talk a little bit more about the applications of RPCA but here we discuss no more details about it. Other related contents including its non-convex development can be found in [109] and [39].

Chapter 3

Basics of Ensemble Learning

3.1 An Overview

The mechanism of ensemble learning functions like ants which is to gather multiple tiny workers to move a huge target. Kearns[84] posted a progress of machine learning class project in 1988 that asking whether the potential of a set of weak learners can be combined to improve the accuracy. This so called *Hypothesis Boosting Problem* was definitively answered by Schapire[134] in 1990. He introduced the *Boosting* which is one of the most widely used and powerful algorithms in ensemble learning. The method was originally built for classification and later adapt itself to regression problems. Just like the question posted by Kearns, the algorithm select a weak learner that is slightly better than random guessing in the training process. For each time and each of these weak learners are trained with the dataset, the ones that are more accurate are rewarded with a candy. In the end, all of the learners are weighted by their success and failure to create a voted machine for classification or a averaged model for regression. Later in the vital paper in 1995 that Schapire and Freund[47] were introducing the most popular boosting algorithm *Adaboost*, a similar story was told in analogy of horse-racing gamblers while they were talking about their improved version of adaboost[49]. A pool of personal experience based suggestions, if possibly, slightly better than random guessing, can build a reliable prediction. At first glance, the phenomena itself is interesting and weird, because the gap between the mathematical principles and the practical results seems to be huge. However, it is just like Schapire's later explanation in [136] that referring to Vapniks[155] great work, "by understanding the nature of learning at its foundation" in terms of both algorithms and this phenomena.

In the same year, Breiman[20] introduced bootstrap aggregation, so called *Bagging*. In fact, the simplicity of bagging is unspeakably shocking but it turned out to be convincing after Breiman demonstrate the improvements of the prediction error UCI repository datasets. The method regenerate training samples by bootstrapping the original observations and later largely applied in decision tree models. A comparison between bagging and two boosting algorithms was raised by Opitz and Maclin[101] in 1999 that bagging was shown contently outperformed its base learner but occasionally much less accurate than boosting while boosting may fluctuate down below its base learner. In Breiman's another paper[21] in 1996, after he talked about Geman's bias-variance decomposition of the error term, he assumed that both bagging and boosting are reducing the variance in order to achieve higher accuracy. Later of that year Schapire and Freund[49] indicated that boosting also reduces the bias by forcing the weak learners to focus on different parts of the instance space. Bauer and Kohavi[11] performed a more thorough comparison in 1999 among several algorithms including bagging and adaboost and unexpectedly concluded that not only boosting but also bagging may reduce the bias part of error in certain real-world datasets. Though many of the answers looped back to "no free launch", bagging was mentioned as appropriate for decision trees and neural networks by Opitz and Maclin in [101]. It is very interesting that the decision tree algorithm may just like a "twitchy sow's ear" in Breiman's[22] analogy that can build up one of his most famous "silk purse" *Random Forest* (RF).

Ho[69] discussed a systematic way of growing trees in her 1995 paper while many of other studies focused on sophisticated pruning procedures. In deliberation of the oblique hyperplanes, she took the advantage of randomization on subspaces of variable dimensions. Later in a more comprehensive study[70] conducted in 1998, her approach was formally named *Random Subspace Method* (RSM). Variables are randomly selected and eight different splitting function were used to construct forests. Like many other ensemble learning, voting through a weighted process or other techniques can be applied in final model aggregation. Strong performances were shown in this paper against boosting and bagging in certain datasets but there was one question left open. the performance of the algorithm seemed largely influenced by the number of dimensions when tested by the data called “dna”. Ho suggested select the roughly about half of the variables and there are rarely a studies have been systematically solved this issue that most of the selections are based on empirical evidence or cross validation. Faced the variance in performances among different classifiers, Ho published a report[71] in 2001 to discuss some empirical observations that lead to the measures of data or problem complexity. Followed by the research of Kuncheva *et al.*[95], the complexity of generated random subspaces are still considered lack of understanding thus further studies are required to illuminate the foundations behind it.

3.2 Boosting

3.2.1 Foundations of Adaboost

Schapire paid his tribute to Vapnik and Chervonenks[156] for uncovering the fundamental mechanism of learning theory in [136]. He summarized that a classifier learnt from data can be considered as effective by three conditions:

- The training process needs support from large amount of observations.
- Model has reasonable fit without having too much training error.
- Parsimony applies, the simpler the better.

Though it is very intuitive to think about The VapnikChervonenkis dimension (VC dim) as the direct explanation of the function of boosting by Schapire. He proved the error is bounded by:

$$\boldsymbol{\mathcal{E}} \leqslant 2^{\boldsymbol{T}} \prod_{t=1}^{\boldsymbol{T}} \sqrt{\boldsymbol{\mathcal{E}}_t \left(1 - \boldsymbol{\mathcal{E}}_t\right)} \tag{3.1}$$

where t is the number of iteration and for each $\boldsymbol{\mathcal{E}}_t$ there exist some $\alpha > 0$ for $\boldsymbol{\mathcal{E}}_t = 1/2 - \alpha$ so each error has the value below $1/2$. This is so called the *weak learning condition*. Given this assumption, it is noticeable that the training error of adaboost algorithm can reduce to 0 at speed of $\boldsymbol{\mathcal{O}}\left(\log n\right)$, where n is the number of observations. So the error of the aggregated classifier from training and testing is just a function of the number of iteration T. The error of classifier applied on sample data is guaranteed to be small. Then Schapire applied Vapnik's[154] Structural Risk Minimization (SRM) to restrict the number of weak learners and their simplicity in order to make the error of the whole domain of $\boldsymbol{X}$ close to the empirical error on the training observations. For a binary classifier class $\boldsymbol{\mathcal{C}}$, if all classifiers have VC dimension $d \geqslant 2$, it can be proved that the upper bound of the class domain $\boldsymbol{\Theta}$ of T linear binary classifiers in $\boldsymbol{\mathcal{C}}$ is:

$$VC\left(\boldsymbol{\Theta}_T\left(\boldsymbol{\mathcal{C}}\right)\right) \leqslant 2\left(d+1\right)\left(T+1\right)\left(\log_2(e\left(T+1\right)\right) \tag{3.2}$$

where e denotes the natural number. There is a closely linear relationship between the VC dimension of the aggregated classifier and the number of iterations. This shows that the classical overfitting phenomena does happen with too many iterations but may be avoided in practical situations.

3.2.2 Adaboost: The Algorithm

Formally, we have the algorithm of adaptive boosting *Adaboost*:

Algorithm 1 Adaptive Boosting

```
procedure ADABOOST($T$)
    Set $\alpha_i^{(1)} \in \boldsymbol{w}^{(1)} = \frac{1}{n}$ for all $i$, where $i = 1, \cdots, n$
    for $t = 1$ to $T$ do
        Set $\boldsymbol{w}^{(t)} = \frac{\boldsymbol{w}^{(t)}}{\sum_{i=1}^{n} w_i^{(t)}}$
        Call the weak learner $f$ and Assign weights $\boldsymbol{w}^{(t)}$
        Compute the error $\widehat{\boldsymbol{\mathcal{E}}}^{(t)} = \sum_{i=1}^{n} \alpha_i^{(t)} |\mathbf{y}_i^{(t)} - \widehat{f}_t(\mathbf{x}_i^{(t)})|$
        Set $\beta^{(t)} = \frac{\widehat{\boldsymbol{\mathcal{E}}}^{(t)}}{1 - \widehat{\boldsymbol{\mathcal{E}}}^{(t)}}$
        Set weights $w_i^{(t+1)} = w_i^{(t)} \left(\beta^{(t)}\right)^{1 - |\mathbf{y}_i^{(t)} - \widehat{f}_t(\mathbf{x}_i^{(t)})|}$
    end for
    Output can be computed by:
        $\widehat{f}_t(\mathbf{x}^{(t)}) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \log\left(\frac{1}{\beta^{(t)}}\right) \widehat{f}_t(\mathbf{x}^{(t)}) \geqslant \frac{1}{2} \log\left(\frac{1}{\beta^{(t)}}\right) \\ 0 & \text{otherwise.} \end{cases}$
end procedure
```

Starting from uniform weights that $\boldsymbol{w}_i^{(1)} = 1/n, foralli$, the weight vector $\boldsymbol{w}^{(t)}$ is updated in each loop by $\beta^{(t)}$ then final output is generated by weighted voting. $\beta^{(t)}$ is actually a function of $\boldsymbol{\mathcal{E}}^{(t)}$ that manipulating the weight vector. Notice for $\beta^{(t)}$ there is a reversed adjustment that the “lost” function is $1 - |\mathbf{y}_i^{(t)} - \widehat{f}_t(\mathbf{x}_i^{(t)})|$. Thus, if the classifier is making a correct decision, the probability assign it will reduce. Vice versa, If the classifier is making an incorrect decision, the probability assign it will increase. Adaptive boosting is not very similar to many other boosting algorithms by the time which the errors of the weak learners are used to adjust the structure. Other forms of adaboost can be found in studies like[135], slightly modified adaboost in applications like SVM are in [99].

3.3 Bagging and Random Subspace

3.3.1 Reasons for Stabilization

In Breiman's original paper, he explained why bagging works well after some real world data experiments. The key of bagging is to stabilize the variance, the square loss was adopted to demonstrate the his point. Suppose we have data examples are independently sampled from joint distribution $\boldsymbol{P}$: $\boldsymbol{\mathcal{D}} = \left\{(y_N, \mathbf{x}_N)\right\}$ where $N = 1, 2, \cdots, n$, y_N are all continuous and $\widehat{f}(\mathbf{x}, \boldsymbol{\mathcal{D}})$ represents our prediction thus the aggregated version $f_A(\mathbf{x}, \boldsymbol{\mathcal{D}})$ is just:

$$f_A(\mathbf{x}, \boldsymbol{P}) = \mathbb{E}\left(\widehat{f}(\mathbf{x}, \boldsymbol{\mathcal{D}})\right) \tag{3.3}$$

The average error $\boldsymbol{\mathcal{E}}$ from prediction over distribution $\boldsymbol{P}$ can be expressed by the expectation of square loss:

$$\boldsymbol{\mathcal{E}} = \mathbb{E}_{\boldsymbol{\mathcal{D}}}\mathbb{E}_{Y,\boldsymbol{X}}\left(Y - f(\boldsymbol{X}, \boldsymbol{\mathcal{D}})\right)^2 \tag{3.4}$$

And if we denote the error after aggregation as $bcfE_A$:

$$\boldsymbol{\mathcal{E}}_A = \mathbb{E}_{Y,\boldsymbol{X}}\left(Y - f_A(\mathbf{x}, \boldsymbol{P})\right)^2 \tag{3.5}$$

Then by Jensen's inequality $\phi\left(\mathbb{E}(Z)\right) \leqslant \mathbb{E}\left(\phi(Z)\right)$. So if $\phi(Z) = Z^2$ we have:

$$\begin{aligned} \boldsymbol{\mathcal{E}} &= \mathbb{E}\left(Y^2\right) - 2\mathbb{E}\left(Yf_A\right) + \mathbb{E}_{Y,\boldsymbol{X}}\mathbb{E}_{\boldsymbol{\mathcal{D}}}\left(\widehat{f}(\boldsymbol{X}, \boldsymbol{\mathcal{D}})\right)^2 \\ &\geqslant \mathbb{E}_{Y,\boldsymbol{X}}\left(Y - f_A(\boldsymbol{X}, \boldsymbol{P})\right)^2 = \boldsymbol{\mathcal{E}}_A \end{aligned} \tag{3.6}$$

So the aggregated predictor has smaller mean squared error. Every time, The more diverse $\boldsymbol{\mathcal{D}}$ are sampled from $\boldsymbol{P}$ the more difference of two sides of the Jensen's inequality $\left(\mathbb{E}_{\boldsymbol{\mathcal{D}}}\left(\widehat{f}(\mathbf{x},\boldsymbol{\mathcal{D}})\right)\right)^2 \leqslant \mathbb{E}_{\boldsymbol{\mathcal{D}}}\left(\widehat{f}(\mathbf{x},\boldsymbol{\mathcal{D}})\right)^2$ can be. Thus, if the base learner is not stable, it can actually travel around inside of $\boldsymbol{P}$ by the bootstrap approximation. However, if the base learner is stable, bagging may not help too much in terms of accuracy. Breiman also showed in classification scenario, bagging is always improving the performance even the classifier is nearly optimal. More details can be found in [20], here we do not discuss further details.

3.3.2 Bagging: The Algorithm

The algorithm of bagging is shockingly refined in terms of concise. Bootstrap samples are generated by uniformly sampling n observations from the original training data with replacement. There are B bootstrap samples and with each one a classifier or predictor $\widehat{f}^{(b)}(\boldsymbol{X},\boldsymbol{\mathcal{D}})$ is computed by using the bth sample. However this procedure is directly related to its robustness which we will talk about it later in chapter 4. Here, formally we have the bootstrap aggregation for classification *Bagging*:

Algorithm 2 Bootstrap Aggregation

```
procedure BAGGING($B$)
    for $b = 1$ to $B$ do
        Draw with replacement $\{i_1^{(b)}, \cdots, i_n^{(b)}\}$ from $\{1, 2, \cdots, n\}$ to form the bootstrap sample $\mathscr{D}^{(b)}$
        Call the $b$th hypothesis $\widehat{f}^{(b)}$ with $\mathscr{D}^{(b)}$
    end for
    Output can be computed by:
        $$\widehat{f}(\mathbf{x}) = \arg\max_{y \in \boldsymbol{Y}} \sum_{b:\widehat{f}^{(b)}=y} 1$$
end procedure
```

In the last step, the majority votes of the labels from all of the hypothesis become the final classifier. In continuous response predictions like regression, bagging will take the model averaging to build the final prediction.

3.3.3 Random Subspace Method

When Breiman[22] published his random forests in 2001, he mentioned two studies that greatly influenced by two studies. One is Amit and Geman's[6] geometrical investigation of the best split of trees in large dimensionality and another important research is Ho's[70] random subspace method (RSM). The method was originally used to build decision trees but it can actually adapt many other algorithms. Skurichina and Duin[141] applied RSM to linear classifiers like Linear Discriminant Analysis (LDA) for two-class problems. Like bagging, RSM also improves the prediction error by stabilizing classifiers especially when many of the linear classifiers are fickle. One of the benefits of RSM for building and aggregating the classifiers is the number of dimensionality may be much smaller than the original data. In sub-feature spaces the sample size does not change. So this method actually increases the relative observations that are available for each loop. When the data was combined with plenty of noise variables classifiers may be able to perform better in random subspaces than the original space. Thus the aggregated decision can outperform a single predictor or classifier. Similar proof of stabilization can be applied just like Breiman's bagging since bootstrap is also used for first step in RSM. Tao and Tang *et al.* combined symmetric bagging and RSM to stabilize relevance support vector machines based feedback schemes. Although RSM structure adapted many of the applications such as face recognition[162] and fMRI classification[94], there is rarely a deeper understanding or any analysis about the complexity of subspaces to solidify its foundation. In Kuncheva *et al.*'s experiments, it is imaginable that the complexity among subspaces are much higher than the complexity among bootstrap subsets. Furthermore, the number of the subspaces selected is directly related to the complexity but the complexity may reduce as more variables are selected. This may just due to the increase probability of selecting the overlapped dimensions. Most importantly, the noisy variables often observed to generate similar complexity comparing to the variables without redundancy. However, the researchers clarified that there is no clear methods of the measure of complexity and even the definition of complexity.

Like the notations we used in bagging, there are B bootstrap samples and with each sample there is our bth hypothesis $\widehat{f}^{(b)}(\boldsymbol{X}, \boldsymbol{\mathcal{D}})$. But within each bootstrap . Here, formally we have the random subspace method for classification *RSM*:

Algorithm 3 Random Subspace Method

```
procedure RANDOM SUBSPACE METHOD($B$)
    for $b = 1$ to $B$ do
        Draw with replacement $\{i_1^{(b)}, \cdots, i_n^{(b)}\}$ from $\{1, 2, \cdots, n\}$ to form the bootstrap sample $\mathscr{D}^{(b)}$
        Draw without replacement from $\{1, 2, \cdots, p\}$ a subset $\{j_1^{(b)}, \cdots, j_d^{(b)}\}$ to form $d$ variables
        Build the $b$th classifier $\widehat{f}^{(b)}$ with $\mathscr{D}^{(b)}$
        Drop unselected variables from $\mathscr{D}^{(b)}$ so that $\mathscr{D}_{sub}^{(b)}$ is $d$ dimensional
        Call the $b$th hypothesis $\widehat{f}^{(b)}(\mathscr{D}_{sub}^{(b)})$
    end for
    Output can be computed by:
        $\widehat{f}(\mathbf{x}) = \arg\max_{y \in \boldsymbol{Y}} \sum_{b : \widehat{f}^{(b)}(\mathscr{D}_{sub}^{(b)}) = y} 1$
end procedure
```

In the last step, the majority votes of the labels from all of the hypothesis become the final classifier. In continuous response predictions like regression, bagging will take the model averaging to build the final prediction.

Chapter 4

Random Subspace MCD

4.1 Outlier Detection

4.1.1 Previous Studies

The definition of outlier was never clear, descriptions from “observation point that is distant from other observations”[59] to “an observation that lies outside the overall pattern of a distribution”[68] can be found in plenty of books. Thus, great number of cases were considered as outliers. For example, the data contains missing values or extreme values in some observations, some of the variables do not come from the same distribution as our objective samples or even the part of the data is unspecified with huge errors. So the question of outlier detection is fully opened as almost no or vague paths to reach an undefined goal. In early multivariate studies, two approaches dealing with outliers seemed to draw majority of the the attentions with different pursuits. The two ways of solving outlier problems are very much like to complement each other. The difference lies in their primary target, one is to build the parametrical estimators for the data and another one is solely hunting for the outliers not matter whether estimators are required. However, all of the studies have one common latent need is discover the intrinsic structure of the data.

Rousseeuw *et al.*[129] [131] proposed *Minimum Volume Ellipsoid* (MVE) as a robust location estimator. Later based on MVE he developed *Minimum Covariance Determinant*[132] in application of outlier detection. Davies[23] proved that the MVE satisfy a local Hölder condition of order $1/2$ and also converges weakly to a non-Gaussian distribution at rate of $n^{-\frac{1}{3}}$. He and Wang[66] establish strong consistency and functional continuity that for MVE estimator can act reasonable if the shape of intrinsic distribution is likely to be elliptically symmetric. This type of estimator is criticized as slow convergence rate due to its large variability and low efficiency. Woodruff and Rocke[128] proposed *MULTOUT* in 1996 that combined several steps in MCD to create a hybrid approach to improve both computational expenses and peformances. Thus, a careful choice of parameters is commonly required. Billor *et al*[17] introduced BACON to find the best subset of the data at the initial process and Pena and Prieto[117]'s Kurtosis 1 chooses directions that maximize and minimize the univariate projected data. Maronna and Zamar[104] proposed their *Orthogonalized Gnanadesikan-Kettenring* (OGK) robust estimator in 2002 that claimed to be better and faster than MCD that deals with relative large dimensional situations.

4.1.2 Dance with Increasing Dimensionality

None of the algorithms we mentioned above can actually cope with high dimensional data. Aggarwal and Yu[2] proposed an algorithm that tries to find m of potential combinations of k subspaces in which the data is sparse. Though comparing to search each subspace that the method largely reduced the number of combinations, just like we mentioned in chapter 1, the number of combinations can rapidly shoot to sky with increasing dimensionality. Zhang et al.[160] in 2004 challenged with the same UCI machine learning repository and explained their *HOS-miner*. The algorithm tries to identify the subspaces that a given point is an outlier. Nguyen *et al.*[112] in 2011 criticized a monotonic behavior in Zhang's research that the condition does not have to be hold the outlier-residing subspaces. Nguyen proposed *High-DOD* that uses modified k-nearest neighbor weight outlier score and applied on normalized ℓ_p norm. Later Kriegel *et al.*[91] criticized *High-DOD* by its process of examine too many subspaces which bias can be generated. By the time, many of the outlier detection algorithms that deal with larger dimensionality are proposed such as *HiCS* by Keller *et al.*[85], *OutRank* by Müller *et al.*[108] and *COP* by Kriegel *et al.*[91] but none of these can actually handle or perform very well in true high dimensional data, especially for "large p, small n" problems. The one that catches our eyes is the PCA-based algorithm proposed by Filzmoser *et al.*[46] that named *PCOut* in 2008. It specifically targeted high dimensional outlier detection by taking advantages of the nature of PCA. The algorithm uses median absolute deviation normalized data to find out the most variable dimensions and use re-defined distances to classify the outliers. A simulation with $p = 2000$ was presented in the paper and a practical example of detecting outliers on a transposed micro-array gene expression dataset. In next section we talk more about the algorithm and compare it with our method in terms of accuracy in simulation study.

4.1.3 Alternatives to Parametric Outlier Detection Methods

The assumption of multivariate Gaussianity of the $\mathbf{x}_i$'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 [139] 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 $\Phi : \mathscr{X} \longrightarrow \mathscr{F}$. The one-class SVM novelty detection solves

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

Subject to

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

Using $\mathscr{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle = \Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j)$, we get

$$\widehat{f}(\mathbf{x}_i) = \texttt{sign}\left(\sum_{j=1}^{n} \widehat{\alpha}_j \mathscr{K}(\mathbf{x}_i, \mathbf{x}_j) - \widehat{\rho}\right)$$

so that any $\mathbf{x}_i$ with $\widehat{f}(\mathbf{x}_i) < 0$ is declared an outlier. The $\widehat{\alpha}_j$'s and $\widehat{\rho}$ 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

$$\mathscr{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{1}{2\sigma^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right\}$$

OCSVM has been extensively studied and applied by many researchers among which [103], [79] and [161], and later enhanced by [5]. 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.

4.2 MCD and PCOut

4.2.1 Minimum Covariance Determinant Estimators

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

$$\phi_p(\mathbf{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^p|\boldsymbol{\Sigma}|}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}. \tag{4.1}$$

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

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \varepsilon, \eta, \gamma) = (1-\varepsilon)\phi_p(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \varepsilon\phi_p(\mathbf{x}; \boldsymbol{\mu}+\eta, \gamma\boldsymbol{\Sigma}), \tag{4.2}$$

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

$$d^2_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{x}_i) = (\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}), \tag{4.3}$$

and experimenters of often address and tackle the outlier detection task in such situations using either the so-called Minimum Covariance Determinant (MCD) Algorithm [129] or some extensions or adaptations thereof.

Algorithm 4 Minimum Covariance Determinant (MCD)

```
1: Select $h$ observations, and form the dataset $\mathscr{D}_H$. $H \subset \{1, \cdots, n\}$.
2: Compute the empirical covariance $\widehat{\mathbf{\Sigma}}_H$ and mean $\widehat{\boldsymbol{\mu}}_H$.
3: Compute the Mahalanobis distances $d^2_{\widehat{\boldsymbol{\mu}}_H, \widehat{\mathbf{\Sigma}}_H}(\mathbf{x}_i), \quad i = 1, \cdots, n$
4: Select the $h$ observations having the smallest Mahalanobis distance.
5: Update $\mathscr{D}_H$ and repeat steps 2 to 5 until $\texttt{det}(\widehat{\mathbf{\Sigma}}_H)$ no longer decreases.
```

The MCD algorithm can be formulated as an optimization problem:

$$(\widehat{H}, \widehat{\boldsymbol{\mu}}_H, \widehat{\mathbf{\Sigma}}_H) = \underset{\boldsymbol{\mu}, \mathbf{\Sigma}, H}{\texttt{argmin}} \{\mathcal{E}(\boldsymbol{\mu}, \mathbf{\Sigma}, H)\} \tag{4.4}$$

where

$$\mathcal{E}(\boldsymbol{\mu}, \mathbf{\Sigma}, H) = \log\{\texttt{det}(\mathbf{\Sigma})\} + \frac{1}{h} \sum_{i \in H} (\mathbf{x}_i - \boldsymbol{\mu})^\top \mathbf{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}). \tag{4.5}$$

The seminal MCD algorithm proposed by [129] turned out to be rather slow and did not scale well as a function of the sample size n. That limitation of MCD led its author to creation of the so-called FAST-MCD [132], 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. It must be noted however 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 $O(p^3)$ where p is the dimensionality of the input space as defined earlier. It becomes crucial therefore 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. As noted before, with the MCD algorithm, h observations have to be selected to compute the robust estimator. Unfortunately, when $n \ll p$, neither the inverse nor the determinant of covariance matrix can be computed. As we'll show later, the $O(p^3)$ complexity of matrix inversion and determinant computatation renders MCD untenable for p as moderate as 500. It is therefore 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 [46], Regularized MCD (R-MCD) by [55] and other ideas by [7], [1], [57], [92]. When instability in the data makes the computation of $\widehat{\mathbf{\Sigma}}$ problematic in p dimension, regularized MCD may be used with objective function

$$\mathcal{E}(\boldsymbol{\mu}, \mathbf{\Sigma}, H, \lambda) = \mathcal{E}(\boldsymbol{\mu}, \mathbf{\Sigma}, H) + \lambda \texttt{trace}(\mathbf{\Sigma}^{-1}), \tag{4.6}$$

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 $\texttt{det}(\widehat{\mathbf{\Sigma}})$ is always zero in such cases. The solution to that added difficulty is addressed by solving:

$$\begin{aligned}\left(\widehat{H}, \widehat{\boldsymbol{\mu}}_H, \widehat{\mathbf{\Sigma}}_H\right) = \texttt{argmax}\Big\{ & \log\{\texttt{det}(\widetilde{\mathbf{\Sigma}})\} \\ & + \frac{1}{h}\sum_{i \in H} (\mathbf{x}_i - \boldsymbol{\mu})^\top \widetilde{\mathbf{\Sigma}}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) + \lambda \texttt{trace}(\widetilde{\mathbf{\Sigma}}^{-1})\Big\}\end{aligned}$$

where the regularized coveriance matrix $\widetilde{\mathbf{\Sigma}}$ is given by:

$$\widetilde{\mathbf{\Sigma}}(\alpha) = (1-\alpha)\widehat{\mathbf{\Sigma}} + \frac{\alpha}{p}\texttt{trace}(\widehat{\mathbf{\Sigma}})\boldsymbol{I}_p \tag{4.7}$$

with $\alpha \in (0, 1)$. For many HDLSS datasets however, the dimensionality p can reach $p \geqslant 10^3$ or even $p \geqslant 10^4$. As a result, even the above direct regularization is computationally intractable, because when p is large, the $O(p^3)$ complexity of the needed matrix inversion and determinant calculation makes the problem computationally untenable. The fastest matrix inversion algorithms like [27] and [97] are theoretically around $O(p^{2.376})$ and $O(p^{2.373})$, 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.

4.2.2 PCOut Algorithm for HDLSS Data

Another approach to outlier detection in the HDLSS context has revolved around extensions and adaptations of PCA that is *PCOut* as we mentioned before. 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 [76] whose proposed ROBPCA, a robust PCA method, which essentially robustifies PCA by combining MCD with the famous *projection pursuit* technique ([32], [98]). 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 appear to have motivated the creation of the so-called PCOut algorithm proposed by [46]. 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:

$$\mathbf{x}_j^* = \frac{\mathbf{x}_j - \widetilde{\mathbf{x}}_j}{MAD(\mathbf{x}_j)}, j = 1, \cdots, p, \tag{4.8}$$

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

$$\boldsymbol{X}^{*\top}\boldsymbol{X}^* = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top \tag{4.9}$$

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.

4.3 Random Subspace Learning Approach to Outlier Detection

4.3.1 Rationale for Random Subspace Learning

We herein propose a technique that combines the concept underlying Random subspace Method or, Random Subspace Learning (RSSL) by Ho[70] 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. Morever, 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 determinant of covariances computed from a sequence $\mathscr{D}_h^{(k)},\ k = 1, \cdots, m$ of different subsets of the original data set $\mathscr{D}$. Each subset contains h observations. More precisely, if $\mathscr{D}_{optimal}$ is the subset of $\mathscr{D}$ whose observations yield the estimated covariance matrix with the smallest (minimum) determinant out of all the m subsets considered, then we must have:

$$\texttt{det}(\widehat{\Sigma}(\mathscr{D}_{optimal})) = \min\left\{\texttt{det}(\widehat{\Sigma}(\mathscr{D}_h^{(1)})), \texttt{det}(\widehat{\Sigma}(\mathscr{D}_h^{(2)})), \cdots, \texttt{det}(\widehat{\Sigma}(\mathscr{D}_h^{(m)}))\right\},$$

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

$$\widehat{\boldsymbol{\mu}}_{\texttt{MCD}} = \widehat{\boldsymbol{\mu}}(\mathscr{D}_{optimal}) \quad \text{and} \quad \widehat{\boldsymbol{\Sigma}}_{\texttt{MCD}} = \widehat{\boldsymbol{\Sigma}}(\mathscr{D}_{optimal}).$$

The number h of observations in each subset is required to be $\frac{n}{2} \leq h < n$. It turns out that $h = [(n+p+1)/2]$ reaches its highest possible breakdown value according to [100]. It is obvious that with $h = [(n+p+1)/2]$ being the highest breakdown point, the requirement $\frac{n}{2} \leq h < n$ cannot 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 $\mathscr{X}$, and define/contruct 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.

4.3.2 Description of RSSL 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 ie sampling observations with replacement) with attribute-bagging (sampling indices of attributes without replacement), to allow efficient ensemble learning in high dimensional spaces. Here we present the algorithm in the form of a framework: 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.

Algorithm 5 Random Subspace Learning (RSSL): Attribute-bagging step

```
1: Randomly draw the number $d < p$ of variables to consider
2: Draw without replacement the indices of $d$ variables of the
   original $p$ variables
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.

Algorithm 6 Random Subspace Learning for Outlier Detection when $p \ll n$

```
procedure RANDOM SUBSPACE OUTLIER($B$)
    for $b = 1$ to $B$ do
        Draw with replacement $\{i_1^{(b)}, \cdots, i_n^{(b)}\}$ from $\{1, 2, \cdots, n\}$ to form the bootstrap sample $\mathscr{D}^{(b)}$
        Draw without replacement from $\{1, 2, \cdots, p\}$ a subset $\{j_1^{(b)}, \cdots, j_d^{(b)}\}$ of $d$ variables
        Drop unselected variables from $\mathscr{D}^{(b)}$ so that $\mathscr{D}_{sub}^{(b)}$ is $d$ dimensional
        Build the $b$th determinant of covariance $\texttt{det}(\widehat{\mathbf{\Sigma}}(\mathscr{D}_{sub}^{(b)}))$
    end for
    Sort the ensemble $\Big\{\texttt{det}(\widehat{\mathbf{\Sigma}}(\mathscr{D}_{sub}^{(b)})),\, b = 1, \cdots, B\Big\}$
    Form $\mathscr{D}^* : \texttt{det}(\mathscr{D}^*) = \texttt{argmin}\Big\{\texttt{det}(\widehat{\mathbf{\Sigma}}(\mathscr{D}_{sub}^{(b)})),\, b = 1, \cdots, B\Big\}$
    Compute $\widehat{\mu}^*$ and $\widehat{\mathbf{\Sigma}}^*$ base on $\mathscr{D}^*$
    We can build the robust distance by:
        $$\widehat{\delta}^*(\mathbf{x}) = (\mathbf{x} - \widehat{\mu}^*)^\top \, \widehat{\mathbf{\Sigma}}^{*^{-1}} \, (\mathbf{x} - \widehat{\mu}^*) \, . \qquad (4.10)$$
end procedure
```

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 $\min(\frac{n}{5}, \sqrt{p})$. 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 ($n \gg p$), our RSSL outlier detection algorithm always robustly yields the robust estimators $\widehat{\mu}^*$ and $\widehat{\mathbf{\Sigma}}^*$ 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:

Algorithm 7 Random Subspace Learning for Outlier Detection when $n \ll p$

```
procedure RANDOM SUBSPACE DETERMINANT COVARIANCE($B$)
    for $b = 1$ to $B$ do
        Draw with replacement $\{i_1^{(b)}, \cdots, i_n^{(b)}\}$ from $\{1, 2, \cdots, n\}$ to form the bootstrap sample $\mathscr{D}^{(b)}$
        Draw without replacement from $\{1, 2, \cdots, p\}$ a subset $\{j_1^{(b)}, \cdots, j_d^{(b)}\}$ of $d$ variables
        Drop unselected variables from $\mathscr{D}^{(b)}$ so that $\mathscr{D}_{sub}^{(b)}$ is $d$ dimensional
        Build the $b$th determinant of covariance $\texttt{det}(\widehat{\mathbf{\Sigma}}(\mathscr{D}_{sub}^{(b)}))$
    end for
    Sort the ensemble $\left\{\texttt{det}(\widehat{\mathbf{\Sigma}}(\mathscr{D}_{sub}^{(b)})),\ b = 1, \cdots, B\right\}$
    Keep the $k$ smallest samples based on elbow to form $\mathscr{D}^{(\eta)}$, where $\eta = 1, \cdots, k$,  $k < B$
    for $j = 2$ to $d$ do
        Select $\nu = j$ most frequent variables left in $\mathscr{D}^{(\eta)}$ to compute $\texttt{det}(\widehat{\mathbf{\Sigma}}(\mathscr{D}_{sub=j}^{(\eta=1)}))$
    end for
    Form $\mathscr{D}^* : \texttt{det}(\mathscr{D}^*) = \texttt{argmax}\left\{\texttt{det}(\widehat{\mathbf{\Sigma}}(\mathscr{D}_{sub=j}^{(\eta=1)})),\ j = 2, \cdots, d\right\}$
    Compute $\widehat{\mu}^*$ and $\widehat{\mathbf{\Sigma}}^*$ base on $\mathscr{D}^*$
    We can build the robust distance by:

        $\widehat{\delta}^*(\mathbf{x}) = (\mathbf{x} - \widehat{\mu}^*)^\top \, \widehat{\mathbf{\Sigma}}^{*^{-1}} \, (\mathbf{x} - \widehat{\mu}^*)\,.$

end procedure
```

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.

4.3.3 Justification RSSL for Outlier Detection

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

Let $\mathbf{x}_i \in \mathscr{D}$ be a random observation in the original dataset $\mathscr{D}$. Let $\mathscr{D}^{(b)}$ denote the bth bootstrapped sample from $\mathscr{D}$. Let $\Pr[\mathbf{x}_i \in \mathscr{D}^{(b)}]$ represent the proportion of observations that are in $\mathscr{D}$ but also present in $\mathscr{D}^{(b)}$. It is easy to prove $\Pr[\mathbf{x}_i \in \mathscr{D}^{(b)}] = 1 - \left(1 - \frac{1}{n}\right)^n$. In other words, if $\Pr[\mathbf{x}_i \notin \mathscr{D}^{(b)}] = \Pr[O_n]$ denotes the observations from $\mathscr{D}$ not present in $\mathscr{D}^{(b)}$, we must have $\Pr[\mathbf{x}_i \notin \mathscr{D}^{(b)}] = \left(1 - \frac{1}{n}\right)^n = \Pr[O_n]$. Since $\Pr[O_n]$ is known to converge to e^{-1} as n goes to infinity. Therefore for each given bootstrapped sample $\mathscr{D}^{(b)}$, 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 $\mathscr{D}^{(b)}$ 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.

4.3.4 RSSL Classification for High Dimensional Data

Since RSSL-MCD method that we discussed in last section can build robust space for the original data, thus the method can be applied to many of the classifiers especially for linear classifiers. Here we select the Fisher's linear discriminant analysis (LDA) as an example. Briefly speaking, for multivariate Gaussian density given class k we have:

$$f_k\left(\mathbf{x}\right) = \frac{1}{\left(2\pi\right)^p\left|\mathbf{\Sigma}_k\right|^{\frac{1}{2}}} e^{-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}_k\right)^\top \mathbf{\Sigma}_k^{-1}\left(\mathbf{x}-\boldsymbol{\mu}_k\right)}.$$

So for the hypothesis $\widehat{H}\left(\boldsymbol{X}\right)$ we find the optimum class for $\mathbf{x}$ by compute the probability:

$$\begin{aligned}\widehat{\delta}\left(\mathbf{x}\right) &= \arg\max_k Pr\left(H = k \mid \boldsymbol{X} = \mathbf{x}\right)\\ &= \arg\max_k f_k\left(\mathbf{x}\right)\pi_k\\ &= \log\left(\arg\max_k f_k\left(\mathbf{x}\right)\pi_k\right)\end{aligned}$$

Replace the density function of multivariate Gaussian, we can easily show that:

$$\widehat{\delta}\left(\mathbf{x}\right) = \arg\max_k\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top \mathbf{\Sigma}_k^{-1}\left(\mathbf{x}-\boldsymbol{\mu}_k\right) + \log\left(\pi_k\right)\right) \tag{4.11}$$

Thus, the estimation of mean and covariance can be replaced by our robust estimators $\widehat{\mu}^*$ and $\widehat{\mathbf{\Sigma}}^{*-1}$. Notice that the observations of the data are divided by k classes for LDA. However, since for variable selection we have to compute and rank the determinants of $\boldsymbol{X}$, the pooled covariance can be computed by:

$$\widehat{\mathbf{\Sigma}}_{pool} = \sum_{k=1}^{K}\frac{n-1}{N-k}\widehat{\mathbf{\Sigma}}_k$$

Thus, formally we have the random subspace learning for linear discriminant analysis *RSSL-LDA*:

Algorithm 8 Random Subspace Learning for LDA when $n \ll p$

```
procedure RANDOM SUBSPACE DETERMINANT COVARIANCE($B$)
    for $b = 1$ to $B$ do
        for $k = 1$ to $K$ do
            Draw with replacement $\{i_1^{(b)}, \cdots, i_{n_k}^{(b)}\}$ from $\{1, 2, \cdots, n_k\}$ to form the bootstrap sample $\mathscr{D}^{(b)}$
            Draw without replacement from $\{1, 2, \cdots, p\}$ a subset $\{j_1^{(b)}, \cdots, j_d^{(b)}\}$ of $d$ variables
            Drop unselected variables from $\mathscr{D}_k^{(b)}$ so that $\mathscr{D}_{sub_k}^{(b)}$ is $d$ dimensional
            Compute $\widehat{\mathbf{\Sigma}}_k$ from $\mathscr{D}_{sub_k}^{(b)}$
        end for
        Compute the pooled covariance $\widehat{\mathbf{\Sigma}}(\mathscr{D}_{subpool}^{(b)})$
        Build the $b$th determinant of covariance by pooled covariance $\texttt{det}(\widehat{\mathbf{\Sigma}}(\mathscr{D}_{subpool}^{(b)}))$
    end for
    Sort the ensemble $\left\{\texttt{det}(\widehat{\mathbf{\Sigma}}(\mathscr{D}_{subpool}^{(b)})),\ b = 1, \cdots, B\right\}$
    Keep the $z$ smallest samples based on elbow to form $\mathscr{D}^{(\eta)}$, where $\eta = 1, \cdots, z$,  $z < B$
    for $j = 2$ to $d$ do
        Select $\nu = j$ most frequent variables left in $\mathscr{D}^{(\eta)}$ to compute $\texttt{det}(\widehat{\mathbf{\Sigma}}(\mathscr{D}_{subpool=j}^{(\eta=1)}))$
    end for
    Form $\mathscr{D}^* : \texttt{det}(\mathscr{D}^*) = \texttt{argmax}\left\{\texttt{det}(\widehat{\mathbf{\Sigma}}(\mathscr{D}_{subpool=j}^{(\eta=1)})),\ j = 2, \cdots, d\right\}$
    Compute $\widehat{\mu}_k^*$ and $\widehat{\mathbf{\Sigma}}_k^*$ base on $\mathscr{D}_k^*$
    We can compute and select the probability of each class by:
        $$\widehat{\delta}^*(\mathbf{x}) = \arg\max_k \left( -\frac{1}{2}(\mathbf{x} - \widehat{\mu}_k^*)^\top \widehat{\mathbf{\Sigma}}_k^{*-1}(\mathbf{x} - \widehat{\mu}_k^*) + \log(\pi_k) \right)$$
end procedure
```

Chapter 5

Implementation Results and Conclusion

5.1 Computational Demonstrations

5.1.1 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 (4.1) and (4.2). In order to assess the effect the covariance between the attributes, we use an AR-type covariance matrix of the following form:

$$\Sigma = \begin{pmatrix} 1 & \rho & \cdots & \cdots & \rho \\ \rho & 1 & \rho & \cdots & \rho \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \rho & \ddots & \rho & 1 & \rho \\ \rho & \cdots & \cdots & \rho & 1 \end{pmatrix} = [(1-\rho)\boldsymbol{I}_p + \rho\mathbf{1_p}\mathbf{1_p}^\top], \tag{5.1}$$

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. ρ is selected between $\{0, 0.25\}$ to show the effect of correlation. 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 `R` within the package called `rrcovHD`.

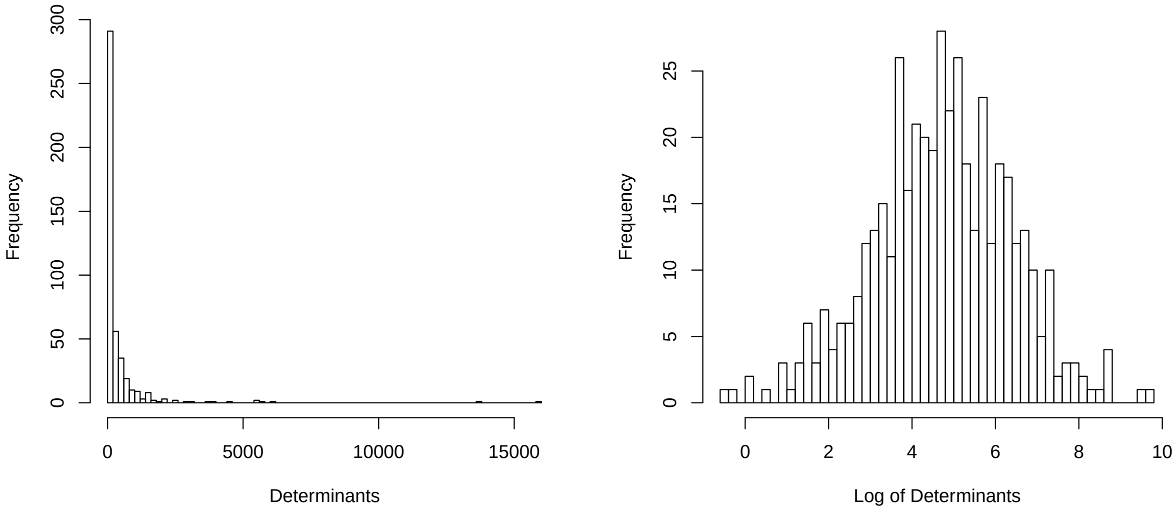

Figure 5.1: (left) Histogram of the distribution of the determinants from all bootstrap samples $\mathscr{D}^{(b)}_{\texttt{sub}}$ when $n = 100$, $p = 3000$; (right) Histogram of log determinants for all the bootstrap samples. Our methodology later selects a portion of samples based on what we call here the elbow.

As can be seen on Figure (5.1), 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.

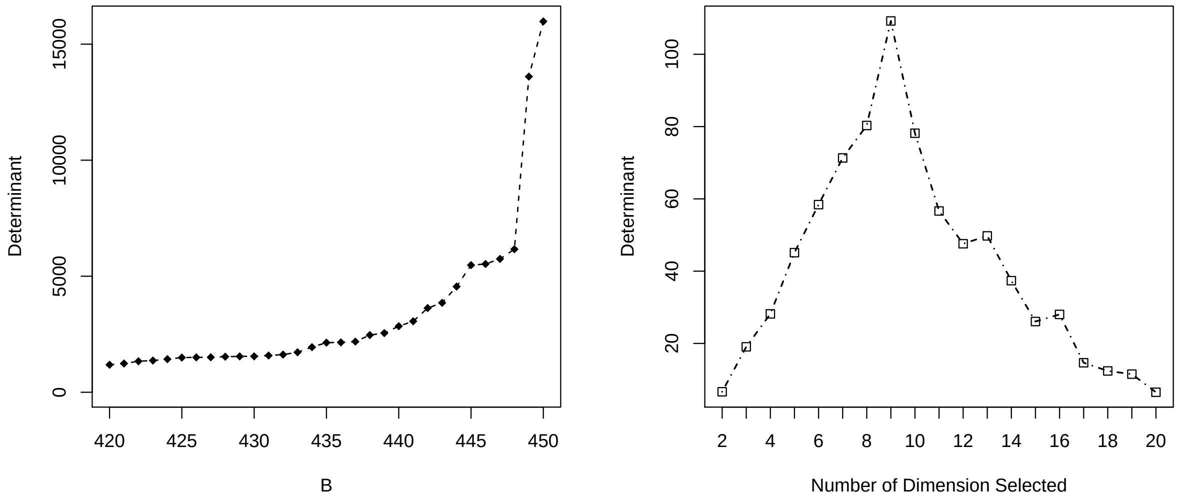

Figure 5.2: (left) Tail of sorted determinants in high dimensional $\mathscr{D}^{(b)}_{\texttt{sub}}$, where $B = 450$. k can be selected before reaching the elbow; (right) The concave shape can be observed by computing determinants of covariance from 2 to m dimension. The cut-off ν for variable selection is based on the decreasing sorted frequency located at the maximum of the determinants.

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 implies the robustness of the classifier. Figure (2.4) 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 (5.2). 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

$\mathscr{D}^{(\eta)}$ 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 $\mathscr{D}^{(\eta)}$ as we can observe in Figure (1.2). 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 (1.2). 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 $\mathscr{D}^{*} = \mathscr{D}^{(\eta=1)}_{sub=\nu}$. Then the robust estimators can be formed by $\widehat{\mu}^{*}$ and $\widehat{\Sigma}^{*}$. Theoretically then we are in a presence of a minimax formulation of our outlier detection problem, namely

$$\{\mathscr{D}^{(*)}, \mathscr{V}^{(*)}\} = \underset{\mathscr{V}^{(b)}}{\texttt{argmax}} \left\{ \underset{\mathscr{D}^{(b)}}{\texttt{argmin}} \{\texttt{det}(\texttt{cov}(\hat{\Sigma}(\mathscr{D}^{(b)}(\mathscr{V}^{(b)}))))\} \right\} \tag{5.2}$$

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

5.1.2 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 $\chi^2_{d,\frac{\alpha}{2}}$ as our cut-off with the typical $\alpha = 10\%$ and $\alpha = 5\%$. As usual, all observations with distances larger than $\chi^2_{d,\frac{\alpha}{2}}$ 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 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,

$$\mathtt{AVE}(\widehat{f}) = \frac{1}{R}\sum_{r=1}^{R}\left\{\frac{1}{m}\sum_{i=1}^{m}\ell(\mathbf{y}_i^{(r)}, \widehat{f}_r(\mathbf{x}_i^{(r)}))\right\}, \tag{5.3}$$

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

$$\ell(\mathbf{y}_i^{(r)}, \widehat{f}_r(\mathbf{x}_i^{(r)})) = 1_{\{\mathbf{y}_i^{(r)} \neq \widehat{f}_r(\mathbf{x}_i^{(r)})\}} = \begin{cases} 1 & \text{if } \mathbf{y}_i^{(r)} \neq \widehat{f}_r(\mathbf{x}_i^{(r)}) \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

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 (5.3) 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.

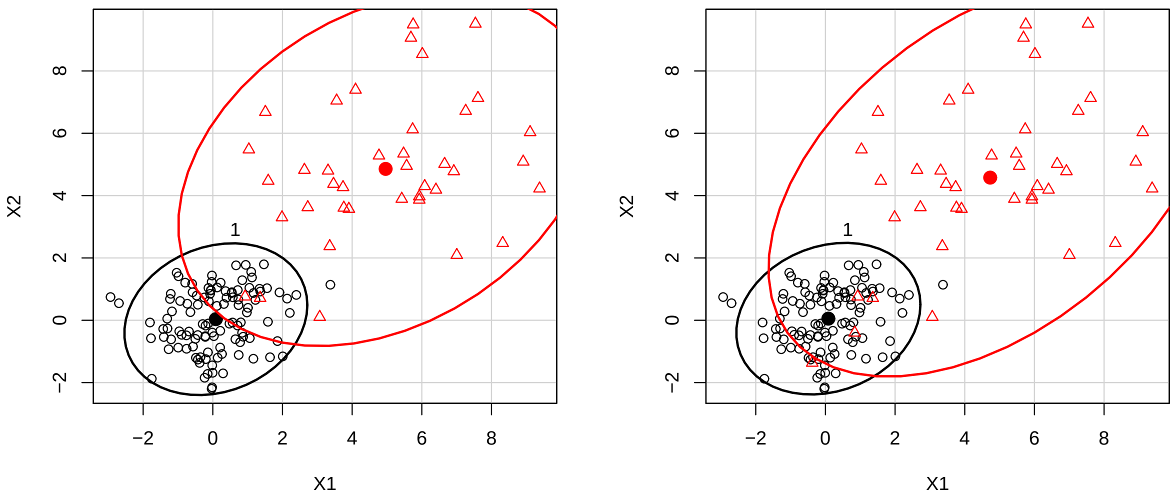

Figure 5.3: (left) The outliers detected in a two dimensional subspace are marked as red triangles. Selection is based on $\widehat{\delta}^*(\mathbf{x}) > \chi^2_{df=d,\alpha=5\%}$; (right) Outliers are selected by $\chi^2_{df=d,\alpha=10\%}$.

The improvement of our random subspace learning algorithm in low dimensional data with $p \in \{30, 40, 50, 60, 70\}$ and relative large sample size $n = 1500$, is demonstrated in figure (5.4) and (5.5) in comparison to *PCOut* and *PCDist*. Despite the correlation ρ may moderately affect both algorithms' performances that the most prominent changes are brought by κ and η. 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 a predictive detection rate close to 96% to 99%.

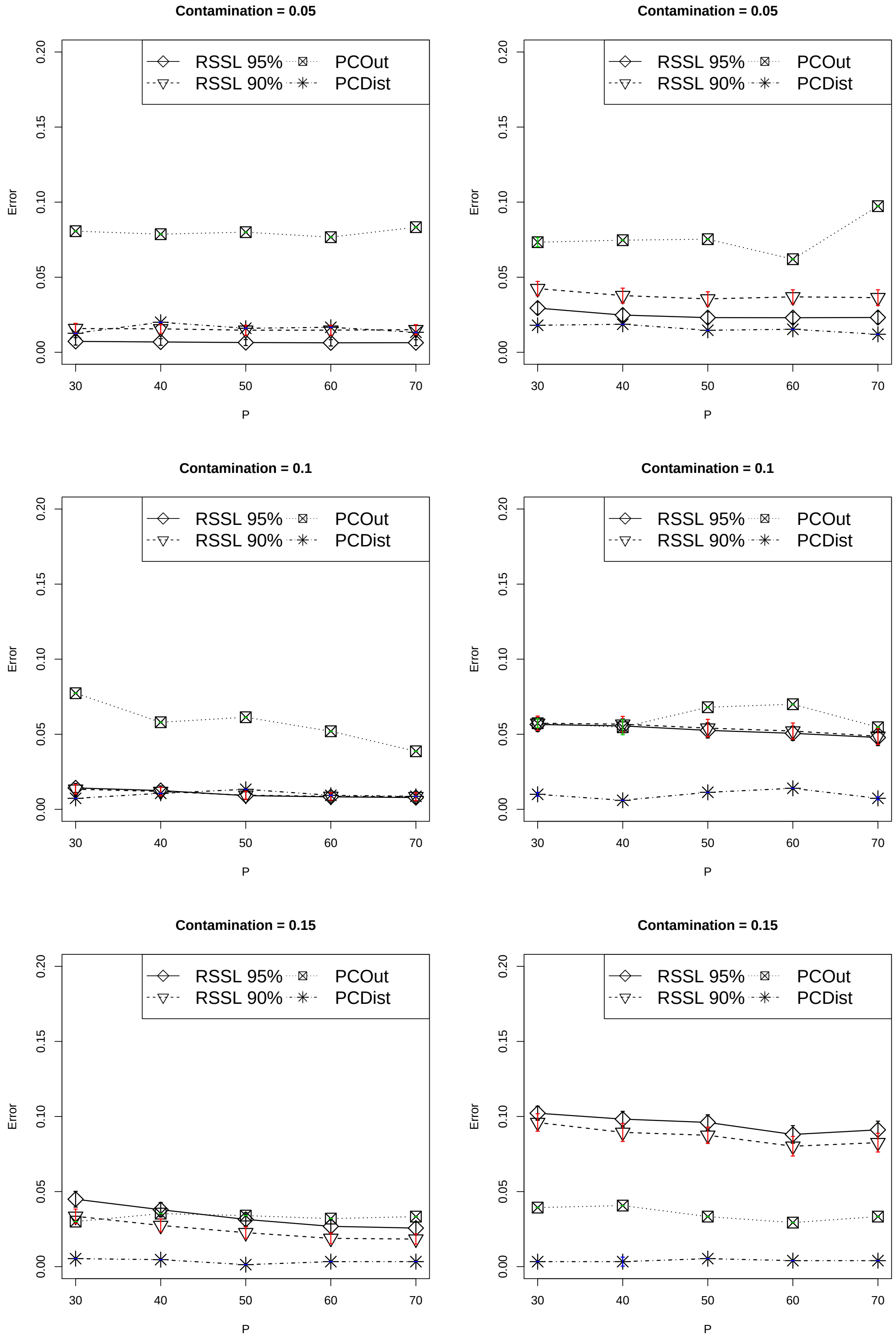

Figure 5.4: ($n = 1500, \rho = 0$) The average error and standard deviation in low dimensional simulation with $\kappa, \eta = 5$ (left column) and $\kappa, \eta = 2$ (right column).

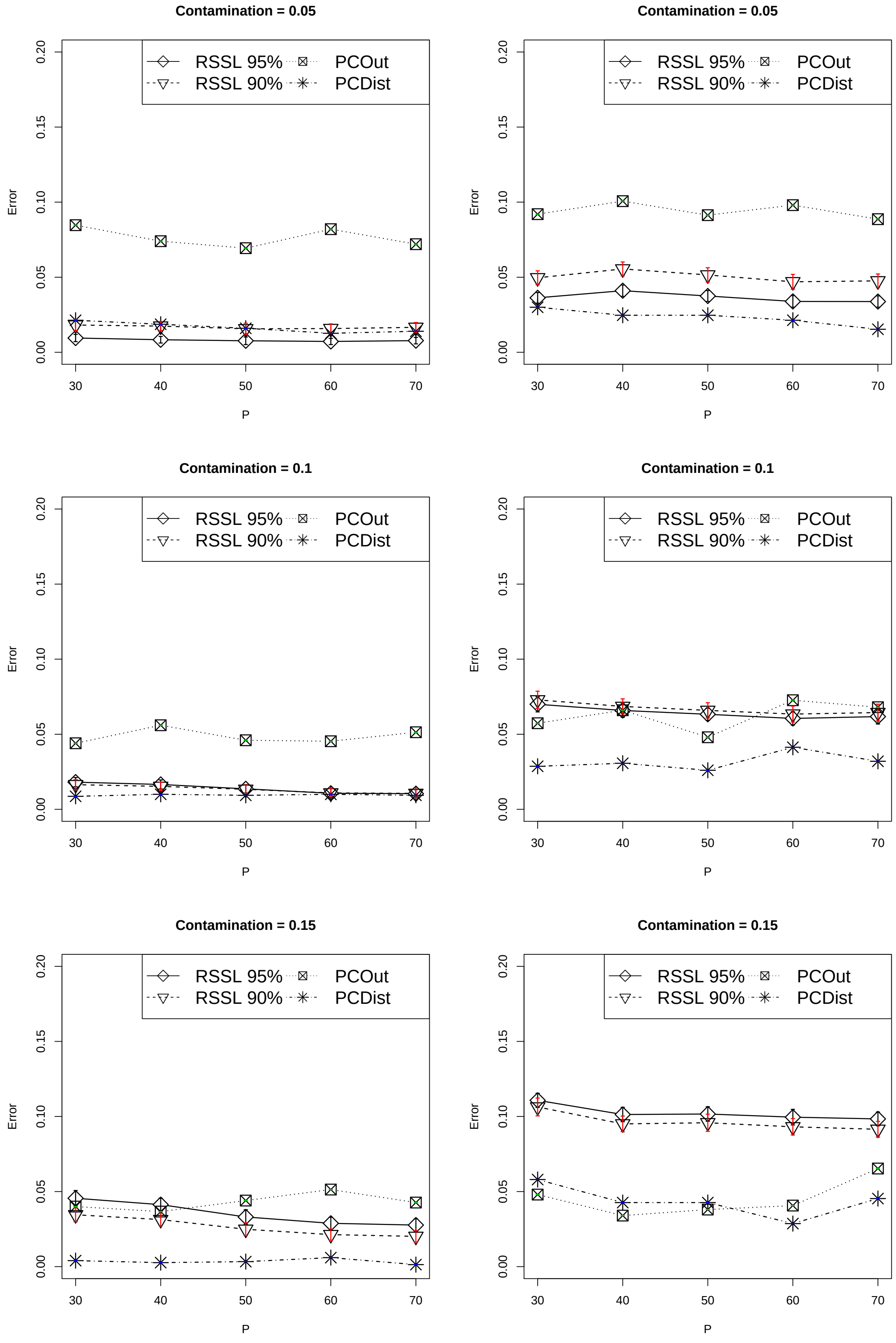

Figure 5.5: $(n = 1500, \rho = 0.25)$ The average error and standard deviation in low dimensional simulation $(n = 1500, \rho = 0)$ with $\kappa, \eta = 5$ (left column) and $\kappa, \eta = 2$ (right column).

In high dimensional settings, namely with $p \in \{1000, 2000, 3000, 4000, 5000\}$ and low sample size $n = 100$. Although the correlation ρ can slightly affect performance, RSSL is also performs reasonably well as shown in figure (5.9) when κ and η are relatively larger. However, as κ and η equals to while contamination rate is severe around 15%, the test is harder for RSSL that causes the accuracy reduce to 90%. When no correlation is added as in figure (5.7), 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 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 (5.7) and (5.9), 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.

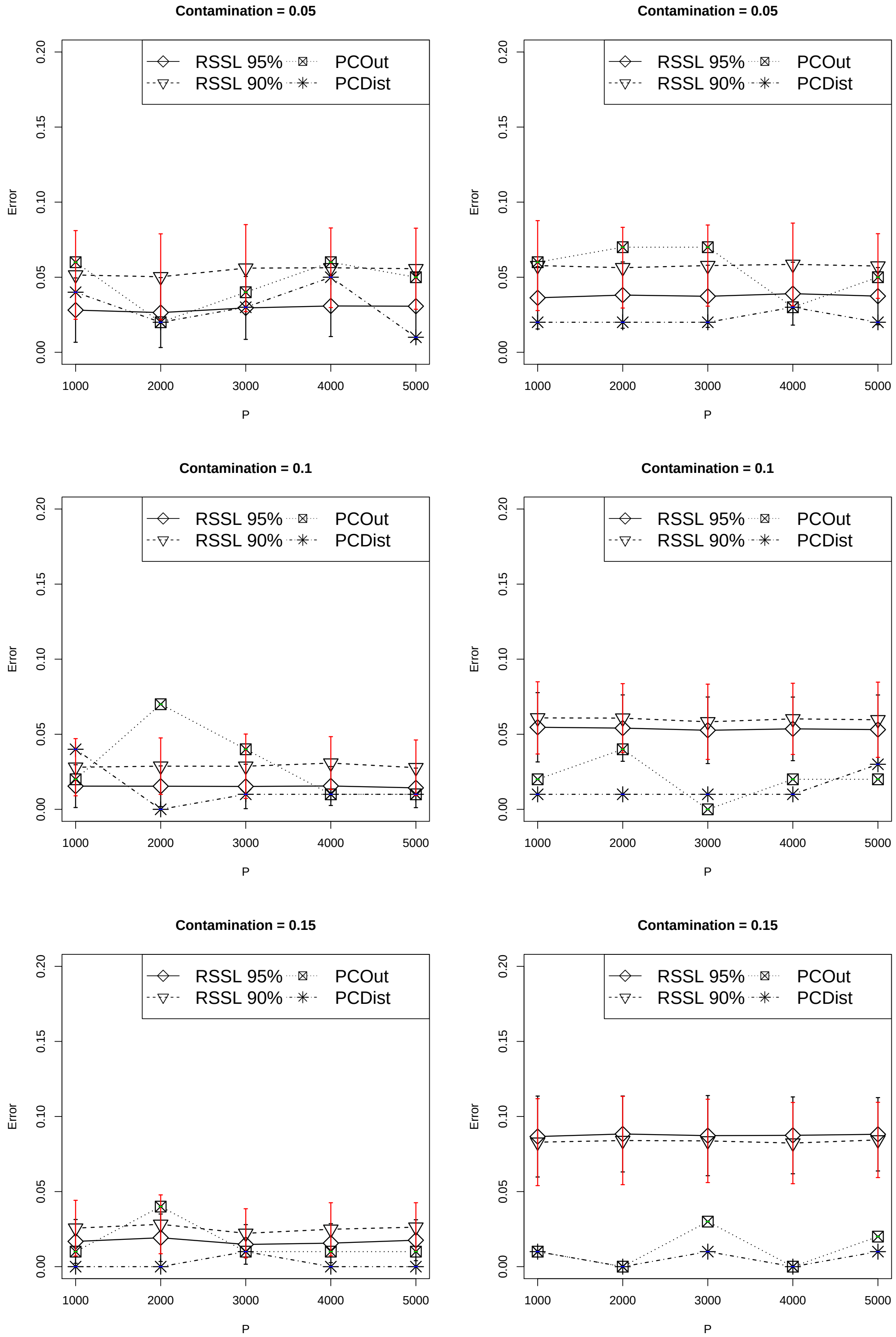

Figure 5.6: ($n = 100, \rho = 0$) The average error and standard deviation in high dimensional simulation with $\kappa, \eta = 5$ (left column) and $\kappa, \eta = 2$ (right column).

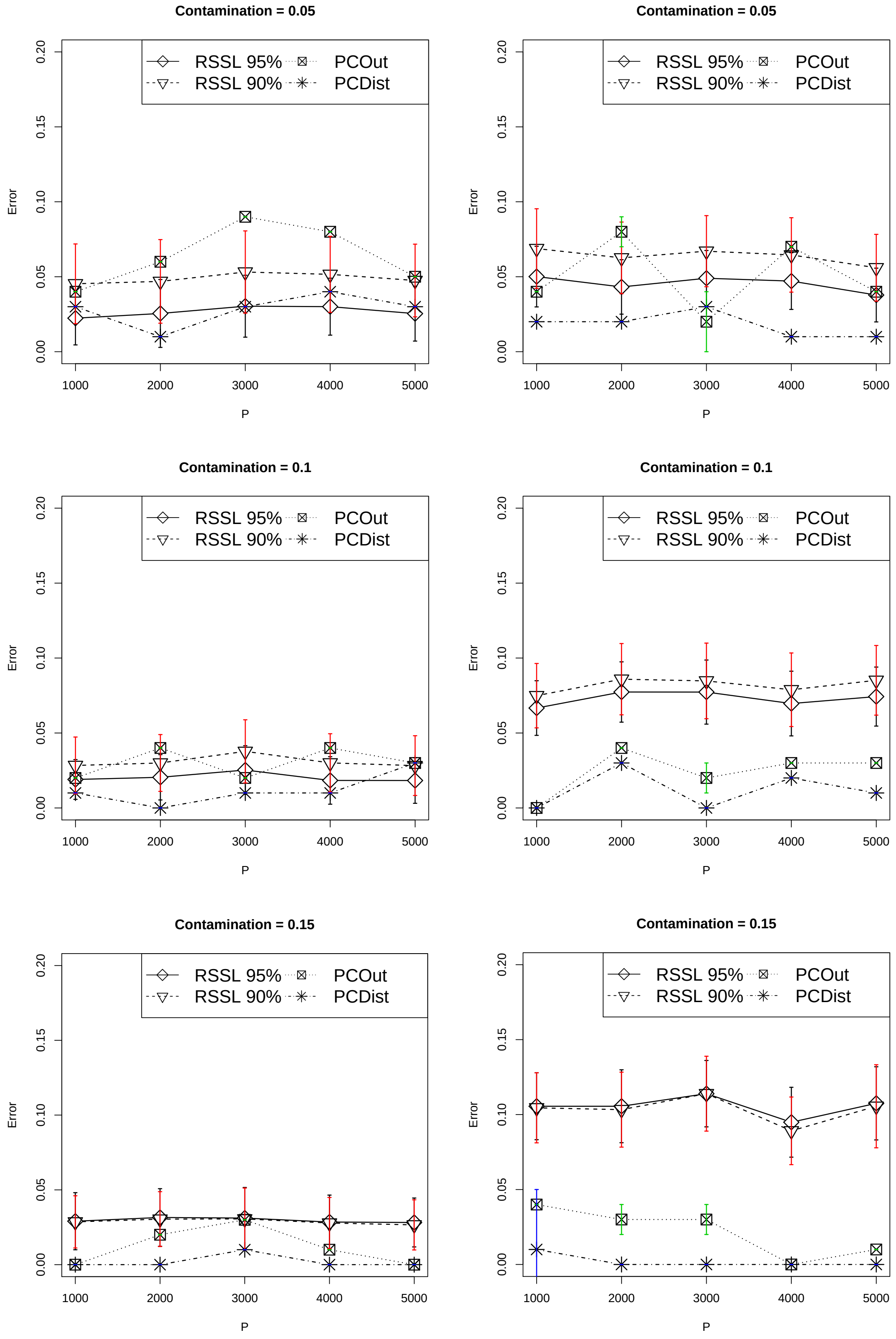

Figure 5.7: $(n = 100, \rho = 0.25)$ The average error and standard deviation in high dimensional simulation with $\kappa, \eta = 5$ (left column) and $\kappa, \eta = 2$ (right column).

5.2 Real Data Classification

5.2.1 The Leukemia Dataset

We consider the data from the cancer classification research conducted by Golub *et al.*. The goal of the research is to correctly distinguish between acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) from DNA-microarry gene expression dataset. The data set contains 47 patients with ALL and 25 patients with AML that adds up to 72 observations. There are 6817 variables that representing human genes and each value of data is the expression level measured by Affymetrix high density oligonucleotide arrays. Here we use a subset of the data because some bioinformatics filtering need to be taken as a preprocessing steps. This procedure is performed by:

- Eliminate the variables with extreme values that less than 100 and larger than 16000
- A base 10 logarithmic transformation is performed for the whole dataset
- Exclude the variables that have transformed observations with value: $\max / \min \leqslant 5$ or $\max - \min \leqslant 500$

Thus, the filtered dataset has unchanged observations $n = 72$ but dimensions $p = 3571$. Such threshold was frequently used by researchers such as [37]. The preprocessed data is already available in `R` package `spikeslab`. Then we can applied our RSSL-LDA and compare with other popular algorithms. *SVM* with Radial Basis (Gaussian) kernel and *RandomForest* are selected due to their adaptation of such high dimensionality. Since this is a real world data situation, we concern large portion of the variables of the benchmark leukemia dataset has the high probability to be correlated, an weighting scheme that taking advantages of F-statistic is adopted by us that to reduce the chance of repeatedly select redundancies. For each feature we give it a weight w_i where $i = 1, 2, \cdots, p$ according its F-statistic with respect to the response $\mathbf{y}$, such procedure is taken before bootstrap aggregation.

5.2.2 Prediction Results

We still use $B = 450$ bootstrap samples for each run of the algorithm, and replicate $R = 200$ times. Since the ratio of between the devotionality and the number of observations is extremely unbalanced, top 60% of bootstrap samples that are ranked by determinant are used to select z of the most frequently appeared variables. On Figure (5.6), just like the situation in our previous outlier detection experiment, huge amount of samples with determinants that are very close to zero which leads an obvious scene of heavy right skewness. This is again a strong evidence that we need to rebuild our estimators to achieve robustness inside of this extremely noisy dataset.

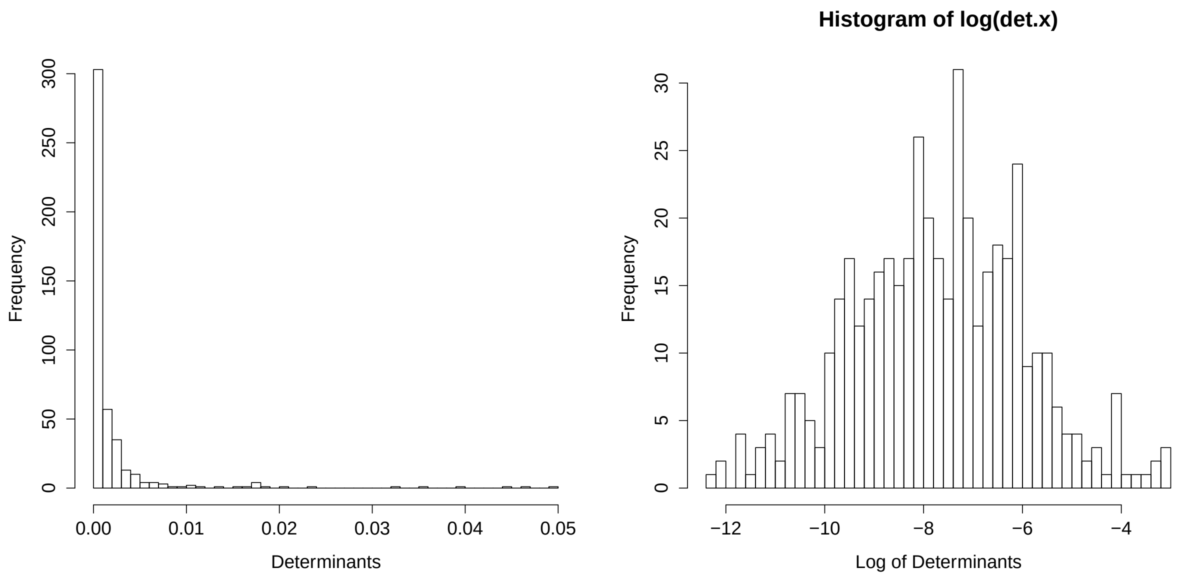

Figure 5.8: The frequency of determinant and (right) log determinant of all random subspaces from leukemia dataset.

For most of situations, the way way determine the value of z can work well like the previous outlier simulation study. The most frequent appeared variables are added one by one and the determinant is computed accumulatively. z equals the dimensionality that has the maximum determinant. However, we a real dataset with massive amount of redundancy is encountered, this way may not work perfectly due to its unexpected complexity among all subspaces. Thus, we may perform a cross validation with values in $\{2, 3, \cdots, z^*\}$ if the maximum is not available since we can roughly estimate a range from previous loops. In this example, we choose $z^* = 10$ and a 3-fold cross validation is performed due to the limited number of observations in each class (47 and 25). Figure (5.7) shows an example of $z = 3$ when a maximum can be obtained.

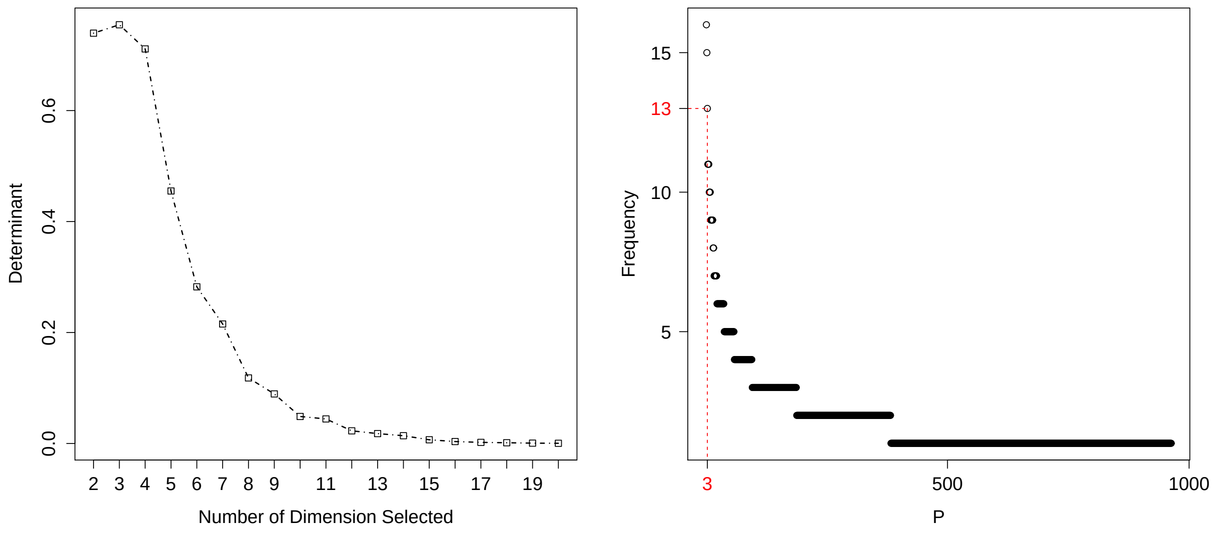

Figure 5.9: (left) The maximum determinant can be obtained from (right) the first 3 most frequently appeared variables.

On Figure (5.9), the result of comparison with SVM with RBF kernel and random forests classifiers in terms of accuracy is shown. The same 3-fold cross validation is performed for both SVM and RF on the training data is performed to assess the quality of their models. The mean error rate of RSSL-LDA is roughly 5% that 2.5% lower than the tuned SVM and RF and the standard deviations are close to each other. Though the practical performances of our algorithm still needs some improvements, there are considerably amount of space that we can explore.

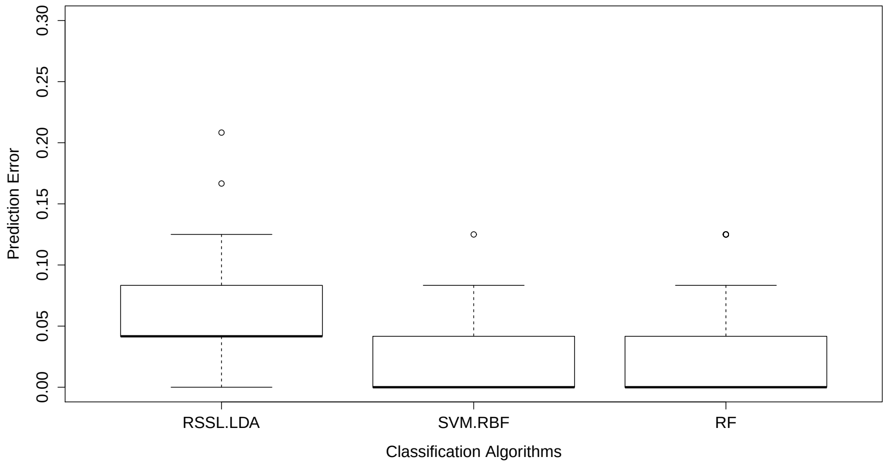

Figure 5.10: Box plot of prediction errors of RSSL-LDA, SVM-RBF and RF.

5.3 Conclusion

We have presented what we can rightfully claim to be a computational efficient, scalable, intuitive appealing and highly predictively accurate outlier detection and classification 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 and classification tasks on high dimension low sample size datasets like DNA Microarray Gene Expression datasets for which the MCD approach proved to be computational untenable. As our computational and real data 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 in terms of accuracy for both low and high dimensional space outlier detection, high dimensional data classification and is computationally very efficient.

Our future interests on of the random subspace frame work can be divided in to two directions. We can examine some function $\mathcal{F}$, different weights and dynamic way of selecting variables that can break down the potential of correlation to efficiently combine linear classifiers or, we can simply experiment on different classifiers. Furthermore, we can extend our field of studies to model aggregation by various weighting, theoretical upper bound and oracle inequalities for convex aggregates.

References

Bibliography

[1] C. Aggarwal and S. Yu. An effective and efficient algorithm for high-dimensional outlier detection. *The VLDB Journal*, 14(2):211–221, April 2005.

[2] Charu C Aggarwal and Philip S Yu. Outlier detection for high dimensional data. In *ACM Sigmod Record*, volume 30, pages 37–46. ACM, 2001.

[3] Ash A Alizadeh, Michael B Eisen, R Eric Davis, Chi Ma, Izidore S Lossos, Andreas Rosenwald, Jennifer C Boldrick, Hajeer Sabet, Truc Tran, Xin Yu, et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, 2000.

[4] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissue probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America,*, 96(12):6745–6750, 1999.

[5] Mennatallah Amer, Markus Goldstein, and Slim Abdennadher. Enhancing one-class support vector machines for unsupervised anomaly detection. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, ODD ’13, pages 8–15, New York, NY, USA, 2013. ACM.

[6] Yali Amit, Donald Geman, and Kenneth Wilder. Joint induction of shape features and tree classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(11):1300–1305, 1997.

[7] Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *Principles of data mining and knowledge discovery*, pages 15–27. Springer, 2002.

[8] N. N. Author. Suppressed for anonymity, 2011.

[9] Thomas F Banchoff. *Beyond the third dimension*. Scientific American Library New York, 1990.

[10] Maurice S Bartlett. Properties of sufficiency and statistical tests. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, pages 268–282, 1937.

[11] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1-2):105–139, 1999.

[12] Richard Bellman. Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences*, 42(10):767–769, 1956.

[13] Eugenio Beltrami. Sulle funzioni bilineari. *Giornale di Matematiche ad Uso degli Studenti Delle Universita*, 11(2):98–106, 1873.

[14] Alberto Bertoni, Raffaella Folgieri, and Giorgio Valentini. Bio-molecular cancer prediction with random subspace ensembles of support vector machines. *Neurocomputing*, 63(0):535 – 539, 2005. New Aspects in Neurocomputing: 11th European Symposium on Artificial Neural Networks.

[15] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is nearest neighbor meaningful? In *Database theoryICDT99*, pages 217–235. Springer, 1999.

[16] S. Bicciato, A. Luchini, and C. Di Bello. Pca disjoint models for multiclass

cancer analysis using gene expression data. *Bioinformatics,*, 19(5):571–578, 2003.

[17] Nedret Billor, Ali S Hadi, and Paul F Velleman. Bacon: blocked adaptive computationally efficient outlier nominators. *Computational Statistics & Data Analysis*, 34(3):279–298, 2000.

[18] Peter Blmann and Torsten Hothorn. Boosting algorithms: Regularization, prediction and model fitting. *Statist. Sci.*, 22(4):477–505, 11 2007.

[19] George EP Box. Robustness in the strategy of scientific model building. *Robustness in statistics*, 1:201–236, 1979.

[20] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[21] Leo Breiman. Bias, variance, and arcing classifiers. *Tech. Rep. 460, Statistics Department, University of California, Berkeley, CA, USA*, 1996.

[22] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[23] RW Butler, PL Davies, and M Jhun. Asymptotics for the minimum covariance determinant estimator. *The Annals of Statistics*, pages 1385–1400, 1993.

[24] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[25] Bertrand Clarke, Ernest Fokoue, and Hao Helen Zhang. *Principles and theory for data mining and machine learning*. Springer Science & Business Media, 2009.

[26] R.A. Cooper and T.J. Weekes. *Data, Models, and Statistical Analysis*. Barnes and Noble,, New York, 1983.

[27] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251 – 280, 1990. Computational algebraic complexity editorial.

[28] C. Croux and C. Dehon. Robust linear discriminant analysis using s-estimators. *The Canadian Journal of Statistics,*, 29(2):473–493, 2001.

[29] C. Croux, P. Filzmoser, and K. Joossens. Robust linear discriminant analysis for multiple groups: Influence and classification efficiencies, 2005.

[30] C. Croux, P. Filzmoser, K. Joossens, and K.U. Leuven. Classification efficiencies for robust discriminant analysis. *Statistica Sinica,*, 18:581–599, 2008.

[31] C. Croux and K. Joossens. Influence of observations on the misclassification probability in quadratic discriminant analysis. *Journal of Multivariate Analysis,*, 96:384–403, 2005.

[32] Christophe Croux and Anne Ruiz-Gazen. A fast algorithm for robust principal components based on projection pursuit. In *Compstat*, pages 211–216. Springer, 1996.

[33] M. Daszykowski, K. Kaczmarek, I. Stansmirova, Y. Vander Hayden, and B. Walczak. Robust simca-bounding influence of outliers. *Chemometrics and Intelligent Laboratory System,*, 87:95–103, 2007.

[34] Misha Denil, David Matheson, and Nando de Freitas. Narrowing the gap: Random forests in theory and in practice. In *International Conference on Machine Learning (ICML)*, 2014.

[35] David L Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, pages 1–32, 2000.

[36] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, 2nd edition, 2000.

[37] S. Dudoit, Fridlyand J., and T.P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association,*, 97(457):77–87, 2002.

[38] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[39] László Erdős, Antti Knowles, Horng-Tzer Yau, Jun Yin, et al. Spectral statistics of erdős–rényi graphs i: Local semicircle law. *The Annals of Probability*, 41(3B):2279–2375, 2013.

[40] Jianqing Fan and Jinchi Lv. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1):101, 2010.

[41] P. Filzmoser, K. Joossens, and C. Croux. Multiple group linear discriminant analysis: Robustness and error rate. In A. Rizzi and M. Vichi, editors, *COMPSTAT 2006-Proceedings in Computational Statistics,*, pages 521–532. Physica-Verlag, Heidelberg, 2006.

[42] P. Filzmoser, S. Serneels, C. Croux, and P. J. Van Espen. Robust multivariate methods: The projection pursuit approach. In M. Spiliopoulou, R. Kruse, C. Borgelt, A. Nurnberger, and W. Gaul, editors, *From Data and Information Analysis to Knowledge Engineering,*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 270–277. Springer Berlin Heidelberg, 2006.

[43] P. Filzmoser, S. Serneels, R. Maronna, and P.J. Van Espen. Robust multivariate methods in chemometrics. In B. Walczak, R.T. Ferre, and S. Brown, editors, *Comprehensive Chemometrics,*, pages 681–722. na, 2009.

[44] P. Filzmoser and V. Todorov. Review of robust multivariate statistical methods in high dimension. *Analytica Chimica Acta,*, 705(1-2):2–14, 2011.

[45] P. Filzmoser and V. Todorov. Robust tools for the imperfect world. *Information Sciences,*, 245:4–20, 2013.

[46] Peter Filzmoser, Ricardo Maronna, and Mark Werner. Outlier identification

in high dimensions. *Computational Statistics & Data Analysis*, 52(3):1694 – 1711, 2008.

[47] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256 – 285, 1995.

[48] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm, 1996.

[49] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997.

[50] J. H. Friedman. Exploratory Projection Pursuit. *Journal of the American Statistical Association*, 82(397):249–266, 1987.

[51] J. H. Friedman and W. Stuetzle. Projection Pursuit Regression. *Journal of the American Statistical Association*, 76(376):817–823, 1981.

[52] J. H. Friedman and J. W. Tukey. A Projection Pursuit algorithm for exploratory data analysis. *IEEE Trans. on computers*, 23(9):881–890, 1974.

[53] J. H. Friedman and J.W. Tukey. A projection pursuit algorithm ror exploratory data analysis. In *IEEE TRANSACTIONS ON COMPUTERS,*. September, 1974.

[54] Jerome H. Friedman and Usama Fayyad. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1:55–77, 1997.

[55] Virgile Fritsch, Gael Varoquaux, Benjamin Thyreau, Jean-Baptiste Poline, and Bertrand Thirion. Detecting outlying subjects in high-dimensional neuroimaging datasets with regularized minimum covariance determinant. In Gabor

Fichtinger, Anne Martel, and Terry Peters, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2011*, volume 6893 of *Lecture Notes in Computer Science*, pages 264–271. Springer Berlin Heidelberg, 2011.

[56] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

[57] Amol Ghoting, Srinivasan Parthasarathy, and Matthew Eric Otey. Fast mining of distance-based outliers in high-dimensional datasets. *Data Mining and Knowledge Discovery*, 16(3):349–364, 2008.

[58] T.R. et. al. Golub. Molecular classification of cancer:discovery and class prediction by gene expression monitoring. *Science,*, 286:531537, 1999. doi: 10.1126/science.286.5439.531.

[59] Frank E Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, 1969.

[60] Y. Guo, T. Hastie, and R. Tibshirani. Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, 8(1):86–100, 2006.

[61] P. Hall. On polynomial-based projection indices for exploratory projection pursuit. *The Annals of Statistics*, 17(2):589–605, 1990.

[62] P.Y. Han and A.T.B. Jin. Random projection with robust linear discriminant analysis model in face recognition. In *CGIV '07, Computer Graphics, Imaging and Visualisation,*, pages 11–15, 2007.

[63] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(10):993–1001, October 1990.

[64] D. M. Hawkins and G. J. McLachlan. High-breakdown linear discriminant

analysis. *Journal of the American Statistical Association,*, 92(437):136–143, 1997.

[65] X He and W.K. Fung. High breakdown estimation for multiple populations with applications to discriminant analysis. *Journal of Multivariate Analysis,*, 72(2):151–162, 2000.

[66] Xuming He and Gang Wang. Cross-checking using the minimum volume ellipsoid estimator. *Statistica Sinica*, pages 367–374, 1996.

[67] Katherine A. Heller, Krysta M. Svore, Angelos D. Keromytis, and Salvatore J. Stolfo. One class support vector machines for detecting anomalous windows registry accesses. In *In Proc. of the workshop on Data Mining for Computer Security*, 2003.

[68] T Hesterberg, DS Moore, S Monaghan, A Clipson, and R Epstein. Introduction to the practice of statistics. *Bootstrap methods and permutation tests*, pages 14–1, 2005.

[69] Tin Kam Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

[70] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):832–844, August 1998.

[71] Tin Kam Ho. Data complexity analysis for classifier combination. In *Multiple Classifier Systems*, pages 53–67. Springer, 2001.

[72] Arthur E Hoerl. Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 58(3):54–59, 1962.

[73] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[74] P. J. Huber. Projection Pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[75] M. Hubert and M. Debruyne. Minimum covariance determinant. *WIREs Comp Stat, 2: 3643. doi: 10.1002/wics.61*, 2:3643, 2010.

[76] M. Hubert and S. Engelen. Robust pca and classification in biosciences. *Bioinformatics,*, 20(11):1728–1736, 2004.

[77] M. Hubert, P.J. Rousseeuw, and S. Van Aelst. High-breakdown robust multivariate methods,. *Statistical Science*, 23:92–119, 2008.

[78] M. Hubert and K. Van Driessen. Fast and robust discriminant analysis. *Computational Statistics and Data Analysis,*, 45(2):301–320, 2004.

[79] Mia Hubert, Peter J Rousseeuw, and Karlien Vanden Branden. Robpca: a new approach to robust principal component analysis. *Technometrics*, 47(1):64–79, 2005.

[80] M. C. Jones and R. Sibson. What is projection pursuit? *Journal of the Royal Statistical Society, Ser. A*, 150:1–36, 1987.

[81] K. Joossens and C. Croux. Empirical comparison of the classification performance of robust linear and quadratic discriminant analysis. In M. Hubert, G. Pison, A. Struyf, and S. Van Aelst, editors, *Theory and Applications of Recent Robust Methods,*, Statistics for Industry and Technology, pages 131–140. Birkhauser Basel, 2004. ISBN 978-3-0348-9636-8.

[82] Camille Jordan. Mémoire sur les formes bilinéaires. *Journal de mathématiques pures et appliquées*, 19:35–54, 1874.

[83] M. J. Kearns. *Computational Complexity of Machine Learning*. PhD thesis, Department of Computer Science, Harvard University, 1989.

[84] Michael Kearns. Thoughts on hypothesis boosting. *Unpublished manuscript*, 45:105, 1988.

[85] Fabian Keller, Emmanuel Müller, and Klemens Böhm. Hics: high contrast subspaces for density-based outlier ranking. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 1037–1048. IEEE, 2012.

[86] S-J. Kim, A. Magnani, and S.P. Boyd. Robust fisher discriminant analysis. *Advances in Neural Information Processing Systems,*, 2005.

[87] S. Klinke and J. Grassmann. Projection pursuit regression. *Wiley Series in Probability and Statistics*, pages 471–496, 2000.

[88] Y. Kondo, M. Salibian-Barrera, and R. Zamar. A robust and sparse k-means clustering algorithm. *arXiv preprint arXiv*, 2012.

[89] Mario Köppen. The curse of dimensionality. In *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*, pages 4–8, 2000.

[90] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[91] H Kriegel, Peer Kroger, Eugen Schubert, and Arthur Zimek. Outlier detection in arbitrarily oriented subspaces. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 379–388. IEEE, 2012.

[92] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In *Advances in Knowledge Discovery and Data Mining*, pages 831–838. Springer, 2009.

[93] L.I. Kuncheva, J.J. Rodriguez, C.O. Plumpton, D.E.J. Linden, and S.J. Johnston. Random subspace ensembles for fmri classification. *Medical Imaging, IEEE Transactions on*, 29(2):531–542, Feb 2010.

[94] Ludmila I Kuncheva, Juan J Rodríguez, Catrin O Plumpton, David EJ Linden, and Stephen J Johnston. Random subspace ensembles for fmri classification. *Medical Imaging, IEEE Transactions on*, 29(2):531–542, 2010.

[95] Ludmila I Kuncheva, Fabio Roli, Gian Luca Marcialis, and Catherine A Shipp. Complexity of data subsets generated by the random subspace method: an experimental investigation. In *Multiple Classifier Systems*, pages 349–358. Springer, 2001.

[96] P. Langley. Crafting papers on machine learning. In Pat Langley, editor, *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pages 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

[97] Francois Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 296–303, New York, NY, USA, 2014. ACM.

[98] Guoying Li and Zhonglian Chen. Projection-Pursuit approach to robust dispersion matrices and principal components: Primary theory and monte carlo. *Journal of the American Statistical Association*, 80(391):759–766, 1985.

[99] Xuchun Li, Lei Wang, and Eric Sung. Adaboost with svm-based component classifiers. *Engineering Applications of Artificial Intelligence*, 21(5):785–795, 2008.

[100] Hendrik P. Lopuhaa and Peter J. Rousseeuw. Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *Ann. Statist.*, 19(1):229–248, 03 1991.

[101] Richard Maclin and David Opitz. An empirical evaluation of bagging and boosting. *AAAI/IAAI*, 1997:546–551, 1997.

[102] Edward C Malthouse. Limitations of nonlinear pca as performed with generic neural networks. *Neural Networks, IEEE Transactions on*, 9(1):165–173, 1998.

[103] Larry M Manevitz and Malik Yousef. One-class svms for document classification. *the Journal of machine Learning research*, 2:139–154, 2002.

[104] Ricardo A Maronna and Ruben H Zamar. Robust estimates of location and dispersion for high-dimensional datasets. *Technometrics*, 2012.

[105] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors. *Machine Learning: An Artificial Intelligence Approach, Vol. I*. Tioga, Palo Alto, CA, 1983.

[106] T. M. Mitchell. The need for biases in learning generalizations. Technical report, Computer Science Department, Rutgers University, New Brunswick, MA, 1980.

[107] Klaus-Robert Mller, Sebastian Mika, Gunnar Rtsch, Koji Tsuda, and Bernhard Schlkopf. An introduction to kernel-based learning algorithms. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 12(2):181–201, 2001.

[108] Emmanuel Müller, Ira Assent, Uwe Steinhausen, and Thomas Seidl. Outrank: ranking outliers in high dimensional data. In *Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on*, pages 600–603. IEEE, 2008.

[109] Praneeth Netrapalli, UN Niranjan, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. Non-convex robust pca. In *Advances in Neural Information Processing Systems*, pages 1107–1115, 2014.

[110] A. Newell and P. S. Rosenbloom. Mechanisms of skill acquisition and the law of practice. In J. R. Anderson, editor, *Cognitive Skills and Their Acquisition*, chapter 1, pages 1–51. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1981.

[111] Jerzy Neyman and Egon S Pearson. *On the problem of the most efficient tests of statistical hypotheses*. Springer, 1992.

[112] Hoang Vu Nguyen, Vivekanand Gopalkrishnan, and Ira Assent. An unbiased distance-based outlier detection approach for high-dimensional data. In *Database Systems for Advanced Applications*, pages 138–152. Springer, 2011.

[113] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

[114] F. Z. Okwonu and A. R. Othman. Robust mlcr linear classification technique: An application to classify aede albopictus mosquito. *International Journal of Computer Science Issues,*, 10(6):266–270, 2013.

[115] J-X. Pan, W-K. Fung, and K-T. Fang. Multiple outlier detection in multivariate data using projection pursuit techniques. *Journal of Statistical Planning and Inference*, 83(1):153–167, 2000.

[116] Pance Panov and Saso Dzeroski. Combining bagging and random subspaces to create better ensembles. In Michael R. Berthold, John Shawe-Taylor, and Nada Lavrafc, editors, *Advances in Intelligent Data Analysis VII*, volume 4723 of *Lecture Notes in Computer Science*, pages 118–129. Springer Berlin Heidelberg, 2007.

[117] Daniel Pea and Francisco J. Prieto. Multivariate outlier detection and robust covariance matrix estimation. *Technometrics*, 43:286–310, 2001.

[118] K Person. On lines and planes of closest fit to system of points in space. philiosophical magazine, 2, 559-572, 1901.

[119] A. M. Pires. Projection-pursuit approach to robust linear discriminant analysis. *Journal Multivariate Analysis,*, 101(10):2464–2485, 2010.

[120] A.M. Pires. Robust linear discriminant analysis and the projection pursuit approach. In R. Dutter, P. Filzmoser, U. Gather, and P. J. Rousseeuw, editors, *Developments in Robust Statistics,*, pages 317–329. Physica-Verlag HD, 2003. ISNB: 978-3-642-63241-9.

[121] G. Pison, S. Van Aelst, and G. Willems. Small sample corrections for lts and mcd. *Metrika,*, 55(1-2):111–123, 2002.

[122] Jrg Polzehl and Deutsche Forschungsgemeinschaft. Projection pursuit discriminant analysis. *Computational Statistics and Data Analysis*, 20:141–157, 1993.

[123] S. et al. Pomeroy. Prediction of central nervous system embryonal tumor outcome based on gene expression. *Nature,*, 415:436–442, 2002.

[124] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *ACM SIGMOD Record*, volume 29, pages 427–438. ACM, 2000.

[125] Sarunas Raudys and Robert PW Duin. Expected classification error of the fisher linear classifier with pseudo-inverse covariance matrix. *Pattern Recognition Letters*, 19(5):385–392, 1998.

[126] P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing 1-penalized log-determinant divergence. *Electronic Journal of Statistics,*, 5:935–980, 2011.

[127] G.M. Reaven and R.G. Miller. An attemp to define nature of chemical diabest using a multidimensional analysis. *Diabetologica,*, 16:17–24, 1979.

[128] David M Rocke and David L Woodruff. Identification of outliers in multivariate data. *Journal of the American Statistical Association*, 91(435):1047–1061, 1996.

[129] P. J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association,*, 79(388):871–880, 1984.

[130] Peter J. Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41:212–223, 1998.

[131] P.J. Rousseeuw. Multivariate estimation with high breakdown point. In

W. Grossmann, G. Pflug, I. Vincze, and W. Wertz, editors, *In Mathematical Statistics and Applications,*, Dordrecht, 1985. Reidel Publishing Company.

[132] P.J. Rousseeuw and K. Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics,*, 41(3):212–223, 1999.

[133] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):211–229, 1959.

[134] Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

[135] Robert E Schapire. A brief introduction to boosting. In *Ijcai*, volume 99, pages 1401–1406, 1999.

[136] Robert E Schapire. Explaining adaboost. In *Empirical inference*, pages 37–52. Springer, 2013.

[137] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *Computational learning theory*, pages 416–426. Springer, 2001.

[138] Bernhard Schölkopf, John C. Platt, John Shawe-taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution, 1999.

[139] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[140] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Artificial Neural NetworksICANN'97*, pages 583–588. Springer, 1997.

[141] Marina Skurichina and Robert PW Duin. Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5(2):121–135, 2002.

[142] A. Stephenson, A.J.and Smith, M.W. Kattan, J. Satagopan, V.E. Reuter, P.T. Scardino, and W.L. Gerald. Integration of gene expression profiling and clinical variables to predict prostate carcinoma recurrence after radical prostatectomy. *Cancer,*, 104(2):290–298, 2005.

[143] W.B. Stern and J.-P. Descoeudres. X-ray fluorescence analysis of archaic greek pottery. *Archaeometry,*, 19:73–86, 1977.

[144] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[145] Andrey Nikolayevich Tikhonov. On the stability of inverse problems. In *Doklady Akademii Nauk SSSR*, volume 39, pages 195–198, 1943.

[146] V. Todorov and P. Filzmoser. An object-oriented framework for robust multivariate analysis. *Journal of Statistical Software,*, 32(3):1–47, 2009.

[147] V. Todorov and P. Filzmoser. Software tools for robust analysis of high-dimensional data. *Austrian Journal of Statistics,*, 43(3-4):255–266, 2014.

[148] V. Todorov and A. M. Pires. Comparative performance of several robust linear discriminant analysis methods. *REVSTAT - Statistical Journal,*, 5(1):63–83, 2007.

[149] J. Tohka, A. Zijdenbos, and A. Evans. Fast and robust parameter estimation for statistical partial volume models in brain mri. *NeuroImage,*, 23:84–97, 2004.

[150] Kagan Tumer and Joydeep Ghosh. Classifier combining: Analytical results

and implications. In *In Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, pages 126–132. AAAI Press, 1995.

[151] Kagan Tumer and Nikunj C. Oza. Decimated input ensembles for improved generalization, 1999.

[152] Michiel van Wezel and Rob Potharst. Improved customer choice predictions using ensemble methods. *European Journal of Operational Research*, 181(1):436 – 452, 2007.

[153] K. Vanden Branden and M. Hubert. Robust classification in high dimensions based on the simca method. *Chemometrics and Intelligent Laboratory Systems,*, 79:10–21, 2005.

[154] V Vapnik and A Sterin. On structural risk minimization or overall risk in a problem of pattern recognition. *Automation and Remote Control*, 10(3):1495–1503, 1977.

[155] Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2013.

[156] Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of Complexity*, pages 11–30. Springer, 2015.

[157] S. Wold. Pattern recognition by means of disjoint principal components models. *Pattern Recognition,*, 8(3):127–139, 1976.

[158] David H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[159] Ming Yuan. High dimensional inverse covariance matrix estimation via linear

programming. *The Journal of Machine Learning Research*, 11:2261–2286, 2010.

[160] Ji Zhang, Meng Lou, Tok Wang Ling, and Hai Wang. Hos-miner: a system for detecting outlyting subspaces of high-dimensional data. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 1265–1268. VLDB Endowment, 2004.

[161] Rui Zhang, Shaoyan Zhang, Sethuraman Muthuraman, and Jianmin Jiang. One class support vector machine for anomaly detection in the communication network performance data. In *Proceedings of the 5th Conference on Applied Electromagnetics, Wireless and Optical Communications*, ELECTROSCIENCE'07, pages 31–37, Stevens Point, Wisconsin, USA, 2007. World Scientific and Engineering Academy and Society (WSEAS).

[162] Yulian Zhu, Jun Liu, and Songcan Chen. Semi-random subspace method for face recognition. *Image and Vision Computing*, 27(9):1358–1370, 2009.

[163] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. Outlier detection in high dimensional data. In *Tutorial at the 12th International Conference on Data Mining (ICDM), Brussels, Belgium*, volume 10, 2012.

[164] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.