APPLICATIONS OF A ROBUST DISPERSION ESTIMATOR

by

Jianfeng Zhang

Doctor of Philosophy in Mathematics, Southern Illinois University, 2011

A Research Dissertation
Submitted in Partial Fulfillment of the Requirements for the
Doctor of Philosophy Degree

Department of Mathematics
in the Graduate School
Southern Illinois University Carbondale
August, 2011

DISSERTATION APPROVAL

Applications of a Robust Dispersion Estimator

By

Jianfeng Zhang

A Dissertation Submitted in Partial

Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in the field of Mathematics Approved by:

Professor David Olive, Chair

Professor Sakthivel Jeyaratnam

Professor Randy Hughes

Professor Walter Wallis

Professor Yanyan Sheng

Graduate School
Southern Illinois University Carbondale
(2011)

AN ABSTRACT OF THE DISSERTATION OF

Jianfeng Zhang, for the Doctor of Philosophy degree in Mathematics, presented on August 10th, 2011, at Southern Illinois University Carbondale.

TITLE: APPLICATIONS OF A ROBUST DISPERSION MATRIX

MAJOR PROFESSOR: Professor David Olive

Robust estimators for multivariate location and dispersion should be $\sqrt{n}$ consistent and highly outlier resistant, but estimators that have been shown to have these properties are impractical to compute. The RMVN estimator is an easily computed outlier resistant robust $\sqrt{n}$ consistent estimator of multivariate location and dispersion, and the estimator is obtained by scaling the classical estimator applied to the "RMVN subset" that contains at least half of the cases. Several robust estimators will be presented, discussed and compared in detail. The applications for the RMVN estimator are numerous, and a simple method for performing robust principal component analysis (PCA), canonical correlation analysis (CCA) and factor analysis is to apply the classical method to the "RMVN subset." Two approaches for robust PCA and CCA will be introduced and compared by simulation studies.

DEDICATION

This dissertation is dedicated to my wife Ping Ye, my daughter Christina Yilei Zhang and my parents whose encouragement have meant to me so much during the pursuit of my graduate degree and the composition of the dissertation.

ACKNOWLEDGMENTS

I would like to thank Professor David Olive for his invaluable assistance and insights leading to the writing of this dissertation. My sincere thanks also goes to the five members of my graduate committee for their patience and understanding during the six years of effort that went into the production of this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

A *multivariate location and dispersion (MLD) model* is a joint distribution with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, where $\boldsymbol{\mu}$ is a $p \times 1$ population location vector and $\boldsymbol{\Sigma}$ is a $p \times p$ symmetric positive definite population dispersion (scatter) matrix. Estimating $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ forms the cornerstone of multivariate data analysis since the estimators are widely used by many classical multivariate methods. Suppose the observed data is $\boldsymbol{x}_i$, for $i = 1, \cdots, n$ on p variables collected in an $n \times p$ matrix $\boldsymbol{X}$.

$$\underset{(n\times p)}{\boldsymbol{X}} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_n^T \end{bmatrix}$$

The most commonly used estimators of multivariate location and dispersion are the classical estimator $(\bar{\boldsymbol{x}}, \boldsymbol{S})$ where $\bar{\boldsymbol{x}}$ is the sample mean and $\boldsymbol{S}$ is the sample covariance-variance matrix. Then

$$\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \quad \text{and} \quad \boldsymbol{S} = \frac{1}{\mathrm{n}-1}\sum_{\mathrm{i}=1}^{\mathrm{n}} (\boldsymbol{x}_{\mathrm{i}} - \bar{\boldsymbol{x}})(\boldsymbol{x}_{\mathrm{i}} - \bar{\boldsymbol{x}})'.$$

An important MLD model is the elliptically contoured $EC_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, g)$ distribution with probability density function.

$$f(\boldsymbol{x}) = k_p |\boldsymbol{\Sigma}|^{-1/2} g\left[(\boldsymbol{x} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right]$$

where g is some known function and k_p is some positive constant.

If random vector $\boldsymbol{X}$ has an elliptically contoured (EC) distribution, then the characteristic function of $\boldsymbol{X}$ is

$$\phi_{\boldsymbol{X}}(\boldsymbol{t}) = \mathrm{E}\{\exp[i\boldsymbol{t}'(\boldsymbol{X})]\} = \exp(i\boldsymbol{t}'\boldsymbol{\mu})\psi(\boldsymbol{t}'\boldsymbol{\Sigma}\boldsymbol{t})$$

for some function ψ. See Johnson (1987, p. 107). If the second moments exist, then

$$E(\boldsymbol{X}) = \boldsymbol{\mu}$$

and

$$\mathrm{Cov}(\boldsymbol{X}) = c_X\boldsymbol{\Sigma}$$

where

$$c_X = -2\psi'(0).$$

A p-dimensional multivariate normal (MVN) $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ distribution has a probability density function

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-(\boldsymbol{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})/2}. \tag{1.1}$$

So $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is just a special case of $EC_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, g)$ with

$$g(u) = e^{-u} \quad \text{and} \quad k_p = \frac{1}{(2\pi)^{p/2}}.$$

The classical estimator $(\bar{\boldsymbol{x}}, \boldsymbol{S})$ plays an important role in multivariate analysis. If $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$ are a random sample of size n from a multivariate normal population, then $(\bar{\boldsymbol{x}}, \frac{n-1}{n}\boldsymbol{S})$ is the MLE of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\bar{\boldsymbol{x}}$ and $\boldsymbol{S}$ are sufficient statistics, and $\bar{\boldsymbol{x}}$ and $\boldsymbol{S}$ are independent. The widely used Hotelling's T^2, which is in honor of Harold

Hotelling, a pioneer in multivariate analysis, is defined as

$$T^2 = n(\bar{\boldsymbol{x}} - \boldsymbol{\mu})'\boldsymbol{S}^{-1}(\bar{\boldsymbol{x}} - \boldsymbol{\mu})$$

Hotelling first obtained that

$$T^2 \sim \frac{(n-1)p}{n-p} F_{p,n-p}$$

where $F_{p,n-p}$ is the F distribution with parameters p and $n-p$. A $100(1-\alpha)\%$ confidence region for the mean $\boldsymbol{\mu}$ of a p-dimensional MVN is the ellipsoid determined by all $\boldsymbol{\mu}$ such that

$$T^2 = n(\bar{\boldsymbol{x}} - \boldsymbol{\mu})'\boldsymbol{S}^{-1}(\bar{\boldsymbol{x}} - \boldsymbol{\mu}) \leq \frac{(n-1)p}{n-p} F_{p,n-p}(1-\alpha)$$

It can be shown that

$$\frac{n}{n+1}(\boldsymbol{x} - \bar{\boldsymbol{x}})'\boldsymbol{S}^{-1}(\boldsymbol{x} - \bar{\boldsymbol{x}}) \text{ is distributed as } \frac{(n-1)p}{n-p} F_{p,n-p}$$

and a $100(1-\alpha)\%$ p-dimensional prediction ellipsoid is given by all $\boldsymbol{x}$ satisfying

$$(\boldsymbol{x} - \bar{\boldsymbol{x}})'\boldsymbol{S}^{-1}(\boldsymbol{x} - \bar{\boldsymbol{x}}) \leq \frac{(n^2-1)p}{n(n-p)} F_{p,n-p}(1-\alpha)$$

where $P\big(F_{p,n-p} \leq F_{p,n-p}(1-\alpha)\big) = 1-\alpha$. The above prediction region for a future observed value $\boldsymbol{x}_f$ is an ellipsoid that is centered at the initial sample mean $\bar{\boldsymbol{x}}$, and its axes are determined by the eigenvectors of $\boldsymbol{S}$. Before any new observations are taken, the probability that $\boldsymbol{x}_f$ falls in the prediction ellipsoid is $1-\alpha$.

Let the $p \times 1$ column vector $T = T(\boldsymbol{X})$ be a multivariate location estimator, and let the $p \times p$ symmetric positive definite matrix $\boldsymbol{C} = \boldsymbol{C}(\boldsymbol{X})$ be a dispersion estimator. The *squared sample Mahalanobis distance* is the scalar

$$D_i^2(T, \boldsymbol{C}) = D^2(\boldsymbol{x}_i, T, \boldsymbol{C}) = (\boldsymbol{x}_i - T)'\boldsymbol{C}^{-1}(\boldsymbol{x}_i - T) \tag{1.2}$$

for each observation $\boldsymbol{x}_i$. Notice that the Euclidean distance of $\boldsymbol{x}_i$ from the estimate of center $T(\boldsymbol{X})$ is $D(\boldsymbol{x}_i, T, \boldsymbol{I}_p)$. The classical Mahalanobis distance uses $(T, \boldsymbol{C}) = (\overline{\boldsymbol{x}}, \boldsymbol{S})$. The population squared Mahalanobis distance is

$$U \equiv D^2 = D^2(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (\boldsymbol{x} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}). \tag{1.3}$$

For EC distributions, U has density

$$h(u) = \frac{\pi^{p/2}}{\Gamma(p/2)} k_p u^{p/2-1} g(u).$$

See Johnson (1987, p. 107-108).

A *principal component analysis* (PCA) is often conducted on the sample covariance matrix or on the sample correlation matrix. The objective is to construct uncorrelated linear combinations of the measured variables that account for much of the variation in the sample. The uncorrelated combinations with the largest variances are called the sample principal components. Suppose the $p \times p$ sample covariance matrix has eigenvalue-eigenvector pairs $(\hat{\lambda}_1, \hat{\boldsymbol{e}}_1)$, $(\hat{\lambda}_2, \hat{\boldsymbol{e}}_2)$, $\cdots$, $(\hat{\lambda}_p, \hat{\boldsymbol{e}}_p)$.

Also rewrite the $n \times p$ observation data matrix $\boldsymbol{X}$ as

$$\underset{(n\times p)}{\boldsymbol{X}} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix} = \begin{bmatrix} X_1 & X_2 & \cdots & X_p \end{bmatrix}$$

where $X_i = \begin{bmatrix} X_{1i} & X_{2i} & \cdots & X_{ni} \end{bmatrix}'$. Then the ith sample principal component is given by

$$\hat{Y}_i = \hat{\boldsymbol{e}}_i'\boldsymbol{X} = \hat{e}_{i1}X_1 + \hat{e}_{i2}X_2 + \cdots + \hat{e}_{ip}X_p$$

where $1 \leq i \leq p$ and $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_p \geq 0$. If the underlying distribution of $\boldsymbol{X}$ is MVN, then the population principal components $Y_i = \boldsymbol{e}_i'(\boldsymbol{X} - \boldsymbol{\mu})$ have an $N_p(\boldsymbol{0}, \boldsymbol{\Lambda})$ distribution, where the diagonal matrix $\boldsymbol{\Lambda}$ has entries $\lambda_1, \lambda_2, \cdots, \lambda_p$. The contour consisting of all $p \times 1$ vectors $\boldsymbol{x}$ satisfying

$$(\boldsymbol{x} - \bar{\boldsymbol{x}})'\boldsymbol{S}^{-1}(\boldsymbol{x} - \bar{\boldsymbol{x}}) = c^2 \tag{1.4}$$

estimates the constant density contour $(\boldsymbol{x} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}) = c^2$. Geometrically, the data may be plotted as n points in p dimensional space. Thus (1.4) defines a hyperellipsoid that is centered at $\bar{\boldsymbol{x}}$ and has axes given by the eigenvectors of $\boldsymbol{S}$. The lengths of these hyperellipsoid axes are proportional to $\sqrt{\lambda_i}$, $i = 1, 2, \cdots, p$.

Despite their important role in multivariate analysis, the classical estimators have a major flaw of being extremely sensitive to the presence of outliers. In consequence, the classical multivariate procedures based on classical estimators are greatly influenced by outliers. Therefore, it is important to consider robust alternative estimators to these estimators. Roughly speaking, a robust statistic (Huber 1981) is resistant to errors in the results produced by small deviations from assumptions (e.g. of normality). This means that if the assumptions are only approximately met, the estimator will still have reasonable efficiency and reasonably small bias. This dissertation is focused on applications of the recently developed RMVN robust estimator of Olive and Hawkins (2010).

One of the measures of robustness is the *breakdown value.* Robust estimators are expected to be bounded despite the presence of distorting outliers. Suppose d of the cases have been replaced by arbitrarily bad contaminated cases, then the rate of contamination is $\gamma = d/n$. For the multivariate location estimator $T(\boldsymbol{X})$, the *breakdown point value* is the smallest value of γ that makes $||T(\boldsymbol{X})||$ arbitrarily large. How to define the breakdown value of the dispersion estimator $\boldsymbol{C}(\boldsymbol{X})$ is a bit more complicated. In linear algebra,

$$\max_{||\boldsymbol{x}||=1} \boldsymbol{x}'\boldsymbol{C}\boldsymbol{x} = \lambda_1 \quad \text{(equality attained when } \boldsymbol{x} = \boldsymbol{e}_1)$$
$$\min_{||\boldsymbol{x}||=1} \boldsymbol{x}'\boldsymbol{C}\boldsymbol{x} = \lambda_p \quad \text{(equality attained when } \boldsymbol{x} = \boldsymbol{e}_p)$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$ are eigenvalues of $\boldsymbol{C}(\boldsymbol{X})$ and $\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_p$ are associated normalized eigenvectors. The spectral decomposition of the positive definite matrix $\boldsymbol{C}(\boldsymbol{X})$ is given by

$$\boldsymbol{C}(\boldsymbol{X}) = \sum_{i=1}^{p} \lambda_i \boldsymbol{e}_i \boldsymbol{e}_i'.$$

For $\|\boldsymbol{e}_i\| = 1$ for all i, the breakdown value of $\boldsymbol{C}(\boldsymbol{X})$ is only determined by eigenvalues. Hence for the dispersion estimator $\boldsymbol{C}(\boldsymbol{X})$, the *breakdown value* is the smallest value of γ needed to drive the largest eigenvalue of $\boldsymbol{C}(\boldsymbol{X})$ to ∞ or to drive the smallest eigenvalue to 0. *High Breakdown* (HB) statistics have $\gamma \to 0.5$ as $n \to \infty$. The sample median is a simple HB location estimator.

Besides high breakdown, consistency (stability), statistical efficiency and computational efficiency are factors to be scrutinized for any robust statistical procedure. A sequence of estimators $\boldsymbol{W}_n = \boldsymbol{W}_n(\boldsymbol{X})$ is called a *consistent sequence of estimators* of the parameter $\boldsymbol{\theta}$ if for every $\epsilon > 0$,

$$\lim_{n \to \infty} \mathrm{P}(||\boldsymbol{W}_n - \boldsymbol{\theta}|| < \epsilon) = 1.$$

Or equivalently,

$$\lim_{n \to \infty} \mathrm{P}(||\boldsymbol{W}_n - \boldsymbol{\theta}|| > \epsilon) = 0.$$

That is, $\boldsymbol{W}_n$ converges to $\boldsymbol{\theta}$ in probability. In mathematical notation,

$$\boldsymbol{W}_n(\boldsymbol{X}) \xrightarrow{P} \boldsymbol{\theta}.$$

As the sample size becomes infinite, the consistent estimator will be arbitrarily close to the parameter with high probability. Two results relevant to consistency are worthy to be mentioned here.

Theorem 1. *If W_n is a sequence of estimators of a scalar parameter θ satisfying*

i) $\lim_{n \to \infty} \mathrm{Var}_\theta \, W_n = 0$*,*

ii) $\lim_{n \to \infty} \mathrm{Bias}_\theta \, W_n = 0$*,*

then W_n is a consistent sequence of estimators of θ.

Proof.

$$\mathrm{E}_\theta[(W_n - \theta)^2] = \mathrm{Var}_\theta \, W_n + [\mathrm{Bias}_\theta \, W_n]^2.$$

By Chebychev's Inequality,

$$\mathrm{P}(|W_n - \theta| \geq \epsilon) \leq \frac{\mathrm{E}_\theta[(W_n - \theta)^2]}{\epsilon^2}.$$

Hence

$$\mathrm{P}(|W_n - \theta| \geq \epsilon) \to 0.$$

$$\square$$

Another result should be mentioned is the consistency of *Maximum Likelihood Estimator (MLE).* Let $X_1, X_2, \cdots, X_n$ be iid $f(x|\theta)$. Let $\tau(\theta)$ be a continuous function of θ. Then under regularity conditions $\tau(\hat{\theta})$ is a consistent estimator of $\tau(\theta)$. See Stuart, Ord, and Arnold (1999).

The property of consistency is concerned with the fact whether the estimator converges to the parameter that is estimated. The statistical efficiency is concerned with asymptotic variance of an estimator. A sequence of estimators W_n is *asymptotically efficient* for a parameter $\tau(\theta)$ if $\sqrt{n}[W_n - \tau(\theta)] \xrightarrow{D} N(0, v(\theta))$ and

$$v(\theta) = \frac{[\tau'(\theta)]^2}{\mathrm{E}_\theta\left(\frac{\partial}{\partial\theta}\log f(X|\theta)^2\right)};$$

that is the asymptotic variance of W_n achieves the Cramér-Rao Lower Bound. See Casella and Berger (2002, p. 471).

In general, MLEs are considered to be asymptotically efficient. So the asymptotic variance of MLE is intuitively used to define the asymptotic efficiency of an estimator. Suppose the estimator W has asymptotic variance v and the MLE has asymptotic covariance v_0. Then the asymptotic efficiency of W is defined by

$$\mathrm{Eff}(W) = \frac{v_0}{v}.$$

The asymptotic efficiency of multidimensional estimators is defined in a similar manner. Let $\boldsymbol{W}_0$ be the MLE of parameter vector $\boldsymbol{\theta}$ with asymptotic covariance matrix $\boldsymbol{V}_0$ and let $\boldsymbol{W}_n$ be an estimator of $\boldsymbol{\theta}$ with asymptotic covariance matrix $\boldsymbol{V}$.

The asymptotic efficiency of $\boldsymbol{W}$ is defined by

$$\mathrm{Eff}(\boldsymbol{W}_n) = \min_{\boldsymbol{c} \neq \boldsymbol{0}} \frac{\boldsymbol{c}'\boldsymbol{V}_0\boldsymbol{c}}{\boldsymbol{c}'\boldsymbol{V}\boldsymbol{c}}.$$

It follows that $\mathrm{Eff}(\boldsymbol{W}_n)$ is equal to the largest eigenvalue of the matrix $\boldsymbol{V}^{-1}\boldsymbol{V}_0$.

The last concept to be introduced in this section is *affine equivariance.*

Definition. Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n]'$ be a data matrix. The multivariate location and dispersion estimator $(T, \boldsymbol{C})$ is *affine equivariant* if for any linear transformation $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{A} + \boldsymbol{B}$,

$$T(\boldsymbol{Z}) = T(\boldsymbol{X}\boldsymbol{A} + \boldsymbol{B}) = \boldsymbol{A}'T(\boldsymbol{X}) + \boldsymbol{b},$$

and

$$\boldsymbol{C}(\boldsymbol{Z}) = \boldsymbol{C}(\boldsymbol{X}\boldsymbol{A} + \boldsymbol{B}) = \boldsymbol{A}'\boldsymbol{C}(\boldsymbol{X})\boldsymbol{A},$$

where $\boldsymbol{A}$ is a constant matrix, $\boldsymbol{b}$ is a constant vector, and $\boldsymbol{B} = \boldsymbol{1}\boldsymbol{b}'$.

According to this definition, it is easy to see that the classical sample mean and sample covariance matrix are affine equivariant. Affine equivariance is naturally desirable because it makes the analysis independent of the measurement scales of variables as well as the translation of the data. Under any nonsingular linear transformations, i.e., canonical correlations analysis, the result remains essentially unchanged. Suppose $\boldsymbol{x}$ has an elliptically contoured distribution, $EC_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, g)$. Denote $(T_\infty(\boldsymbol{x}), C_\infty(\boldsymbol{x}))$ as the asymptotic values of the affine equivariant estimators of $\boldsymbol{\mu}$ and $c\boldsymbol{\Sigma}$. Then

$$T_\infty(\boldsymbol{x}) = \boldsymbol{\mu}, \quad C_\infty(\boldsymbol{x}) = c\boldsymbol{\Sigma},$$

where c is a constant. See Maronna, Martin and Yohai (2006, p. 217).

If $(T, \boldsymbol{C})$ is affine equivariant, so is $(T, D^2_{(c_n)}(T, \boldsymbol{C})\ \boldsymbol{C})$ where $D^2_{(j)}(T, \boldsymbol{C})$ is the jth order statistic of the D_i^2. The following proposition shows that the Mahalanobis distances are invariant under affine transformations. See Rousseeuw and Leroy (1987, p. 252-262) for similar results.

Proposition 1. *If* $(T, \boldsymbol{C})$ *is affine equivariant, then*

$$D_i^2(\boldsymbol{W}) \equiv D_i^2(T(\boldsymbol{W}), \boldsymbol{C}(\boldsymbol{W})) =$$

$$D_i^2(T(\boldsymbol{Z}), \boldsymbol{C}(\boldsymbol{Z})) \equiv D_i^2(\boldsymbol{Z}). \tag{1.5}$$

Proof. Since $\boldsymbol{Z} = \boldsymbol{W}\boldsymbol{A} + \boldsymbol{B}$ has ith row

$$\boldsymbol{z}_i' = \boldsymbol{x}_i'\boldsymbol{A} + \boldsymbol{b}',$$

$$D_i^2(\boldsymbol{Z}) = [\boldsymbol{z}_i - T(\boldsymbol{Z})]'\boldsymbol{C}^{-1}(\boldsymbol{Z})[\boldsymbol{z}_i - T(\boldsymbol{Z})]$$

$$= [\boldsymbol{A}'(\boldsymbol{x}_i - T(\boldsymbol{W}))]'[\boldsymbol{A}'\boldsymbol{C}(\boldsymbol{W})\boldsymbol{A}]^{-1}[\boldsymbol{A}'(\boldsymbol{x}_i - T(\boldsymbol{W}))]$$

$$= [\boldsymbol{x}_i - T(\boldsymbol{W})]'\boldsymbol{C}^{-1}(\boldsymbol{W})[\boldsymbol{x}_i - T(\boldsymbol{W})] = D_i^2(\boldsymbol{W}).$$

□

The RMVN estimator given in Chapter 2 is $\sqrt{n}$ consistent and will be used to detect outliers and to replace the classical estimator in methods such as the Hotelling's T-test, Principal Component Analysis and Canonical Correlation Analysis. This dissertation compares multivariate statistical methods that use the classical estimator with methods that use the recently developed FCH estimator, *Fast Minimum Covariance Determinant* (FMCD) estimator, RMVN estimator and OGK estimator. These estimators are defined in the next chapter.

CHAPTER 2

MLD ROBUST ESTIMATORS COMPARISON

This chapter introduces and compares some well-known robust estimators of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, such as the Huber M-estimator, *minimum volume ellipsoid* (MVE) estimator, the *minimum covariance determinant* (MCD) estimator, the *fast minimum covariance determinant* (FMCD) estimator, the FCH estimator, the RMVN estimator and so on. The work of this chapter closely follows Olive (2004), Olive (2008), and Olive and Hawkins (2010).

2.1 HUBER M-ESTIMATOR

M-estimators are defined by generalizing MLEs. By the density function of multivariate normal distribution given in (1.1), the normal density function $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ can be rewritten as

$$f(\boldsymbol{x}) = \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \rho\big(d(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})\big),$$

where $\rho(t) = (2\pi)^{-p/2} \exp(-t/2)$ and $d(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (\boldsymbol{x} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})$. Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$ be an iid sample with a $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ distribution. Then the likelihood function is

$$L(\mu, \boldsymbol{\Sigma}) = \frac{1}{|\boldsymbol{\Sigma}|^{n/2}} \prod_{i=1}^{n} \rho\big(d(\boldsymbol{x}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma})\big)$$

and

$$-2 \log L(\mu, \boldsymbol{\Sigma}) = n \log |\boldsymbol{\Sigma}| + \sum_{i=1}^{n} \rho(d_i) \tag{2.1}$$

where $d_i = d(\boldsymbol{x}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Maximizing $L(\mu, \boldsymbol{\Sigma})$ now becomes minimizing (2.1). Differentiating the right hand side of (2.1) with respect to $\boldsymbol{\mu}$ and setting the derivative equal 0, one obtains

$$0 + \sum_{i=1}^{n} \rho'(d_i) \frac{\partial d_i}{\partial \boldsymbol{\mu}} = 0. \tag{2.2}$$

That is

$$\sum_{i=1}^{n} \rho'(d_i) \frac{\partial (\boldsymbol{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = 0. \tag{2.3}$$

Recall in linear algebra,

$$\frac{\partial \boldsymbol{b}'\boldsymbol{A}\boldsymbol{b}}{\partial \boldsymbol{b}} = (\boldsymbol{A}+\boldsymbol{A}')\boldsymbol{b}, \tag{2.4}$$

$$\frac{\partial \boldsymbol{b}'\boldsymbol{A}\boldsymbol{b}}{\partial \boldsymbol{A}} = \boldsymbol{b}\boldsymbol{b}', \tag{2.5}$$

and

$$\frac{\partial |\boldsymbol{A}|}{\partial \boldsymbol{A}} = |\boldsymbol{A}|\boldsymbol{A}^{-1}. \tag{2.6}$$

Using (2.4), (2.3) becomes

$$\sum_{i=1}^{n} \rho'(d_i)(\boldsymbol{x}-\boldsymbol{\mu}) = 0. \tag{2.7}$$

Now using (2.5) and (2.6) to differentiate right hand side of (2.1) with respect to $\boldsymbol{\Sigma}$ and setting the derivative equal 0, one obtains

$$n\frac{1}{|\boldsymbol{\Sigma}|}(|\boldsymbol{\Sigma}| \cdot \boldsymbol{\Sigma}^{-1}) + \sum_{i=1}^{n} \rho'(d_i)(\boldsymbol{x}_i-\boldsymbol{\mu})(\boldsymbol{x}_i-\boldsymbol{\mu})' = 0.$$

That is,

$$\frac{1}{n}\sum_{i=1}^{n} \rho'(d_i)(\boldsymbol{x}_i-\boldsymbol{\mu})(\boldsymbol{x}_i-\boldsymbol{\mu})' = \boldsymbol{\Sigma}. \tag{2.8}$$

Equations (2.7) and (2.8) together form the system of estimating equations for the MLE, $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$.

$$\sum_{i=1}^{n} W(d_i)(\boldsymbol{x}-\hat{\boldsymbol{\mu}}) = 0 \tag{2.9}$$

$$\frac{1}{n}\sum_{i=1}^{n} W(d_i)(\boldsymbol{x}_i-\hat{\boldsymbol{\mu}})(\boldsymbol{x}_i-\hat{\boldsymbol{\mu}})' = \hat{\boldsymbol{\Sigma}} \tag{2.10}$$

where $W = \rho'$ is called a weight function. Note that for the normal distribution, $W = \rho' \equiv 1$, and the solutions of the system of equations above are actually the sample mean and sample covariance matrix, the MLE of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

The M-estimator is generalized to be solutions of

$$\sum_{i=1}^{n} W_1(d_i)(\boldsymbol{x} - \hat{\boldsymbol{\mu}}) = 0 \tag{2.11}$$

$$\frac{1}{n}\sum_{i=1}^{n} W_2(d_i)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})' = \hat{\boldsymbol{\Sigma}} \tag{2.12}$$

where W_1 and W_2 are not necessarily equal. From (2.12), one can intuitively look at $\hat{\boldsymbol{\Sigma}}$ as a weighted covariance matrix. Note that (2.12) actually does not give an explicit expression for $\hat{\boldsymbol{\Sigma}}$ since $W_2(d_i)$ depends on $\hat{\boldsymbol{\Sigma}}$. However, (2.12) suggests an iterative procedure to find the solution. From (2.11), $\hat{\boldsymbol{\mu}}$ can be expressed as the weighted mean

$$\hat{\boldsymbol{\mu}} = \frac{\sum_{i=1}^{n} W_1(d_i)\boldsymbol{x}_i}{\sum_{i=1}^{n} W_1(d_i)}. \tag{2.13}$$

Again, one should be aware that (2.13) is not an explicit expression for $\hat{\boldsymbol{\mu}}$ either since $W_1(d_i)$ depends on $\hat{\boldsymbol{\mu}}$. An iterative procedure could be adopted to compute $\hat{\boldsymbol{\mu}}$.

The *Huber function* is a very popular choice for weight functions. It is given by

$$\rho(x) = \begin{cases} x^2 & \text{if } |x| \le k \\ 2k|x| - k^2 & \text{if } |x| > k \end{cases}$$

with derivative $2\psi(x)$ where

$$\psi(x) = \begin{cases} x & \text{if } |x| \le k \\ \text{sgn}(x)k & \text{if } |x| > k \end{cases}$$

where $\operatorname{sgn}(x)$ gives the sign of the value of x. The weight function is defined by

$$W(x) = \begin{cases} \psi(x)/x & \text{if } |x| \neq 0 \\ \psi'(0) & \text{if } |x| = 0. \end{cases}$$

So the the weight function corresponding to Huber's ψ is

$$W(x) = \begin{cases} 1 & \text{if } |x| \leq k \\ \dfrac{k}{|x|} & \text{if } |x| > k. \end{cases}$$

Another popular choice is the *bisquare* function:

$$\rho(x) = \begin{cases} 1 - [1-(x/k)^2]^3 & \text{if } |x| \leq k \\ 1 & \text{if } |x| > k \end{cases}$$

which works better for heavy tail distributions, such as the Cauchy distribution. See Maronna, Martin, and Yohai (2006, p. 29). It has been proved that the M-estimators are affine equivariant, and the M-estimator is consistent if $\boldsymbol{x}$ has an elliptical distribution. Unfortunately the breakdown point of the M-estimator has been shown no more than $1/(p+1)$. So when the dimension of $\boldsymbol{x}$ gets larger, the M-estimator performance gets worse.

2.2 THE MVE AND THE MCD ESTIMATORS

For a data set X, let $\boldsymbol{D}(X, T, \boldsymbol{C})$ be a vector of $D(\boldsymbol{x}_i, T, \boldsymbol{C})$, $i = 1, \cdots, n$. One way to define the estimator $(T, \boldsymbol{C})$ is by

$$\hat{\sigma}\big(\boldsymbol{D}(X, T, \boldsymbol{C})\big) = \min, \qquad |\boldsymbol{C}| = 1 \tag{2.14}$$

where $\hat{\sigma}$ is a robust scale. The constraint $\boldsymbol{C} = 1$ is used to prevent the case that the small eigenvalues (close to 0) trivially cause small distances $D(\boldsymbol{x}, T, \boldsymbol{C})$. If $\hat{\sigma}$ is chosen to be sample median, then $(T, \boldsymbol{C})$ is called the *minimum volume ellipsoid* (MVE) estimator. Among all ellipsoids $\{\boldsymbol{x} : D^2(\boldsymbol{x}, \mu, \boldsymbol{\Sigma}) \leq h^2\}$ containing at least half of the data points, the one given by MVE estimate has the minimum volume. The equation

$$D^2(\boldsymbol{x}, T, \boldsymbol{C}) = (\boldsymbol{x} - T)'\boldsymbol{C}^{-1}(\boldsymbol{x} - T) = h^2$$

defines an ellipsoid centered at T. The volume of the hyperellipsoid $\{\boldsymbol{x} : D^2(\boldsymbol{x}, T, \boldsymbol{C}) \leq h^2\}$ is equal to

$$k_p |\boldsymbol{C}|^{1/2} h^p \tag{2.15}$$

where $k_p = \dfrac{2\pi^{p/2}}{p\Gamma(p/2)}$. See Johnson and Wichern (1998, p. 132). The complexity of computing MVE is very high: no feasible approach has been found to compute MVE when n and p are not small.

If the robust scale is chosen to be a trimmed scale

$$\hat{\sigma} = \sum_{i=1}^{m} D_{(i)},$$

where $D_{(i)}$ is ith order statistic of $D_i = D(\boldsymbol{x}_i, T, \boldsymbol{C})$ and $1 \leq m \leq n$, then $(T, \boldsymbol{C})$ defined by (2.14) is called a *minimum covariance determinant* (MCD) estimator. For each hyperellipsoid $\{\boldsymbol{x} : D^2(\boldsymbol{x}, \hat{\mu}, \hat{\boldsymbol{\Sigma}}) \leq h^2\}$ containing at least m data points, compute the classical covariance matrix of the data points in the hyperellipsoid. Given the MCD estimator $(T, \boldsymbol{C})$, the determinant computed from the sample variance matrix of hyperellipsoid $\{\boldsymbol{x} : D^2(\boldsymbol{x}, T, \boldsymbol{C}) \leq h^2\}$ is a minimum. See Maronna, Martin and Yohai (2006, section 6.4).

Butler et al. (1993) and Cator and Lopuhaä (2009) showed the MCD estimator is consistent, asymptotically normal, and affine equivariant. Although the

MCD estimator was introduced as early as 1984, it had not been practically used due to its enormous computation complexity. The fastest estimator of the multivariate location and dispersion that has been shown to be both consistent and high breakdown is the MCD estimator with complexity $O(n^{\delta})$ where $\delta = 1 + p(p+3)/2$. See Bernholt and Fischer (2004). The Fast-MCD algorithm of Rousseeuw and Van Driessen (1999) does not compute MCD, and it will be discussed in the following section.

2.3 CONCENTRATION ALGORITHM

One of many practical techniques for computing robust estimators of multivariate location and dispersion is the *concentration* technique. In general, concentration begins with some initial estimators of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, which are called starts, such as the classical estimator $(\bar{\boldsymbol{x}}, \boldsymbol{S})$ of some random subset with $p+1$ cases or the classical estimator $(\bar{\boldsymbol{x}}, \boldsymbol{S})$ computed from all n cases. For each start, a concentration algorithm generates a corresponding new estimator, which is called an attractor. Then one of the attractors is chosen to be used in the final "robust estimator" based on some criterion.

Suppose that there are K starts used for a concentration. Let $(T_{0,j}, \boldsymbol{C}_{0,j})$ be the jth start, where $1 \leq j \leq K$. Compute the squared Mahalanobis distances of n observations

$$D_i^2(T_{0,j}, \boldsymbol{C}_{0,j}) = (\boldsymbol{x}_i - T_{0,j}(\boldsymbol{X}))^T \boldsymbol{C}_{0,j}^{-1}(\boldsymbol{X})(\boldsymbol{x}_i - T_{0,j}(\boldsymbol{X})) \tag{2.16}$$

where $1 \leq i \leq n$. At the next iteration, the classical estimator $(T_{1,j}, \boldsymbol{C}_{1,j})$ is computed from the $c_n \approx n/2$ cases corresponding to the smallest Mahalabonis distances computed by (2.16). By continuing this iteration k times, a sequence of estimators $(T_{0,j}, \boldsymbol{C}_{0,j}), (T_{1,j}, \boldsymbol{C}_{1,j}), ..., (T_{k,j}, \boldsymbol{C}_{k,j})$ is obtained corresponding to the jth start

$(T_{0,j}, \boldsymbol{C}_{0,j})$. The last estimator $(T_{k,j}, \boldsymbol{C}_{k,j})$ of this sequence is said to be jth attractor. An empirical choice for k is 10. Once all K attractors are obtained, the one optimizing the criterion will be used in the final "robust estimator".

2.4 THE FAST-MCD AND THE OGK ESTIMATORS

Rousseeuw and Van Driessen (1999) introduced the Fast-MCD algorithm to approximate the MCD estimator. The major part of the algorithm is the concentration step (C-step). Provided an initial location and dispersion estimator $(T_{old}, \boldsymbol{C}_{old})$,

1. Compute the squared Mahalanobis distances of all n cases

$$D_i^2(T_{old}, \boldsymbol{C}_{old}) = (\boldsymbol{x}_i - T_{old})' \boldsymbol{C}_{old}^{-1} (\boldsymbol{x}_i - T_{old}).$$

2. Construct a subset H by choosing h cases with smallest Mahalanobis distances. It is common to take $h \approx n/2$.

3. Compute the new location and dispersion estimator $(T_{new}, \boldsymbol{C}_{new})$ by

$$T_{new} = \frac{1}{h} \sum_{i \in H} \boldsymbol{x}_i \quad \text{and}$$

$$\boldsymbol{C}_{new} = \frac{1}{h} \sum_{i \in H} (\boldsymbol{x}_i - T_{new})(\boldsymbol{x}_i - T_{new})'.$$

The complete algorithm works as following.

First, generate K elemental subsets, i.e. $K = 500$. Each subset contains $p+1$ cases randomly drawn from the original dataset.

Second, compute the sample mean and sample covariance matrix for each elemental subset as initial estimators $(T_{0,j}, \boldsymbol{C}_{0,j})$. Make sure each $\boldsymbol{C}_{0,j}$ is non-singular. If it is singular, random data drawn from original data set are added to the associated subset till $\boldsymbol{C}_{0,j}$ becomes non-singular.

Third, apply C-step to $(T_{0,j}, \boldsymbol{C}_{0,j})$ obtaining $(T_{1,j}, \boldsymbol{C}_{1,j})$ for all $1 \leq j \leq K$. Apply C-step again to $(T_{1,j}, \boldsymbol{C}_{1,j})$ obtaining $(T_{2,j}, \boldsymbol{C}_{2,j})$ for all $1 \leq j \leq K$.

Fourth, choose only 10 estimators with the smallest determinant from all $(T_{2,j}, \boldsymbol{C}_{2,j})$ where $1 \leq j \leq K$. Apply C-steps further only for those 10 chosen estimators until convergence to get 10 attractors.

Fifth, choose the attractor (out of 10) with the smallest determinant. Denote the chosen attractor as $(T_A, \boldsymbol{C}_A)$.

Lastly, reweight the chosen attractor $(T_A, \boldsymbol{C}_A)$ to obtain Fast-MCD $(T_F, \boldsymbol{C}_F)$.

$$T_F = \sum_{i=1}^{n} w_i \boldsymbol{x}_i / \sum_{i=1}^{n} w_i$$

$$\boldsymbol{C}_F = d_n \Big(\sum_{i=1}^{n} w_i (\boldsymbol{x}_i - T_F)(\boldsymbol{x}_i - T_F)' \Big) / \sum_{i=1}^{n} w_i$$

where d_n is a correction factor so that $(T_F, \boldsymbol{C}_F)$ can be a better estimator of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ when the clean data have a multivariate normal distribution and w_i is a weight function defined by

$$w_i = \begin{cases} 1 & \text{if } D(\boldsymbol{x}_i, T_A, \boldsymbol{C}_A) \leq \sqrt{\chi^2_{p,.975}} \\ 0 & \text{otherwise} \end{cases}$$

On the third step, only two C-steps are applied to all K elemental subsets for the purpose of increasing the computational efficiency. Moreover, Rousseeuw and Van Driessen (1999) give a theorem that after each C-step is applied, $|\boldsymbol{C}_{new}| \leq |\boldsymbol{C}_{old}|$ with equality only if $\boldsymbol{C}_{new} = \boldsymbol{C}_{old}$. The theorem guarantees the convergence of the determinants to a local min, obtained by applying finite C-steps iteratively.

The Fast-MCD estimator program is available in R. After loading the MASS library, call the function cov.mcd.

Maronna and Zamar (2002) introduced the orthogonalized Gnanadesikan-Kettenring (OGK) estimator. The OGK estimator is based on a robust estimator $\hat{\sigma}_{jk}$ of covariance σ_{jk} proposed by Gnanadesikan and Kettenring (1972). Let X_j and X_k be a pair of two random variables, m() be the robust mean function, and $\sigma()$ be the robust standard deviation function. The Gnanadesikan and Kettenring robust estimator of covariance σ_{jk} is calculated as

$$\hat{\sigma}_{jk} = \frac{1}{4}\big(\sigma(X_j + X_k)^2 - \sigma(X_j - X_k)^2\big).$$

When $p > 2$, Gnanadesikan and Kettenring's robust estimator, $\boldsymbol{A}$, of $\boldsymbol{\Sigma}$ is

$$\boldsymbol{A} = \begin{bmatrix} \hat{\sigma}_{11} & \hat{\sigma}_{12} & \cdots & \hat{\sigma}_{1p} \\ \hat{\sigma}_{21} & \hat{\sigma}_{22} & \cdots & \hat{\sigma}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\sigma}_{n1} & \hat{\sigma}_{n2} & \cdots & \hat{\sigma}_{np} \end{bmatrix}.$$

Unfortunately $\boldsymbol{A}$ is not affine equivariant and can be non positive definite. Maronna and Zamar (2002) proposed an algorithm based on $\hat{\sigma}_{jk}$ but yielding a positive definite covariance matrix estimator. Let the data matrix

$$X = [X_1, X_2, \cdots, X_p] = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n]'.$$

1. Compute $\boldsymbol{D} = \text{diag}(\sigma(X_1), \sigma(X_2), \cdots, \sigma(X_p))$. Define $\boldsymbol{y}_i = \boldsymbol{D}^{-1}\boldsymbol{x}_i$, where $i = 1, 2, \cdots, n$.

2. Compute the Gnanadesikan and Kettenring correlation matrix $\boldsymbol{U} = (U_{ij})$ with

$$U_{jk} = \begin{cases} \frac{1}{4}\big(\sigma(X_j + X_k)^2 - \sigma(X_j - X_k)^2\big) & \text{if } j \neq k \\ 1 & \text{if } j = k \end{cases}$$

3. Decompose the matrix $\boldsymbol{U} = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}'$, where $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_p]$ is an orthogonal matrix of eigenvectors of $\boldsymbol{U}$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_p)$ is a diagonal matrix of eigenvalues.

4. Define $\boldsymbol{z}_i = \boldsymbol{E}'\boldsymbol{x}_i$ and construct the matrix $Z = [Z_1, Z_2, \cdots, Z_p]$, $\boldsymbol{\Gamma} = \text{diag}(\sigma(Z_1)^2, \sigma(Z_2)^2, \cdots, \sigma(Z_p)^2)$ and $\nu = \big(\text{m}(Z_1), \text{m}(Z_2), \cdots, \text{m}(Z_p)\big)$. Compute $\boldsymbol{A} = \boldsymbol{D}\boldsymbol{E}$. Then the location and dispersion estimator $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$ of $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ is computed as:

$$(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}) = (\boldsymbol{A}\nu, \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}')$$

5. Iterate the procedure by replacing $\boldsymbol{U}$ in step 2 by $\boldsymbol{E}\boldsymbol{\Gamma}\boldsymbol{E}'$ until convergence.

6. Transform converged $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$ back to get the estimator $(\hat{\boldsymbol{\mu}}_{OGK}, \hat{\boldsymbol{\Sigma}}_{OGK})$ by

$$(\hat{\boldsymbol{\mu}}_{OGK}, \hat{\boldsymbol{\Sigma}}_{OGK}) = (\boldsymbol{D}\hat{\boldsymbol{\mu}}, \boldsymbol{D}\hat{\boldsymbol{\Sigma}}\boldsymbol{D}').$$

7. Compute the OGK estimator $(T_{OGK}, \boldsymbol{C}_{OGK})$ by a reweighting process.

$$T_{OGK} = \sum_{i=1}^{n} w_i \boldsymbol{x}_i / \sum_{i}^{n} w_{i=1}$$

and

$$\boldsymbol{C}_{OGK} = \Big(\sum_{i=1}^{n} w_i (\boldsymbol{x}_i - T_{OGK})(\boldsymbol{x}_i - T_{OGK})'\Big) / \sum_{i=1}^{n} w_i$$

where w_i is a weight function defined by

$$w_i = \begin{cases} 1 & \text{if } D(\boldsymbol{x}_i, \hat{\boldsymbol{\mu}}_{OGK}, \hat{\boldsymbol{\Sigma}}_{OGK}) \leq c \\ 0 & \text{otherwise} \end{cases}$$

where $c = \sqrt{\chi^2_{p,.975}}\text{MED}(D(\boldsymbol{x}_i, \hat{\boldsymbol{\mu}}_{OGK}, \hat{\boldsymbol{\Sigma}}_{OGK}))/\sqrt{\chi^2_{p,.5}}$. Maronna and Zamar (2002) proposed weighted mean for m() and τ-scale of Yohai and Zamar (1988) for $\sigma()$ in the OGK algorithm. As the Fast-MCD algorithm, the OGK estimator uses reweighting process to improve the simulated statistical efficiency.

The Orthogonalized Gnanadesikan-Kettenring (OGK) MLD estimator is described in in Maronna and Zamar (2002). It can be implemented by the R function covOGK from the robustbase library as was the covMcd concentration algorithm.

2.5 THE ELEMENTAL, THE MB AND THE DGK ESTIMATORS

Three important starts will be discussed in this section. Hawkins and Olive (1999) and Rousseeuw and Van Driessen (1999) use elemental starts. The DGK (Devlin, Gnanadesikan, Kettenring 1975, 1981) estimator is generated by concentration that has only one start, the classical estimator $(\bar{\boldsymbol{x}}, \boldsymbol{S})$ computed from all n cases. The Olive (2004) median ball (MB) estimator is generated by concentration that uses $(T_{0,j}, \boldsymbol{C}_{0,j}) = (\text{MED}(\boldsymbol{X}), \boldsymbol{I}_p)$ as the only start, where MED($\boldsymbol{X}$) is coordinatewise median. So 50% of cases furthest in Euclidean distance from the sample median MED($\boldsymbol{X}$) are trimmed for computing the MB start.

For the algorithm using concentration with randomly selected elemental starts, $(T_{0,j}, \boldsymbol{C}_{0,j})$ is the classical estimator applied to a randomly selected "elemental set" of $p+1$ cases. Although this algorithm is computationally efficient, it has theoretical drawbacks.

Proposition 2 (Inconsistent and Zero Breakdown Elemental Estimator)**.** *Suppose K randomly selected elemental starts are used with k step concentration to produce the attractors for a data set with size n, then the resulting estimator is inconsistent and zero breakdown if K and k are fixed and free of n. See Olive and Hawkins (2010).*

Proof. Each elemental start $(T_{0,j}, \boldsymbol{C}_{0,j})$ is the classical estimator applied to a randomly selected subset of $p+1$ cases. So each elemental start is zero breakdown. Changing one case can make an elemental start breakdown. A breakdown start implies a breakdown attractor. Hence the breakdown value of final estimator is bounded by $K/n \to 0$ as $n \to \infty$.

Without loss of generality, assume $\boldsymbol{x}_i$ are iid (independent and identically distributed) random variables and $\boldsymbol{x}_i$ do not have point mass at $\boldsymbol{\mu}$. That is,

$$\mathrm{P}(\boldsymbol{x}_i = \boldsymbol{\mu}) < 1.$$

Let $\bar{\boldsymbol{x}}_{0,j}$ be the jth start: sample mean applied to $p+1$ cases. There exits $\epsilon > 0$ such that

$$\mathrm{P}(||\bar{\boldsymbol{x}}_{0,j} - \boldsymbol{\mu}|| > \epsilon) \equiv \delta_\epsilon > 0.$$

Thus

$$\begin{aligned} & \mathrm{P}(\min_j ||\bar{\boldsymbol{x}}_{0,j} - \boldsymbol{\mu}|| > \epsilon) \\ = \ & \mathrm{P}(\text{all } ||\bar{\boldsymbol{x}}_{0,j} - \boldsymbol{\mu}|| > \epsilon) \\ \to \ & \delta_\epsilon^K > 0 \quad \text{as } n \to \infty \end{aligned}$$

Therefore the start that minimizes $||\bar{\boldsymbol{x}}_{0,j} - \boldsymbol{\mu}||$ is inconsistent. The elemental concentration needs $K_n \to \infty$ as $n \to \infty$ to obtain a consistent estimator. □

Proposition 2 shows that concentration using elemental starts does not produce a high breakdown and $\sqrt{n}$ consistent robust estimator. Hubert, Rousseeuw and Van Aelst (2008) claim that "MCD" can be efficiently computed with the Fast-MCD (or FMCD) estimator. The claim is plainly false because the FMCD uses an elemental concentration algorithm. Whether it is consistent or not is unknown. MCD, on the other hand, is consistent by Theorem 3. Theorem 3 will be discussed later in this section.

Like Proposition 2, the following proposition shows theory of the algorithm estimator depends on the theory of attractors, not on the estimator corresponding to the criterion. One can see Olive and Hawkins (2010) appendix for the proof.

Proposition 3. *Suppose that $(T_j, \boldsymbol{C}_j)$ are K attractors of $(\boldsymbol{\mu}, a\boldsymbol{\Sigma})$ for some constant $a > 0$. Let $(T_A, \boldsymbol{C}_A)$ be the final estimator obtained by choosing one of the K attractors where K is fixed.*

(i) If all of the attractors $(T_j, \boldsymbol{C}_j)$ are consistent, then $(T_A, \boldsymbol{C}_A)$ is consistent.

(ii) If all of the attractors $(T_j, \boldsymbol{C}_j)$ are consistent with the same rate, e.g., n^δ where $0 < \delta \le .5$, then $(T_A, \boldsymbol{C}_A)$ is consistent with the same rate n^δ.

(iii) If all of the attractors $(T_j, \boldsymbol{C}_j)$ are high breakdown, then $(T_A, \boldsymbol{C}_A)$ is high breakdown.

The rest of this section shows that the MB estimator is high breakdown and the DGK and MCD estimators are $\sqrt{n}$ consistent.

Lemma 1. *If the classical estimator $(\overline{\boldsymbol{x}}_B, \boldsymbol{S}_B)$ is applied to c_n cases that are contained in some bounded region where $p + 1 \le c_n \le n$, then the maximum eigenvalue λ_1 of $\boldsymbol{S}_B$ is bounded. See Olive and Hawkins (2010).*

Proof. The largest eigenvalue of a $p \times p$ matrix $\boldsymbol{A}$ is bounded above by $p \max |a_{i,j}|$ where $a_{i,j}$ is the (i,j) entry of $\boldsymbol{A}$. See Datta (1995, p. 403). Denote the c_n cases by $\boldsymbol{z}_1, ..., \boldsymbol{z}_{c_n}$. Then the (i,j)th element $a_{i,j}$ of $\boldsymbol{A} = \boldsymbol{S}_B$ is

$$a_{i,j} = \frac{1}{c_n - 1} \sum_{m=1}^{c_n} (z_{i,m} - \overline{z}_m)(z_{j,m} - \overline{z}_j).$$

Hence the maximum eigenvalue λ_1 is bounded. □

Theorem 2. *Suppose $(T, \boldsymbol{C})$ is a high breakdown start estimator where $\boldsymbol{C}$ is symmetric and positive definite if the contamination proportion d_n/n is less than the breakdown value. Each concentration uses the $c_n \approx n/2$ cases corresponding to the smallest distances. Then the concentration attractor $(T_k, \boldsymbol{C}_k)$ is a high breakdown estimator provided that k is fixed. See Olive and Hawkins (2010).*

Proof. Following Leon (1986, p. 280), if $\boldsymbol{A}$ is a symmetric and positive definite matrix with eigenvalues $\tau_1 \geq \cdots \geq \tau_n$, then for any nonzero vector $\boldsymbol{x}$,

$$0 < ||\boldsymbol{x}||^2 \tau_n \leq \boldsymbol{x}' \boldsymbol{A} \boldsymbol{x} \leq ||\boldsymbol{x}||^2 \tau_1.$$

Let $\lambda_1 \geq \cdots \geq \lambda_n > 0$ be the eigenvalues of $\boldsymbol{C}$. Then $\dfrac{1}{\lambda_n} \geq \cdots \geq \dfrac{1}{\lambda_1} > 0$ are the eigenvalues of $\boldsymbol{C}^{-1}$. And

$$\frac{1}{\lambda_1} ||\boldsymbol{x} - T||^2 \leq (\boldsymbol{x} - T)' \boldsymbol{C}^{-1} (\boldsymbol{x} - T) \leq \frac{1}{\lambda_n} ||\boldsymbol{x} - T||^2. \tag{2.17}$$

Let $D^2_{(i)}$ denote the order statistics of the $D^2_i(T, \boldsymbol{C})$. Since $(T, \boldsymbol{C})$ is a high breakdown, then $1/\lambda_n$ and MED($||\boldsymbol{x} - T||^2$) are both bounded even for the number of outliers d_n near $n/2$. Therefore, $D^2_{(i)} < V$ for some constant V that only depends on the clean data but not on the outliers even if i and d_n are near $n/2$.

Following Johnson and Wichern (1998, p. 132), the boundary of the set $\{\boldsymbol{x}|(\boldsymbol{x}-T)^T\boldsymbol{C}^{-1}(\boldsymbol{x}-T) \leq h^2\} = \{\boldsymbol{x}|D^2_{\boldsymbol{x}} \leq h^2\}$ is a hyperellipsoid centered at T with axes of length $2h\sqrt{\lambda_i}$. Hence $\{\boldsymbol{x}|D^2_{\boldsymbol{x}} \leq D^2_{(c_n)}\}$ is contained in some ball about the origin of radius r where r does not depend on the number of outliers even for d_n near $n/2$. This is the set containing the cases used to compute $(T_1, \boldsymbol{C}_1)$. Since the set is bounded, T_1 is bounded and the largest eigenvalue $\lambda_{1,1}$ of $\boldsymbol{C}_1$ is bounded by Lemma 1. Since $0 < det(\boldsymbol{C}_{MCD}) \leq det(\boldsymbol{C}_0)$, the smallest eigenvalue $\lambda_{n,0}$ is bounded away from 0. Since these bounds do not depend on the outliers even for d_n near $n/2$, $(T_0, \boldsymbol{C}_0)$ is a high breakdown estimator. Now repeat the argument with $(T_0, \boldsymbol{C}_0)$ in place of $(T, \boldsymbol{C})$ and $(T_1, \boldsymbol{C}_1)$ in place of $(T_0, \boldsymbol{C}_0)$. Then $(T_1, \boldsymbol{C}_1)$ is high breakdown. Repeating the argument iteratively shows $(T_k, \boldsymbol{C}_k)$ is high breakdown. □

The MB estimator $(T_{k,M}, \boldsymbol{C}_{k,M})$ uses $(T_{0,j}, \boldsymbol{C}_{0,j}) = (\text{MED}(\boldsymbol{X}), \boldsymbol{I}_p)$ as the only start, which is high breakdown. Theorem 2 implies the MB estimator is also high breakdown.

Lopuhaä (1999) shows that if a start $(T, \boldsymbol{C})$ is a consistent estimator of $(\boldsymbol{\mu}, s\boldsymbol{\Sigma})$, then the attractor $(T_k, \boldsymbol{C}_k)$ is a consistent estimator of $(\boldsymbol{\mu}, a\boldsymbol{\Sigma})$ That is, if

$$(T, \boldsymbol{C}) \xrightarrow{P} (\boldsymbol{\mu}, s\boldsymbol{\Sigma}),$$

then

$$(T_k, \boldsymbol{C}_k) \xrightarrow{P} (\boldsymbol{\mu}, a\boldsymbol{\Sigma})$$

where $a, s > 0$ are some constants. The constant a depends on s, p, and the elliptically contoured distribution, but does not otherwise depend on the consistent start. The constant a also depends on the weight function $I(D^2_i(T, \boldsymbol{C}) \leq h^2)$ where h^2 is a positive constant and the indicator is 1 if $D^2_i(T, \boldsymbol{C}) \leq h^2$ and 0 otherwise.

Following Olive and Hawkins (2010), to see that the Lopuhaä (1999) theory extends to concentration where the weight function uses $h^2 = D^2_{(c_n)}(T, \boldsymbol{C})$, note that $(T, \tilde{\boldsymbol{C}}) \equiv (T, D^2_{(c_n)}(T, \boldsymbol{C})\ \boldsymbol{C})$ is a consistent estimator of $(\boldsymbol{\mu}, b\boldsymbol{\Sigma})$ where $b > 0$ is derived in (2.19), and weight function $I(D^2_i(T, \tilde{\boldsymbol{C}}) \leq 1)$ is equivalent to the concentration weight function $I(D^2_i(T, \boldsymbol{C}) \leq D^2_{(c_n)}(T, \boldsymbol{C}))$. If $(T, \boldsymbol{C})$ is a $\sqrt{n}$ consistent estimator of $(\mu, s\ \boldsymbol{\Sigma})$, then

$$\begin{aligned}
D^2(T, \boldsymbol{C}) &= (\boldsymbol{x} - T)^T \boldsymbol{C}^{-1} (\boldsymbol{x} - T) \\
&= (\boldsymbol{x} - \boldsymbol{\mu} + \boldsymbol{\mu} - T)^T [\boldsymbol{C}^{-1} - s^{-1}\boldsymbol{\Sigma}^{-1} + s^{-1}\boldsymbol{\Sigma}^{-1}](\boldsymbol{x} - \boldsymbol{\mu} + \boldsymbol{\mu} - T) \\
&= (\boldsymbol{x} - \boldsymbol{\mu})^T [s^{-1}\boldsymbol{\Sigma}^{-1}](\boldsymbol{x} - \boldsymbol{\mu}) \\
&\quad + (\boldsymbol{x} - T)^T [\boldsymbol{C}^{-1} - s^{-1}\boldsymbol{\Sigma}^{-1}](\boldsymbol{x} - T) \\
&\quad + (\boldsymbol{x} - \boldsymbol{\mu})^T [s^{-1}\boldsymbol{\Sigma}^{-1}](\boldsymbol{\mu} - T) \\
&\quad + (\boldsymbol{\mu} - T)^T [s^{-1}\boldsymbol{\Sigma}^{-1}](\boldsymbol{x} - \boldsymbol{\mu}) \\
&\quad + (\boldsymbol{\mu} - T)^T [s^{-1}\boldsymbol{\Sigma}^{-1}](\boldsymbol{\mu} - T) \\
&= s^{-1} D^2(\mu, \boldsymbol{\Sigma}) + O_P(n^{-1/2}). \qquad (2.18)
\end{aligned}$$

Thus the sample percentiles of $D^2_i(T, \boldsymbol{C})$ are consistent estimators of the percentiles of $s^{-1}D^2(\mu, \boldsymbol{\Sigma})$. Suppose $c_n/n \to \xi \in (0, 1)$ as $n \to \infty$, and let $D^2_\xi(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be the ξth percentile of the population squared distances. Then

$$D^2_{(c_n)}(T, \boldsymbol{C}) \xrightarrow{P} s^{-1} D^2_\xi(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

and

$$b\boldsymbol{\Sigma} = s^{-1} D^2_\xi(\boldsymbol{\mu}, \boldsymbol{\Sigma}) s\boldsymbol{\Sigma} = D^2_\xi(\boldsymbol{\mu}, \boldsymbol{\Sigma})\boldsymbol{\Sigma}.$$

Thus

$$b = D^2_\xi(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \qquad (2.19)$$

does not depend on $s > 0$ or $\delta \in (0, 0.5]$.

The following assumption (E1) gives a class of distributions where we can prove that the new robust estimators are $\sqrt{n}$ consistent. Cator and Lopuhaä (2009) show that MCD is consistent provided that the MCD functional is unique. Distributions where the functional is unique are called "unimodal," and rule out, for example, a spherically symmetric uniform distribution. Theorem 4 shows that under (E1), both MCD and DGK are estimating $(\boldsymbol{\mu}, a_{MCD}\boldsymbol{\Sigma})$. Olive (2008, § 10.7) and Olive and Hawkins (2010) prove that FCH estimator is a $\sqrt{n}$ consistent estimator of $(\mu, d\ \boldsymbol{\Sigma})$ under the following assumption (E1).

Assumption(E1):

(i) The $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ are iid from a "unimodel" distribution with nonsingular $\text{Cov}(\boldsymbol{x}_i)$;

(ii) g is continuously differentiable with finite 4th moment:

$$\int (\boldsymbol{x}^T\boldsymbol{x})^2 g(\boldsymbol{x}^T\boldsymbol{x}) d\boldsymbol{x} < \infty.$$

Theorem 3. *Assume that (E1) holds and that $(T, \boldsymbol{C})$ is a consistent estimator of $(\boldsymbol{\mu}, s\boldsymbol{\Sigma})$ with rate n^δ where the constants $s > 0$ and $0 < \delta \leq 0.5$. Then the classical estimator $(\overline{\boldsymbol{x}}_{t,j}, \boldsymbol{S}_{t,j})$ computed from the $c_n \approx n/2$ of cases with the smallest distances $D_i(T, \boldsymbol{C})$ is a consistent estimator of $(\boldsymbol{\mu}, a_{MCD}\boldsymbol{\Sigma})$ with the same rate n^δ. See Olive and Hawkins (2010).*

Proof. By Lopuhaä (1999) the estimator is a consistent estimator of $(\boldsymbol{\mu}, a\boldsymbol{\Sigma})$ with rate n^δ. By the remarks above, a will be the same for any consistent estimator of $(\boldsymbol{\mu}, s\boldsymbol{\Sigma})$ and a does not depend on $s > 0$ or $\delta \in (0, 0.5]$. Hence the result follows if $a = a_{MCD}$. The MCD estimator is a $\sqrt{n}$ consistent estimator of $(\boldsymbol{\mu}, a_{MCD}\boldsymbol{\Sigma})$ by Butler, Davies and Jhun (1993) and Cator and Lopuhaä (2009). If the MCD estimator is the start, then it is also the attractor by Rousseeuw and Van Driessen

(1999) who show that concentration does not increase the MCD criterion. Hence $a = a_{MCD}$. □

Theorem 3 shows that $a = a_{MCD}$ where $\xi = 0.5$. Hence concentration with a consistent estimator of $(\boldsymbol{\mu}, s\boldsymbol{\Sigma})$ with rate n^{δ} as a start results in a consistent estimator of $(\boldsymbol{\mu}, a_{MCD}\boldsymbol{\Sigma})$ with rate n^{δ}. This result can be applied iteratively for a finite number of concentration steps. Hence DGK is a $\sqrt{n}$ consistent estimator of the same quantity that MCD is estimating. It is not known if the results hold if concentration is iterated to convergence. For multivariate normal data, $D^2(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim \chi^2_p$.

2.6 THE FCH AND THE RMVN ROBUST ESTIMATORS

This section discusses two $\sqrt{n}$ consistent and outlier resistant estimators, the FCH estimator and RMVN estimator. FCH and RMVN are introduced by Olive and Hawkins (2010).

Both FCH and MBA estimators use two attractors: DGK $(T_{k,D}, \boldsymbol{C}_{k,D})$ and MB $(T_{k,M}, \boldsymbol{C}_{k,M})$. The MBA estimator uses the attractor with the smallest determinant. The FCH estimator, however, uses a location criterion to choose the attractor: if the DGK location estimator $T_{k,D}$ has a greater Euclidean distance from MED$(\boldsymbol{X})$ than $n/2$ cases, then FCH uses the MB attractor $(T_{k,M}, \boldsymbol{C}_{k,M})$ as the final estimator. Otherwise, FCH uses the attractor that has the smallest determinant. In other words, the FCH estimator only uses the attractor with the smallest determinant if

$$||T_{k,D} - \text{MED}(\boldsymbol{X})|| \leq \text{MED}(D_i(\text{MED}(\boldsymbol{X}), \boldsymbol{I}_p)).$$

Let $(T_A, \boldsymbol{C}_A)$ be the attractor finally used. Then the FCH estimator $(T_F, \boldsymbol{C}_F)$ takes

$T_F = T_A$ and

$$\boldsymbol{C}_F = \frac{\text{MED}(D_i^2(T_A, \boldsymbol{C}_A))}{\chi^2_{p,0.5}} \boldsymbol{C}_A \tag{2.20}$$

where $\chi^2_{p,0.5}$ is the 50th percentile of a chi–square distribution with p degrees of freedom.

In particular, $\boldsymbol{C}_F$ estimates $d\boldsymbol{\Sigma}$ with $d = 1$ when $\boldsymbol{x}_1, ..., \boldsymbol{x}_n \overset{\text{iid}}{\sim} N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

For MVN data, $U = D^2(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \sim \chi^2_p$. Similarly, suppose $(T_F, \boldsymbol{C}_F)$ is a consistent estimator of $(\boldsymbol{\mu}, d\ \boldsymbol{\Sigma})$, and that $P(U \leq u_\alpha) = \alpha$. Then, by the scaling in (2.20),

$$\begin{aligned}
(T_F, \boldsymbol{C}_F) &= \left(T_A, \frac{\text{MED}\left(D_i^2(T_A, \boldsymbol{C}_A)\right)}{\chi^2_{p,0.5}} \boldsymbol{C}_A\right) \\
&\overset{P}{\to} \left(\boldsymbol{\mu}, \frac{\text{MED}\left(d^{-1} D^2(\boldsymbol{\mu}, \boldsymbol{\Sigma})\right)}{\chi^2_{p,0.5}} d\boldsymbol{\Sigma}\right) \\
&= \left(\boldsymbol{\mu}, \frac{u_{0.5}}{\chi^2_{p,0.5}} \boldsymbol{\Sigma}\right) \\
&= \left(\boldsymbol{\mu}, d_F \boldsymbol{\Sigma}\right)
\end{aligned}$$

where $d_F = \dfrac{u_{0.5}}{\chi^2_{p,0.5}}$. It is obvious that $d_F = 1$ for MVN data since $u_{0.5} = \chi^2_{p,0.5}$.

The next theorem shows the FCH estimator $\sqrt{n}$ consistent and T_{FCH} is high breakdown.

Theorem 4. *T_{FCH} is high breakdown. Suppose (E1) holds. If $(T_A, \boldsymbol{C}_A)$ is the DGK or MB attractor with the smallest determinant, then $(T_A, \boldsymbol{C}_A)$ is a $\sqrt{n}$ consistent estimator of $(\boldsymbol{\mu}, a_{MCD}\boldsymbol{\Sigma})$. Hence the MBA and FCH estimators are outlier resistant $\sqrt{n}$ consistent estimators of $(\boldsymbol{\mu}, c\boldsymbol{\Sigma})$ where $c = 1$ for multivariate normal data. See Olive and Hawkins (2010).*

Proof. T_{FCH} is high breakdown since it is a bounded distance from $\text{MED}(\boldsymbol{X})$ even if the number of outliers is close to $n/2$. Under (E1) the FCH and MBA estimators are asymptotically equivalent since $\|T_{k,D} - \text{MED}(\boldsymbol{X})\| \to 0$ in probability. The estimator satisfies $0 < det(\boldsymbol{C}_{MCD}) \leq det(\boldsymbol{C}_A) \leq det(\boldsymbol{S}_{0,M}) < \infty$ by Theorem 2 if up to nearly 50% of the cases are outliers. If the distribution is spherical about $\boldsymbol{\mu}$, then the result follows from Pratt (1959) and Theorem 3 since both starts are $\sqrt{n}$ consistent. Otherwise, the MB estimator $\boldsymbol{S}_{k,M}$ is a biased estimator of $a_{MCD}\boldsymbol{\Sigma}$. But the DGK estimator $\boldsymbol{S}_{k,D}$ is a $\sqrt{n}$ consistent estimator of $a_{MCD}\boldsymbol{\Sigma}$ by Theorem 3 and $\|\boldsymbol{C}_{MCD} - \boldsymbol{S}_{k,D}\| = O_P(n^{-1/2})$. Thus the probability that the DGK attractor minimizes the determinant goes to one as $n \to \infty$, and $(T_A, \boldsymbol{C}_A)$ is asymptotically equivalent to the DGK estimator $(\overline{\boldsymbol{x}}_{k,D}, \boldsymbol{S}_{k,D})$.

Let $P(U \leq u_\alpha) = \alpha$ where U is given by (1.3). Then the scaling in (2.20) makes $\boldsymbol{C}_F$ a consistent estimator of $c\boldsymbol{\Sigma}$ where $c = u_{0.5}/\chi^2_{p,0.5}$, and $c = 1$ for multivariate normal data. □

We also considered several estimators that use the MB and DGK estimators as attractors. CMVE is a concentration algorithm like FCH, but the "MVE" criterion is used in place of the MCD criterion. A standard method of reweighting can be used to produce the RMBA, RFCH and RCMVE estimators. RMVN uses a slightly modified method of reweighting so that RMVN gives good estimates of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for multivariate normal data, even when certain types of outliers are present.

The RFCH estimator uses two standard reweighting steps. Let $(\hat{\boldsymbol{\mu}}_1, \tilde{\boldsymbol{\Sigma}}_1)$ be the classical estimator applied to the n_1 cases with $D_i^2(T_{FCH}, \boldsymbol{C}_{FCH}) \leq \chi^2_{p,0.975}$, and let

$$\hat{\boldsymbol{\Sigma}}_1 = \frac{\text{MED}(D_i^2(\hat{\boldsymbol{\mu}}_1, \tilde{\boldsymbol{\Sigma}}_1))}{\chi^2_{p,0.5}} \tilde{\boldsymbol{\Sigma}}_1.$$

Then let $(T_{RFCH}, \tilde{\boldsymbol{\Sigma}}_2)$ be the classical estimator applied to the cases with $D_i^2(\hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}}_1) \leq \chi^2_{p,0.975}$, and let

$$\boldsymbol{C}_{RFCH} = \frac{\text{MED}(D_i^2(T_{RFCH}, \tilde{\boldsymbol{\Sigma}}_2))}{\chi^2_{p,0.5}} \tilde{\boldsymbol{\Sigma}}_2.$$

RMBA and RFCH are $\sqrt{n}$ consistent estimators of $(\boldsymbol{\mu}, c\boldsymbol{\Sigma})$ by Lopuhaä (1999) where the weight function uses $h^2 = \chi^2_{p,0.975}$, but the two estimators use nearly 97.5% of the cases if the data is multivariate normal. We conjecture CMVE and RMVE are also $\sqrt{n}$ consistent estimators of $(\boldsymbol{\mu}, c\boldsymbol{\Sigma})$.

RMVN is a weighted FCH estimator. Let $(\hat{\boldsymbol{\mu}}_1, \tilde{\boldsymbol{\Sigma}}_1)$ be the classical estimator applied to the n_1 cases with $D_i^2(T_{FCH}, \boldsymbol{C}_{FCH}) \leq \chi^2_{p,0.975}$, and let $q_1 = \min\{0.5(0.975)n/n_1, 0.995\}$ and

$$\hat{\boldsymbol{\Sigma}}_1 = \frac{\text{MED}(D_i^2(\hat{\boldsymbol{\mu}}_1, \tilde{\boldsymbol{\Sigma}}_1))}{\chi^2_{p,q_1}} \tilde{\boldsymbol{\Sigma}}_1.$$

Then let $(T_{RMVN}, \tilde{\boldsymbol{\Sigma}}_2)$ be the classical estimator applied to the n_2 cases with $D_i^2(\hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}}_1)) \leq \chi^2_{p,0.975}$. Let $q_2 = \min\{0.5(0.975)n/n_2, 0.995\}$, and

$$\boldsymbol{C}_{RMVN} = \frac{\text{MED}(D_i^2(T_{RMVN}, \tilde{\boldsymbol{\Sigma}}_2))}{\chi^2_{p,q_2}} \tilde{\boldsymbol{\Sigma}}_2.$$

RMVN is $\sqrt{n}$ consistent by Lopuhaä (1999). If the bulk of the data is $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the RMVN estimator can give useful estimates of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for certain types of outliers where FCH and RFCH estimate $(\boldsymbol{\mu}, d_E\boldsymbol{\Sigma})$ for $d_E > 1$. To see this claim, let $0 \leq \gamma < 0.5$ be the outlier proportion. If $\gamma = 0$, then $n_i/n \xrightarrow{P} 0.975$ and $q_i \xrightarrow{P} 0.5$. If $\gamma > 0$, suppose the outlier configuration is such that the $D_i^2(T_{FCH}, \boldsymbol{C}_{FCH})$ are roughly χ^2_p for the clean cases, and the outliers have larger D_i^2 than the clean cases. Then $\text{MED}(D_i^2) \approx \chi^2_{p,q}$ where $q = 0.5/(1-\gamma)$. For example, if $n = 100$ and $\gamma = 0.4$, then there are 60 clean cases, $q = 5/6$, and the quantile $\chi^2_{p,q}$ is being estimated instead of $\chi^2_{p,0.5}$. Now $n_i \approx n(1-\gamma)0.975$, and q_i estimates q. Thus $\boldsymbol{C}_{RMVN} \approx \boldsymbol{\Sigma}$. Of course consistency cannot generally be claimed when outliers are present.

Simulation 1

Simulations suggested $(T_{RMVN}, \boldsymbol{C}_{RMVN})$ gives useful estimates of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for a variety of outlier configurations. Using 20 runs and $n = 1000$, the averages of the dispersion matrices were computed when the bulk of the data are iid $N_2(\boldsymbol{0}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} = diag(1, 2)$. For clean data, FCH, RFCH and RMVN give $\sqrt{n}$ consistent estimators of $\boldsymbol{\Sigma}$, while FMCD and OGK seem to be approximately unbiased for $\boldsymbol{\Sigma}$. The median ball estimator was scaled using (2.15) and estimated $diag(1.13, 1.85)$.

Next the data had $\gamma = 0.4$ and the outliers had $\boldsymbol{x} \sim N_2((0, 15)^T, 0.0001\boldsymbol{I}_2)$, a near point mass at the major axis. FCH, MB and RFCH estimated $2.6\boldsymbol{\Sigma}$ while RMVN estimated $\boldsymbol{\Sigma}$. FMCD and OGK failed to estimate $d\ \boldsymbol{\Sigma}$. Note that $\chi^2_{2,5/6}/\chi^2_{2,0.5} = 2.585$. See Table 2.1.

Table 2.1. Average Dispersion Matrices for Near Point Mass Outliers

RMVN	FMCD
$\begin{bmatrix} 1.002 & -0.014 \\ -0.014 & 2.024 \end{bmatrix}$	$\begin{bmatrix} 0.055 & 0.685 \\ 0.685 & 122.46 \end{bmatrix}$
OGK	MB
$\begin{bmatrix} 0.185 & 0.089 \\ 0.089 & 36.244 \end{bmatrix}$	$\begin{bmatrix} 2.570 & -0.082 \\ -0.082 & 5.241 \end{bmatrix}$

Table 2.2. Average Dispersion Matrices for Mean Shift Outliers

RMVN	FMCD
$\begin{bmatrix} 0.990 & 0.004 \\ 0.004 & 2.014 \end{bmatrix}$	$\begin{bmatrix} 2.530 & 0.003 \\ 0.003 & 5.146 \end{bmatrix}$
OGK	MB
$\begin{bmatrix} 19.671 & 12.875 \\ 12.875 & 39.724 \end{bmatrix}$	$\begin{bmatrix} 2.552 & 0.003 \\ 0.003 & 5.118 \end{bmatrix}$

Next the data had $\gamma = 0.4$ and the outliers had $\boldsymbol{x} \sim N_2((20, 20)^T, \boldsymbol{\Sigma})$, a mean shift with the same covariance matrix as the clean cases. Rocke and Woodruff (1996) suggest that outliers with mean shift are hard to detect. FCH, FMCD, MB and RFCH estimated $2.6\boldsymbol{\Sigma}$ while RMVN estimated $\boldsymbol{\Sigma}$, and OGK failed. See Table

2.2.

Example 1. Tremearne (1911) recorded *height* $= x_1$ and *height while kneeling* $=$ x_2 of 112 people. Figure 2.1 shows a scatterplot of the data. Case 3 has the largest Euclidean distance of 214.767 from $\text{MED}(\boldsymbol{X}) = (1680, 1240)^T$, but if the distances correspond to the contours of a covering ellipsoid, then case 44 has the largest distance. The hypersphere (circle) centered at $\text{MED}(\boldsymbol{X})$ that covers half the data is small because the data density is high near $\text{MED}(\boldsymbol{X})$. The median Euclidean distance is 59.661 and case 44 has Euclidean distance 77.987. Hence the intersection of the sphere and the data is a highly correlated clean ellipsoidal region. The Rousseeuw and Van Driessen (1999) DD plot is a plot of classical distances (MD) versus "robust" distances (RD). Figure 2.2 shows the DD plot using the MB estimator. Notice that both the classical and MB estimators give the largest distances to cases 3 and 44. As the dimension p gets larger, outliers that can not be detected by marginal methods (case 44 in Example 1) become harder to detect.

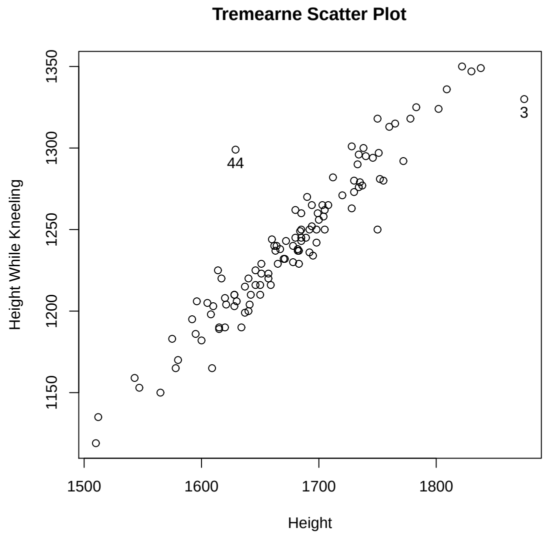

Figure 2.1. Plots for Tremearne (1911) Data

Example 2. The estimators can be useful when the data is not elliptically con-

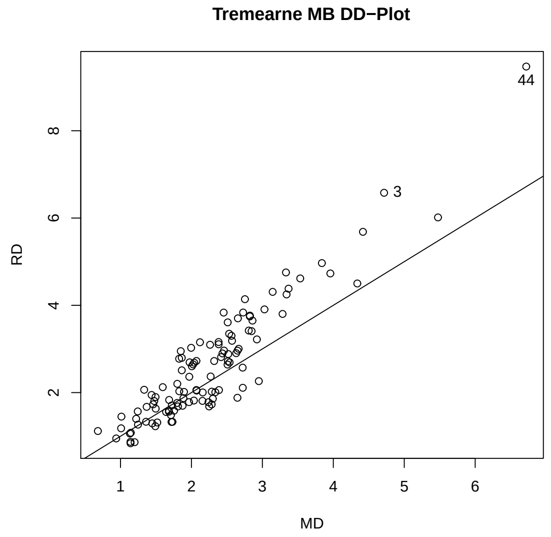

Figure 2.2. DD Plots for Tremearne (1911) Data

toured. The Gladstone (1905-6) data has 12 variables on 267 persons after death. Head measurements were *breadth, circumference, head height, length* and *size* as well as *cephalic index* and *brain weight. Age*, *height* and three categorical variables *ageclass* (0: under 20, 1: 20-45, 2: over 45), *sex* and *cause* of death (1: acute, 2: not given, 3: chronic) were also given. Figure 2.3 shows the DD plots for the FCH, CMVE, FMCD and MB estimators. The plots were similar and six outliers correspond to the six infants in the data set.

Olive (2002) showed that if a consistent robust estimator is scaled as in (2.20), then the plotted points in the DD plot will cluster about the identity line with unit slope and zero intercept if the data is multivariate normal, and about some other line through the origin if the data is from some other elliptically contoured distribution with a nonsingular covariance matrix. Since multivariate procedures tend to perform well for elliptically contoured data, the DD plot is useful even if outliers are not present.

Simulation 2

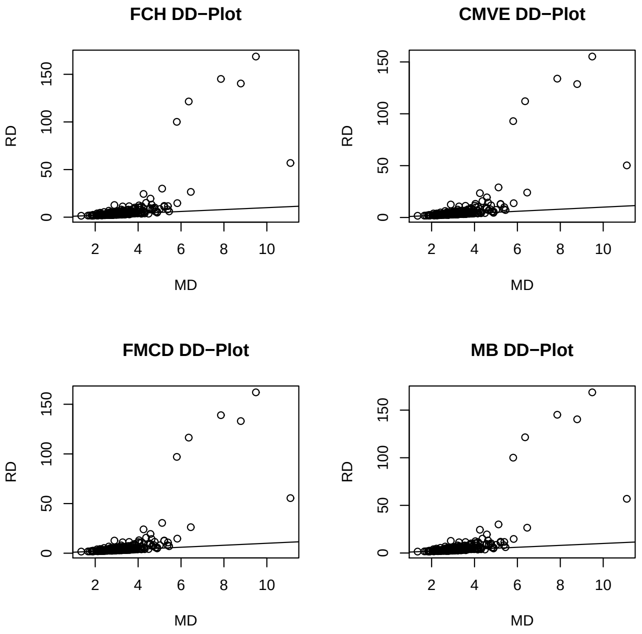

Figure 2.3. DD Plots for Gradstone (1905-6) Data

Table 2.3. Scaled Variance $nS^2(T_p)$ and $nS^2(C_{p,p})$

p	n	V	FCH	RFCH	RMVN	DGK	OGK	CLAS	FMCD	MB
5	50	C	216.0	72.4	75.1	209.3	55.8	47.12	153.9	145.8
5	50	T	12.14	6.50	6.88	10.56	6.70	4.83	8.38	13.23
5	5000	C	307.6	64.1	68.6	325.7	59.3	48.5	60.4	309.5
5	5000	T	18.6	5.34	5.33	19.33	6.61	4.98	5.40	20.20
10	100	C	817.3	276.4	286.0	725.4	229.5	198.9	459.6	610.4
10	100	T	21.40	11.42	11.68	20.13	12.75	9.69	14.05	24.13
10	5000	C	955.5	237.9	243.8	966.2	235.8	202.4	233.6	975.0
10	5000	T	29.12	10.08	10.09	29.35	12.81	9.48	10.06	30.20

If $W_{in} \sim N(0, \tau^2/n)$ for $i = 1, ..., r$ and if S^2_W is the sample variance of the W_{in}, then $E(nS^2_W) = \tau^2$ and $V(nS^2_W) = 2\tau^4/(r-1)$. So $nS^2_W \pm \sqrt{5}SE(nS^2_W) \approx \tau^2 \pm \sqrt{10}\tau^2/\sqrt{r-1}$. So for $r = 1000$ runs, expect nS^2_W to be between $\tau^2 - 0.1\tau^2$ and $\tau^2 + 0.1\tau^2$ with high confidence. Similar results hold for many estimators if W_{in} is $\sqrt{n}$ consistent and asymptotically normal and if n is large enough. If W_{in} has less than $\sqrt{n}$ rate, e.g. $n^{1/3}$ rate, then the scaled sample variance $nS^2_W \to \infty$ as $n \to \infty$.

Table 2.3 considers $W = T_p$ and $W = C_{p,p}$ for eight estimators, $p = 5$ and 10 and $n = 10p$ and 5000 when $\boldsymbol{x} \sim N_p(\boldsymbol{0}, diag(1, ..., p))$. For the classical estimator, $T_p = \overline{x}_p \sim N(0, p/n)$, and $nS^2(T_p) \approx p$ while $C_{p,p}$ is the sample variance of n iid $N(0, p)$ random variables. Hence $nS^2(C_{p,p}) \approx 2p^2$. RFCH, RMVN, FMCD and possibly OGK use a "reweight for efficiency" concentration step that uses a random number of cases with percentage close to 97.5%. These four estimators had similar behavior. DGK, FCH and MB used about 50% of the cases and had similar behavior. By Lopuhaä (1999), estimators with less than $\sqrt{n}$ rate still have zero efficiency after the reweighting. Although FMCD, MB and OGK have not been proven to be $\sqrt{n}$ consistent, their values did not blow up even for $n = 5000$.

Simulation 3

The collection of rpack.txt functions from www.math.siu.edu/olive/rpack.txt has the function *corrsim2*. Put it into R and type library(MASS). It generates data $\boldsymbol{x}$ then multiplies $\boldsymbol{x}$ by $diag(\sqrt{1}, ..., \sqrt{p})$ where p is the dimension of the $\boldsymbol{x}$ vector. There are 7 distributions for $\boldsymbol{x}$ considered:

1: $N_p(0, \boldsymbol{I}_p)$

2: $0.6N_p(0, \boldsymbol{I}_p) + 0.4N_p(0, 25\boldsymbol{I}_p)$

3: $0.4N_p(0, \boldsymbol{I}_p) + 0.6N_p(0, 25\boldsymbol{I}_p)$

4: $0.9N_p(0, \boldsymbol{I}_p) + 0.1N_p(0, 25\boldsymbol{I}_p)$

5: $0.75N_p(0, \boldsymbol{I}_p) + 0.25N_p(0, 25\boldsymbol{I}_p)$

6,7: multivariate t_3, t_5.

There are 7 estimators of multivariate location and dispersion type = 1 MBA, 2 RMBA, 3 cov.mcd, 4 FCH, 5 RFCH, 6 CMVE, and 7 RCMVE. The function computes the correlation between the robust Mahalanobis distances and the classical Mahalanobis distances. For each of the 49 $\boldsymbol{x}$ type (xt) and estimator type (et) combinations, the smallest value of n such that the correlation is 0.95 or higher is found for $p = 2, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95$ and 100. Table 2.4 and 2.5 present the result of simulation 3 for $xt = 1, 2, 3, 7$ and $et = 4, 5, 6, 7$. Both tables show FCH and CMVE are nearly identical, and RFCH and RCMVE are nearly identical. As p increases, n increases with a smaller rate of change than p. For example, $n \approx 50p$ when $p = 2$, but $n \approx 10p$ when $p = 100$.

2.7 OUTLIER RESISTANCE

Geometrical arguments suggest that the MB estimator has considerable outlier resistance. Suppose the outliers are far from the bulk of the data. Let the "median ball" correspond to the half set of data closest to MED($\boldsymbol{X}$) in Euclidean distance. If the outliers are outside of the median ball, then the initial half set in the iteration leading to the MB estimator will be clean. Thus the MB estimator will tend to give the outliers the largest MB distances unless the initial clean half set has very high correlation in a direction about which the outliers lie. This property holds for very general outlier configurations. The FCH estimator tries to use the DGK attractor if det($\boldsymbol{C}_{DGK}$) is small and the DGK location estimator T_{DGK} is in the median ball. Distant outliers that make det($\boldsymbol{C}_{DGK}$) small also drag T_{DGK} outside of the median ball. Then FCH uses the MB attractor.

Compared to OGK and FMCD, the MB estimator is vulnerable to outliers

Table 2.4. Smallest n for $corr(RD, MD) > .95$,   xt=1,2

p	xt	et	n	p	xt	et	n	p	xt	et	n	p	xt	et	n
2	1	4	170	25	1	5	220	50	1	6	870	75	1	7	760
2	1	5	30	25	1	6	520	50	1	7	470	80	1	4	1170
2	1	6	130	25	1	7	220	55	1	4	830	80	1	5	810
2	1	7	30	30	1	4	560	55	1	5	510	80	1	6	1330
5	1	4	260	30	1	5	260	55	1	6	940	80	1	7	820
5	1	5	50	30	1	6	580	55	1	7	530	85	1	4	1240
5	1	6	250	30	1	7	270	60	1	4	890	85	1	5	870
5	1	7	50	35	1	4	600	60	1	5	570	85	1	6	1440
10	1	4	340	35	1	5	310	60	1	6	1010	85	1	7	880
10	1	5	100	35	1	6	640	60	1	7	580	90	1	4	1320
10	1	6	320	35	1	7	320	65	1	4	960	90	1	5	930
10	1	7	100	40	1	4	650	65	1	5	630	90	1	6	1520
15	1	4	390	40	1	5	360	65	1	6	1130	90	1	7	950
15	1	5	130	40	1	6	710	65	1	7	640	95	1	4	1390
15	1	6	390	40	1	7	370	70	1	4	1030	95	1	5	990
15	1	7	140	45	1	4	710	70	1	5	690	95	1	6	1620
20	1	4	450	45	1	5	410	70	1	6	1160	95	1	7	1010
20	1	5	170	45	1	6	780	70	1	7	700	100	1	4	1470
20	1	6	460	45	1	7	420	75	1	4	1120	100	1	5	1060
20	1	7	180	50	1	4	770	75	1	5	740	100	1	6	1730
25	1	4	500	50	1	5	470	75	1	6	1250	100	1	7	1080
2	2	4	40	25	2	5	90	50	2	6	130	75	2	7	180
2	2	5	30	25	2	6	90	50	2	7	130	80	2	4	190
2	2	6	30	25	2	7	90	55	2	4	140	80	2	5	190
2	2	7	30	30	2	4	100	55	2	5	140	80	2	6	190
5	2	4	60	30	2	5	100	55	2	6	140	80	2	7	190
5	2	5	50	30	2	6	100	55	2	7	140	85	2	4	200
5	2	6	50	30	2	7	100	60	2	4	160	85	2	5	200
5	2	7	50	35	2	4	110	60	2	5	160	85	2	6	200
10	2	4	70	35	2	5	110	60	2	6	150	85	2	7	200
10	2	5	60	35	2	6	110	60	2	7	150	90	2	4	210
10	2	6	60	35	2	7	110	65	2	4	160	90	2	5	210
10	2	7	60	40	2	4	120	65	2	5	160	90	2	6	210
15	2	4	70	40	2	5	120	65	2	6	160	90	2	7	210
15	2	5	70	40	2	6	110	65	2	7	160	95	2	4	220
15	2	6	70	40	2	7	110	70	2	4	170	95	2	5	220
15	2	7	70	45	2	4	120	70	2	5	170	95	2	6	220
20	2	4	80	45	2	5	120	70	2	6	170	95	2	7	220
20	2	5	80	45	2	6	120	70	2	7	170	100	2	4	230
20	2	6	80	45	2	7	120	75	2	4	180	100	2	5	230
20	2	7	80	50	2	4	140	75	2	5	180	100	2	6	230
25	2	4	90	50	2	5	140	75	2	6	180	100	2	7	230

Table 2.5. Smallest n for $corr(RD, MD) > .95$, xt=3,7

p	xt	et	n	p	xt	et	n	p	xt	et	n	p	xt	et	n
2	3	4	50	25	3	5	280	50	3	6	490	75	3	7	730
2	3	5	40	25	3	6	250	50	3	7	500	80	3	4	760
2	3	6	40	25	3	7	290	55	3	4	520	80	3	5	760
2	3	7	40	30	3	4	290	55	3	5	520	80	3	6	770
5	3	4	80	30	3	5	320	55	3	6	540	80	3	7	790
5	3	5	90	30	3	6	300	55	3	7	550	85	3	4	800
5	3	6	80	30	3	7	330	60	3	4	560	85	3	5	800
5	3	7	90	35	3	4	330	60	3	5	570	85	3	6	820
10	3	4	130	35	3	5	360	60	3	6	580	85	3	7	840
10	3	5	140	35	3	6	350	60	3	7	590	90	3	4	850
10	3	6	120	35	3	7	370	65	3	4	600	90	3	5	850
10	3	7	140	40	3	4	380	65	3	5	610	90	3	6	880
15	3	4	160	40	3	5	400	65	3	6	630	90	3	7	880
15	3	5	190	40	3	6	390	65	3	7	640	95	3	4	890
15	3	6	170	40	3	7	410	70	3	4	650	95	3	5	890
15	3	7	190	45	3	4	430	70	3	5	660	95	3	6	920
20	3	4	200	45	3	5	440	70	3	6	680	95	3	7	920
20	3	5	240	45	3	6	440	70	3	7	690	100	3	4	940
20	3	6	210	45	3	7	450	75	3	4	700	100	3	5	950
20	3	7	240	50	3	4	470	75	3	5	700	100	3	6	980
25	3	4	240	50	3	5	480	75	3	6	730	100	3	7	980
2	7	4	80	25	7	5	190	50	7	6	290	75	7	7	410
2	7	5	30	25	7	6	170	50	7	7	300	80	7	4	410
2	7	6	80	25	7	7	190	55	7	4	290	80	7	5	410
2	7	7	30	30	7	4	200	55	7	5	320	80	7	6	410
5	7	4	100	30	7	5	220	55	7	6	290	80	7	7	410
5	7	5	70	30	7	6	200	55	7	7	320	85	7	4	410
5	7	6	100	30	7	7	220	60	7	4	320	85	7	5	430
5	7	7	70	35	7	4	210	60	7	5	340	85	7	6	430
10	7	4	120	35	7	5	230	60	7	6	320	85	7	7	430
10	7	5	100	35	7	6	230	60	7	7	350	90	7	4	440
10	7	6	120	35	7	7	230	65	7	4	340	90	7	5	440
10	7	7	100	40	7	4	230	65	7	5	360	90	7	6	450
15	7	4	140	40	7	5	260	65	7	6	350	90	7	7	450
15	7	5	140	40	7	6	240	65	7	7	360	95	7	4	460
15	7	6	140	40	7	7	260	70	7	4	360	95	7	5	460
15	7	7	140	45	7	4	250	70	7	5	360	95	7	6	480
20	7	4	150	45	7	5	270	70	7	6	360	95	7	7	480
20	7	5	170	45	7	6	270	70	7	7	370	100	7	4	480
20	7	6	150	45	7	7	290	75	7	4	390	100	7	5	480
20	7	7	170	50	7	4	270	75	7	5	390	100	7	6	490
25	7	4	170	50	7	5	300	75	7	6	410	100	7	7	490

Table 2.6. Number of Times Mean Shift Outliers had the Largest Distances

p	γ	n	pm	MBA	FCH	RFCH	RMVN	OGK	FMCD	MB
10	.1	100	4	49	49	85	84	38	76	57
10	.1	100	5	91	91	99	99	93	98	91
10	.4	100	7	90	90	90	90	0	48	100
40	.1	100	5	3	3	3	3	76	3	17
40	.1	100	8	36	36	37	37	100	49	86
40	.25	100	20	62	62	62	62	100	0	100
40	.4	100	20	20	20	20	20	0	0	100
40	.4	100	35	44	98	98	98	95	0	100
60	.1	200	10	49	49	49	52	100	30	100
60	.1	200	20	97	97	97	97	100	35	100
60	.25	200	25	60	60	60	60	100	0	100
60	.4	200	30	11	21	21	21	17	0	100
60	.4	200	40	21	100	100	100	100	0	100

that lie within the median ball. If the bulk of the data is highly correlated with the major axis of an ellipsoidal region, then the distances based on the clean data can be very large for outliers that fall within the median ball. The outlier resistance of the MB estimator decreases as p increases since the volume of the median ball rapidly increases with p.

A simple simulation for outlier resistance is to count the number of times the minimum distance of the outliers is larger than the maximum distance of the clean cases. The simulation used 100 runs. If the count was 97, then in 97 data sets the outliers can be separated from the clean cases with a horizontal line in the DD plot, but in 3 data sets the robust distances did not achieve complete separation.

The clean cases had $\boldsymbol{x} \sim N_p(\boldsymbol{0}, diag(1,2,...,p))$. Outlier types were the mean shift $\boldsymbol{x} \sim N_p(pm\boldsymbol{1}, diag(1,2,...,p))$ where $\boldsymbol{1} = (1,...,1)^T$, and $\boldsymbol{x} \sim N_p((0,...,0,pm)^T, 0.0001\boldsymbol{I}_p)$, a near point mass at the major axis. Notice that the clean data can be transformed to a $N_p(\boldsymbol{0}, \boldsymbol{I}_p)$ distribution by multiplying $\boldsymbol{x}_i$ by $\text{diag}(1, 1/\sqrt{2}, ..., 1/\sqrt{p})$, and this transformation changes the location of the near point mass to $(0,...,0,pm/\sqrt{p})^T$.

For near point mass outliers, an ellipsoid with very small volume can cover half of the data if the outliers are at one end of the ellipsoid and some of the clean data are at the other end. This half set will produce a classical estimator with very small determinant by (2.15). In the simulations for large γ, as the near point mass is moved very far away from the bulk of the data, only the classical, MB and OGK estimators did not have numerical difficulties. Since the MCD estimator has smaller determinant than DGK while MVE has smaller volume than DGK, estimators like FAST-MCD and MBA that use the MVE or MCD criterion without using location information will be vulnerable to these outliers. Following Hawkins and Olive (2002), FAST-MCD is also vulnerable to outliers if γ is slightly larger than γ_o given by

$$\gamma_o \approx \min\left(0.5, 1 - \left[1 - (0.2)^{1/K}\right]^{1/(p+1)}\right) 100\%.$$

Table 2.7. Number of Times Near Point Mass Outliers had the Largest Distances

p	γ	n	pm	MBA	FCH	RFCH	RMVN	OGK	FMCD	MB
10	.1	100	40	73	92	92	92	100	95	100
10	.25	100	25	0	99	99	90	0	0	99
10	.4	100	25	0	100	100	100	0	0	100
40	.1	100	80	0	0	0	0	79	0	80
40	.1	100	150	0	65	65	65	100	0	99
40	.25	100	90	0	88	87	87	0	0	88
40	.4	100	90	0	91	91	91	0	0	91
60	.1	200	100	0	0	0	0	13	0	91
60	.25	200	150	0	100	100	100	0	0	100
60	.4	200	150	0	100	100	100	0	0	100
60	.4	200	20000	0	100	100	100	64	0	100

Tables 2.6 and 2.7 help illustrate the results for the simulation. Large counts and small pm for fixed γ suggest greater ability to detect outliers. Values of p were 5, 10, 15, ..., 60. First consider the mean shift outliers and Table 4. For $\gamma = 0.25$ and 0.4, MB usually had the highest counts. For $5 \leq p \leq 20$ and the mean shift, the

OGK estimator often had the smallest counts, although FMCD could not handle 40% outliers for $p = 20$. For $25 \leq p \leq 60$, OGK usually had the highest counts for $\gamma = 0.05$ and 0.1. For $p \geq 30$, FMCD could not handle 25% outliers even for enormous values of pm.

In Table 2.7, FCH greatly outperformed MBA although the only difference between the two estimators is that FCH uses a location criterion as well as the MCD criterion. OGK performed well for $\gamma = 0.05$ and $20 \leq p \leq 60$ (not tabled). For large γ, OGK often has large bias for $c\boldsymbol{\Sigma}$. Then the outliers may need to be enormous before OGK can detect them. Also see Table 2.2, where OGK gave the outliers the largest distances for all runs, but $\boldsymbol{C}_{OGK}$ does not give a good estimate of $c\boldsymbol{\Sigma} = c \;\; diag(1, 2)$.

The table 6.13 in Appendix D compares the FCH, RFCH, CMVE, RCMVE and MB estimators. It shows the result of a simulation function *mldsim5* available at http://www.math.siu.edu/olive/rpack.txt. The bulk of the data X is $N\big((0, ..., 0)', diag(1, ..., p)\big)$. Four types of the outliers are simulated.

1: $N\big((0, ..., pm)', .0001\boldsymbol{I}_p\big)$

2: $N\big((pm, ..., 0)', .0001\boldsymbol{I}_p\big)$

3: $N\big((pm, ..., pm)', diag(1, ..., p)$

4: $X[i, p] = pm.$

The program counts the number of times all of the outliers have larger Mahalanobis distances than the biggest distance of the clean cases. The table shows how large pm is before the outliers are large, say that the counts are greater than 90. Typically MB works best but not always.

CHAPTER 3

ROBUST PRINCIPAL COMPONENT ANALYSIS

3.1 INTRODUCTION

Principal component analysis (PCA) is used to explain the dispersion structure with a few uncorrelated linear combinations of the original variables, called principal components. The analysis is used for data reduction and interpretation.

Let $\boldsymbol{x} = [X_1, X_2, \cdots, X_p]'$ be a p-dimensional random vector with the covariance matrix $\boldsymbol{\Sigma} = \mathrm{Cov}(\boldsymbol{x})$. The first principal component is the linear combination $\boldsymbol{a}_1'\boldsymbol{x}$ that maximizes $\mathrm{Var}(\boldsymbol{a}'\boldsymbol{x})$ subject to $||\boldsymbol{a}|| = 1$.

$$\boldsymbol{a}_1' = \underset{||\boldsymbol{a}||=1}{\arg\max}\, \mathrm{Var}(\boldsymbol{a}'\boldsymbol{x}) \tag{3.1}$$

The second principal component is the linear combination $\boldsymbol{a}_2'\boldsymbol{x}$ that maximizes $\mathrm{Var}(\boldsymbol{a}'\boldsymbol{x})$ subject to $||\boldsymbol{a}|| = 1$ and $\mathrm{Cov}(\boldsymbol{a}'\boldsymbol{x}, \boldsymbol{a}_1'\boldsymbol{x}) = 0$.

$$\boldsymbol{a}_2' = \underset{||\boldsymbol{a}||=1, \boldsymbol{a}\perp\boldsymbol{a}_1}{\arg\max}\, \mathrm{Var}(\boldsymbol{a}'\boldsymbol{x}) \tag{3.2}$$

In general, the jth principal component is the linear combination $\boldsymbol{a}_j'\boldsymbol{x}$ that maximizes $\mathrm{Var}(\boldsymbol{a}'\boldsymbol{x})$ subject to $||\boldsymbol{a}|| = 1$ and $\mathrm{Cov}(\boldsymbol{a}'\boldsymbol{x}, \boldsymbol{a}_k'\boldsymbol{x}) = 0$ for all $k < j$.

$$\boldsymbol{a}_i' = \underset{||\boldsymbol{a}||=1, \boldsymbol{a}_i\perp\boldsymbol{a}_1, \cdots \boldsymbol{a}_{i-1}}{\arg\max}\, \mathrm{Var}(\boldsymbol{a}_i'\boldsymbol{x}) \tag{3.3}$$

Assume $\boldsymbol{\Sigma}$ has eigenvalue-eigenvector pairs $(\lambda_1, \boldsymbol{e}_1), (\lambda_2, \boldsymbol{e}_2), ..., (\lambda_p, \boldsymbol{e}_p)$ where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$. Then the jth principal component is given by

$$Y_j = \boldsymbol{e}_j^T \boldsymbol{x} = e_{j1}' X_1 + e_{j2}' X_2 + \cdots + e_{jp}' X_p.$$

In addition,

$$\sum_{j=1}^{p} \mathrm{Var}(X_i) = \lambda_1 + \lambda_2 + \cdots + \lambda_p = \sum_{j=1}^{p} \mathrm{Var}(Y_i).$$

Thus the proportion of total variance explained by jth principal component is

$$\frac{\mathrm{Var}(Y_j)}{\sum_{j=1}^{p} \mathrm{Var}(Y_i)} = \frac{\lambda_j}{\lambda_1 + \lambda_2 + \cdots + \lambda_p} \qquad j = 1, 2, \cdots, p$$

The analysis can also be based on the p eigenvalue eigenvector pairs $(\lambda_i, \boldsymbol{e}_i)$ of the correlation matrix $\boldsymbol{\rho}$. However, since the $(\lambda_j, \boldsymbol{e}_j)$ derived from $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ are different from the ones derived from $\boldsymbol{\rho}$, the principal components derived from $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ are consequently different from the ones derived from $\boldsymbol{\rho}$. In general, the results of PCA are not invariant under affine transformation. PCA can be invariant only under orthogonal transformations. Hence a previous rescaling or standardizing of the variables are usually recommended if the units of measurement are not commensurate.

The sample analogs use the sample covariance matrix. Assume that the sample covariance matrix $\boldsymbol{S}$ has eigenvalue-eigenvector pairs $(\hat{\lambda}_1, \hat{\boldsymbol{e}}_1), (\hat{\lambda}_2, \hat{\boldsymbol{e}}_2), ..., (\hat{\lambda}_p, \hat{\boldsymbol{e}}_p)$ where $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_p \geq 0$. Then the principal components corresponding to the jth case are $\hat{Y}_{1j} = \hat{\boldsymbol{e}}_1' \boldsymbol{x}_j, ..., \hat{Y}_{pj} = \hat{\boldsymbol{e}}_p' \boldsymbol{x}_j$. The estimated proportion of the total population variance due to the ith principal component is $\hat{\lambda}_i / \sum_{j=1}^{p} \hat{\lambda}_j$. The analysis can also be based on the p eigenvalue eigenvector pairs $(\hat{\lambda}_i, \hat{\boldsymbol{e}}_i)$ of the sample correlation matrix $\boldsymbol{R}$. However, for classical PCA based on sample covariance matrix or sample correlation matrix, outliers possibly have a distorting result on the results.

Example 3. The 11th observation in the table below is an obvious outlier.

Without deleting the outlier,

$$S = \begin{bmatrix} 188.88 & -3.95 \\ -3.95 & 98.39 \end{bmatrix}, \quad \hat{\lambda}_1 = 189.05 \text{ and } \hat{\lambda}_2 = 98.21.$$

Table 3.1. Outlier Effect on PCA

	1	2	3	4	5	6	7	8	9	10	11	12
x	41	38	63	63	76	58	51	55	60	85	44	52
y	25	24	17	12	18	17	24	23	19	13	-12	15

Hence the first principal component of the sample covariance matrix of the data explains $\frac{188.98}{188.98 + 98.39} = 65.8\%$ of the variablity.

After deleting the outlier,

$$S = \begin{bmatrix} 188.85 & -44.93 \\ -44.93 & 21.16 \end{bmatrix}, \quad \hat{\lambda}_1 = 200.13 \text{ and } \hat{\lambda}_2 = 9.89.$$

Hence the first principal component of the sample covariance matrix of the data explains $\frac{200.13}{200.13 + 9.89} = 95.3\%$ of the variablity. The distorting effect of the outlier on PCA in this example is significant.

To avoid drawing false and misleading conclusions from contaminated data, one has to robustify the PCA procedure. A simple and intuitively appealing way is to replace the classical covariance or correlation matrix with the robust dispersion covariance matrix or correlation matrix. See Croux and Haesbroeck (2000). Another very popular method is Projection-Pursuit (PP) approach. The advantage of PP approach is that it produces high breakdown robust estimators. With an underlying elliptically contoured distribution, PP estimators are consistent. See Li and Cheng (1985). The next section discusses the PP approach in detail. The last section of this chapter is focused on robustifying PCA method (RPCA) based on the robust covariance or correlation matrices, such as FMCD and RMVN.

3.2 ROBUST PROJECTION-PURSUIT PCA

Projection pursuit is a statistical technique for finding the most "interesting" low-dimensional projections of a high-dimensional point cloud by numerically maximizing a certain objective function called *projection index* (PI). See Huber (1985). The PCA actually can be looked as a PP-technique since it searches the directions that have maximum variances. The classical PCA uses the variance function as the projection index. It has a serious drawback since the variance function is zero breakdown. To robustify, a robust PP-technique (RPP) uses a robust scale instead of the variance as its PI. Projection-Pursuit techniques start from the initial definition of (3.1) with a robust scale.

Let $\boldsymbol{x}$ be a p-dimensional random vector and let $S(\cdot)$ be a robust scale. The robust principal components of $\boldsymbol{x}$, denoted by $S_i(\boldsymbol{x})$ and $\boldsymbol{\alpha}_i$, are defined by

$$\begin{aligned}
\boldsymbol{a}_1(\boldsymbol{x}) &= \underset{||\boldsymbol{a}||=1}{\arg\max} S(\boldsymbol{a}'\boldsymbol{x}), \\
S_1(\boldsymbol{x}) &= S(\boldsymbol{a}_1'\boldsymbol{x}), \\
\boldsymbol{a}_2(\boldsymbol{x}) &= \underset{||\boldsymbol{a}||=1,\boldsymbol{a}\perp\boldsymbol{a}_1}{\arg\max} \quad S(\boldsymbol{a}'\boldsymbol{x}) \\
S_2(\boldsymbol{x}) &= S(\boldsymbol{a}_2'\boldsymbol{x}), \\
&\vdots \\
\boldsymbol{a}_p(\boldsymbol{x}) &= \underset{||\boldsymbol{a}||=1,\boldsymbol{a}\perp\boldsymbol{a}_1,\cdots,\boldsymbol{a}_{p-1}}{\arg\max} \quad S(\boldsymbol{a}'\boldsymbol{x}) \\
S_p(\boldsymbol{x}) &= S(\boldsymbol{a}_p'\boldsymbol{x}),
\end{aligned}$$

The robust covariance matrix, $\boldsymbol{C}(\boldsymbol{x})$ is then defined by

$$\boldsymbol{C}(\boldsymbol{x}) = \sum_{i=1}^{p} S_i(\boldsymbol{x})\boldsymbol{a}_i\boldsymbol{a}_i' \tag{3.4}$$

In general, RPP estimators are not affine equivariant. However, they are rotationally equivariant. That is, for any orthogonal matrix P,

$$S_i(P\boldsymbol{x}) = S_i(\boldsymbol{x}), \qquad \boldsymbol{a}_i(P\boldsymbol{x}) = P\boldsymbol{a}_i(\boldsymbol{x}), \quad \text{and} \quad \mathrm{Var}(P\boldsymbol{x}) = P\,\mathrm{Var}(\boldsymbol{x})P'. \tag{3.5}$$

The proof is quite straightforward from the definition of RPP. Since P is orthogonal, $||P\boldsymbol{a}|| = ||\boldsymbol{a}||$ and $\boldsymbol{a}'P = (P'\boldsymbol{a})' = (P\boldsymbol{a})'$. By definition of RPP,

$$S_1(\boldsymbol{x}) = \max_{||\boldsymbol{a}||=1} S(\boldsymbol{a}'\boldsymbol{x})$$

and

$$S_1(P\boldsymbol{x}) = \max_{||\boldsymbol{a}||=1} S(\boldsymbol{a}'P\boldsymbol{x}) = \max_{||P\boldsymbol{a}||=1} S((P\boldsymbol{a})'\boldsymbol{x}) = S_1(\boldsymbol{x}).$$

It follows that

$$\boldsymbol{a}_1(P\boldsymbol{x}) = \arg\max_{||\boldsymbol{a}||=1} S(\boldsymbol{a}'P\boldsymbol{x}) = P\boldsymbol{a}_1(\boldsymbol{x}),$$

and

$$\begin{aligned}
\boldsymbol{C}(P\boldsymbol{x}) &= \sum_{i=1}^{p} S_i(P\boldsymbol{x})\boldsymbol{a}_i(P\boldsymbol{x})\big(\boldsymbol{a}_i(P\boldsymbol{x})\big)' \\
&= \sum_{i=1}^{p} S_i(\boldsymbol{x})(P\boldsymbol{a}_i)(P\boldsymbol{a}_i)' \\
&= P\left[\sum_{i=1}^{p} S_i(\boldsymbol{x})\boldsymbol{a}_i\boldsymbol{a}_i'\right]P' \\
&= P\boldsymbol{C}(\boldsymbol{x})P'.
\end{aligned}$$

Li and Chen (1985) showed that $\boldsymbol{C}(\boldsymbol{x})$ is affine equivariant when $\boldsymbol{x} \sim EC(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The result implies the asymptotic unbiasedness of $\boldsymbol{C}(\boldsymbol{x})$ in the sense of

$$\boldsymbol{C}_\infty(\boldsymbol{x}) = s\boldsymbol{\Sigma}$$

where $\boldsymbol{C}_\infty(\boldsymbol{x})$ is the asymptotic value of $\boldsymbol{C}(\boldsymbol{x})$ and s is a constant. Moreover, because $\boldsymbol{C}(\boldsymbol{x})$ is defined to be $\sum_{i=1}^{p} S_i(\boldsymbol{x})\boldsymbol{a}_i\boldsymbol{a}_i'$ and $|\boldsymbol{a}_i| = 1$ for all $1 \leq i \leq p$, $\boldsymbol{C}(\boldsymbol{x})$ has the same breakdown value as $S(\cdot)$. The RPP estimator will be high breakdown if a high breakdown robust scale $S(\cdot)$ is used. Li and Chen (1985) used a HB Huber M-estimator $S(\cdot)$ for their Monte Carlo simulation. However, the M-estimator is not practical when the dimension p is not small. Croux, Filzmoser, Oliveira (2007) discussed an algorithm using normalized median absolute deviation (NMAD) as the robust scale

$$S(x) = \text{NMAD} = 1.48\text{Med}\big(|x - \text{Med}(x)|\big)$$

where Med() is the median function. This PCA estimator is also impractical to compute.

3.3 RPCA BASED ON ROBUST ESTIMATORS

The RPCA method performs the classical principal component analysis on the RMVN subset, using either the sample covariance matrix $\boldsymbol{C}_U = \boldsymbol{S}_U$ or the sample correlation matrix $\boldsymbol{R}_U$ applied to the RMVN subset. Under (E1), $\boldsymbol{C}_U$ is a $\sqrt{n}$ consistent outlier resistant estimator of $c\boldsymbol{\Sigma} = d\text{Cov}(\boldsymbol{X})$ where $c > 0$ and $d > 0$ are some constants. For the sample correlation matrix, $\boldsymbol{R}_U = \boldsymbol{D}_U^{-1/2}\boldsymbol{S}_U\boldsymbol{D}_U^{-1/2}$, where

$$\underset{(p\times p)}{\boldsymbol{D}_U^{-1/2}} = \begin{bmatrix} \frac{1}{\sqrt{s_{11}}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sqrt{s_{22}}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sqrt{s_{pp}}} \end{bmatrix}$$

and s_{ii} are diagonal entries of $\boldsymbol{S}_U$. $\boldsymbol{S}_U \xrightarrow{P} d\text{Cov}(\boldsymbol{X})$ implies that $\boldsymbol{R}_U$ converges to population correlation matrix in probability. Therefore the sample correlation matrix $\boldsymbol{R}_U$ is a $\sqrt{n}$ consistent estimator of the population correlation matrix.

Theorem 5. *Under (E1), the correlation of the eigenvalues computed from the classical PCA and RPCA converges to 1 in probability.*

Proof. Let $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ denote the population covariance matrix, $\text{Cov}(\boldsymbol{X})$. Let $\boldsymbol{S}$ be the classical sample covariance matrix and $\boldsymbol{S}_U$ be the sample covariance matrix of RMVN subset. Under (E1),

$$\boldsymbol{S} \xrightarrow{P} \boldsymbol{\Sigma}_{\boldsymbol{x}}$$

and

$$\boldsymbol{S}_U \xrightarrow{P} d\boldsymbol{\Sigma}_{\boldsymbol{x}}.$$

It is a known fact that the eigenvalues are continuous functions of the dispersion estimator. Hence consistent estimators of dispersion give consistent estimators of the population eigenvalues. That is, the eigenvalues of $\boldsymbol{S}$ converges to the eigenvalues of $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ in probability whereas the eigenvalues of $\boldsymbol{S}_U$ converges to the eigenvalues of d$\boldsymbol{\Sigma}_{\boldsymbol{x}}$ in probability. See Eaton and Tyler (1991) and Bhatia, Elsner and Krause (1990). If

$$\boldsymbol{\Sigma}_{\boldsymbol{x}}\ \boldsymbol{e} = \lambda \boldsymbol{e},$$

then

$$(d\boldsymbol{\Sigma}_{\boldsymbol{x}})\ \boldsymbol{e} = (d\lambda)\ \boldsymbol{e}.$$

Hence the population eigenvalues of $\Sigma_{\boldsymbol{x}}$ and $d\boldsymbol{\Sigma}_{\boldsymbol{x}}$ differ by a positive multiple d. If $\boldsymbol{\Lambda}' = [\lambda_1, \lambda_2, \cdots, \lambda_p]$ is a vector of eigenvalues of $\boldsymbol{\Sigma}_{\boldsymbol{x}}$, then $d\boldsymbol{\Lambda}'$ is a vector of eigenvalues of $d\boldsymbol{\Sigma}_{\boldsymbol{x}}$. Eigenvalues of $\boldsymbol{S} \xrightarrow{P} \boldsymbol{\Lambda}'$ and eigenvalues of $\boldsymbol{S}_U \xrightarrow{P} d\boldsymbol{\Lambda}'$. Therefore, correlation of eigenvalues computed from PCA and RPCA converges to $\text{Corr}(\boldsymbol{\Lambda}', d\boldsymbol{\Lambda}') = 1$. The proof for $\boldsymbol{R}$ and $\boldsymbol{R}_U$ is similar. □

For principal components, a scree plot is a plot of component number versus eigenvalue, and often there is a sharp bend in the plot when the components are no longer important. See Cattell (1966). The above theorem suggests making the robust scree plot and the classical scree plot.

Example 4. Buxton (1920) gives various measurements on 87 men including *height*, *head length*, *nasal height*, *bigonal breadth* and *cephalic index*. Five *heights* were recorded to be about 19mm with the true heights recorded under head length. Performing a classical principal components analysis on these five variables using the covariance matrix resulted in a first principal component that was created by the outliers. See Figure 3.1 where the second principal component is plotted versus the first. Significantly affected by outliers, Figure 3.1 provides false information regarding the correlation between first two principal components. One should expect to see a random scatter plot since principal components are supposed to be uncorrelated. The robust PCA, or the classical PCA performed after the outliers are removed, resulted in a first principal component that was approximately $-$ *height* with $\hat{\boldsymbol{e}}_1 \approx (-1.000, 0.002, -0.023, -0.002, -0.009)^T$ while the second robust principal component was based on the eigenvector $\hat{\boldsymbol{e}}_2 \approx (-0.005, 0.848, -0.054, - 0.048, 0.525)^T$. The plot of the first two robust principal components, with the outliers deleted, is shown in Figure 3.2. These two components explain about 86% of the variance.

Figure 3.3 is a classical scree plot and Figure 3.4 is a robust scree plot with five principal components. Both of them show the fifth eigenvalue is relatively small. It appears that four principal components should be used in order to summarize the total sample variance effectively .

The eigenvectors are not continuous functions of the dispersion estimator, and the sample size may need to be massive before the robust and classical eigenvectors or principal components have high absolute correlation. In the software, sign changes

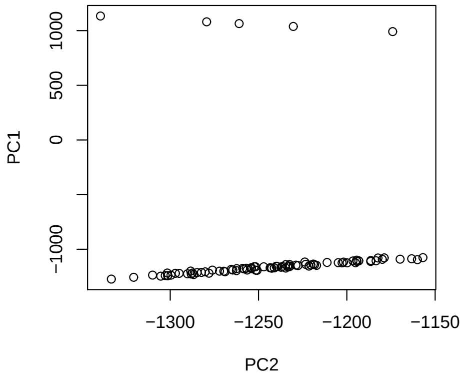

Figure 3.1. First Two Principal Components for Buxton data

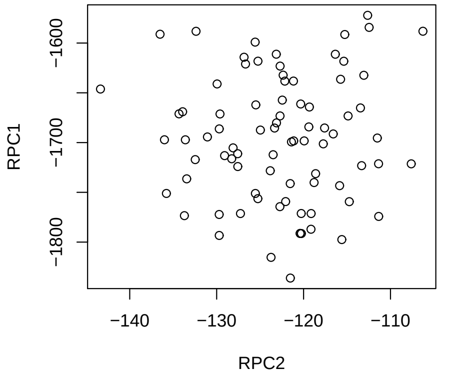

Figure 3.2. First Two Robust Principal Components with Outliers Omitted

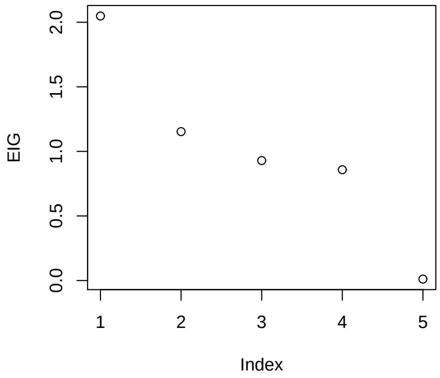

Figure 3.3. Scree Plot

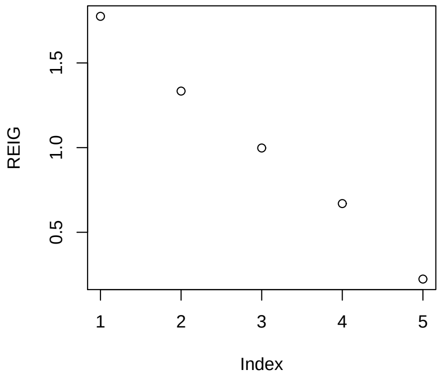

Figure 3.4. Robust Scree Plot

in the eigenvectors are common, since $\boldsymbol{\Sigma_x} \, \boldsymbol{e} = \lambda \boldsymbol{e}$ implies that $\boldsymbol{\Sigma_x} \, (-\boldsymbol{e}) = \lambda(-\boldsymbol{e})$.

The literature for robust PCA is large, but the "high breakdown" methods are impractical or not backed by theory. Some of these methods may be useful as outlier diagnostics. Spherical principal components is a bounded influence approach suggested by Locantore, Marron, Simpson, Tripoli, Zhang and Cohen (1999). Boente and Fraiman (1999) claim that basis of the eigenvectors is consistently estimated by spherical principal components for elliptically contoured distributions. Also see Maronna, Martin and Yohai (2006, pp. 212-213).

3.4 SIMULATION

In simulations for principal component analysis, FCH, RMVN, OGK and FAST-MCD seem to estimate $c\boldsymbol{\Sigma_x}$ if $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{\mu}$ where $\boldsymbol{z} = (z_1, ..., z_p)^T$ and the z_i are iid from a continuous distribution with variance σ^2. Here $\boldsymbol{\Sigma_x} = \text{Cov}(\boldsymbol{x}) = \sigma^2 \boldsymbol{A}\boldsymbol{A}^T$. The bias for the MB estimator seemed to be small. It is known that affine equivariant estimators give unbiased estimators of $c\boldsymbol{\Sigma_x}$ if the distribution of z_i is also symmetric. DGK and FAST-MCD are affine equivariant. FCH and RMVN are asymptotically equivalent to a scaled DGK estimator. But in the simulations the results also held for skewed distributions.

The simulations used 1000 runs where $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{z}$ and $\boldsymbol{z} \sim N_p(\boldsymbol{0}, \boldsymbol{I}_p)$, $\boldsymbol{z} \sim LN(\boldsymbol{0}, \boldsymbol{I}_p)$ where the marginals are iid lognormal(0,1), or $\boldsymbol{z} \sim MVT_p(1)$, a multivariate t distribution with 1 degree of freedom so the marginals are iid Cauchy(0,1). The choice $\boldsymbol{A} = diag(\sqrt{1}, ..., \sqrt{p})$ results in $\boldsymbol{\Sigma} = diag(1, ..., p)$. Note that the population eigenvalues will be proportional to $(p, p-1, ..., 1)^T$ and the population "variance explained" by the ith principal component is $\lambda_i / \sum_{j=1}^{p} \lambda_j = 2(p+1-i)/[p(p+1)]$. For $p = 4$, these numbers are 0.4, 0.3 and 0.2 for the first three principal components. If the "correlation" option is used, then the population "correlation matrix" is the identity matrix $\boldsymbol{I}_p$, the ith population eigenvalue is proportional to $1/p$ and the population “variance explained” by the ith principal component is $1/p$.

Table 3.2 shows the mean “variance explained” along with the standard deviations for the first three principal components when $p = 4$. Also a_i and p_i are the average absolute value of the correlation between the ith eigenvectors or the ith principal components of the classical and robust methods. Two rows were used for each “n–data type” combination. The a_i are shown in the top row while the p_i are in the lower row. The values of a_i and p_i were similar. The standard deviations were slightly smaller for the classical PCA for normal data. The classical method failed to estimate (0.4,0.3,0.2) for the Cauchy data. For the lognormal data, RPCA gave better estimates, and the p_i were not high except for $n = 10000$. More PCA simulation results for $p \leq 20$ are available in Appendix A.

Table 3.2. Variance Explained by PCA and RPCA, $p = 4$

n	type	M/S	vexpl	rvexpl	$a_1(p_1)$	$a_2(p_2)$	$a_3(p_3)$
40	N	M	0.445,0.289,0.178	0.472,0.286,0.166	0.895	0.821	0.825
		S	0.050,0.037,0.032	0.062,0.043,0.037	0.912	0.813	0.804
100	N	M	0.419,0.295,0.191	0.425,0.293,0.189	0.952	0.926	0.963
		S	0.033,0.030,0.024	0.040,0.032,0.027	0.956	0.923	0.953
400	N	M	0.404,0.298,0.198	0.406,0.298,0.198	0.994	0.991	0.996
		S	0.019,0.017,0.014	0.021,0.019,0.015	0.995	0.990	0.994
40	C	M	0.765,0.159,0.056	0.514,0.275,0.147	0.563	0.519	0.511
		S	0.165,0.112,0.051	0.078,0.055,0.040	0.776	0.383	0.239
100	C	M	0.762,0.156,0.060	0.455,0.286,0.173	0.585	0.527	0.528
		S	0.173,0.112,0.055	0.054,0.041,0.034	0.797	0.377	0.269
400	C	M	0.756,0.162,0.060	0.413,0.296,0.194	0.608	0.562	0.575
		S	0.172,0.113,0.054	0.030,0.025,0.022	0.796	0.397	0.308
40	L	M	0.539,0.256,0.139	0.521,0.268,0.146	0.610	0.509	0.530
		S	0.127,0.075,0.054	0.099,0.061,0.047	0.643	0.439	0.398
100	L	M	0.482,0.270,0.165	0.459,0.279,0.172	0.647	0.555	0.566
		S	0.180,0.063,0.052	0.077,0.047,0.041	0.654	0.492	0.474
400	L	M	0.437,0.282,0.185	0.416,0.290,0.194	0.748	0.639	0.739
		S	0.080,0.048,0.044	0.049,0.035,0.033	0.727	0.594	0.690
10000	L	M	0.400,0.301,0.200	0.402,0.300,0.199	0.982	0.967	0.991
		S	0.027,0.023,0.018	0.013,0.011,0.009	0.976	0.967	0.989

To compare affine equivariant and non-equivariant estimators, Maronna and

Zamar (2002) suggest using $\boldsymbol{A}_{i,i} = 1$ and $\boldsymbol{A}_{i,j} = \rho$ for $i \neq j$ and $\rho = 0, 0.5, 0.7, 0.9,$ and 0.99. Then $\boldsymbol{\Sigma} = \boldsymbol{A}^2$. If ρ is high, or if p is high and $\rho \geq 0.5$, then the data are concentrated about the line with direction $\mathbf{1} = (1, ..., 1)^T$. For $p = 50$ and $\rho = 0.99$, the population variance explained by the first principal component is 0.999998. If the "correlation" option is used, then there is still one extremely dominant principal component unless both p and ρ are small.

Table 3.3 shows the mean "variance explained" along with the standard deviations multiplied by 10^7 for the first principal component. The a_1 value is given but p_1 was always 1.0 to many decimal places even with Cauchy data. Hence the eigenvectors from the robust and classical methods could have low absolute correlation, but the data was so tightly clustered that the first principal components from the robust and classical methods had absolute correlation near 1.

Table 3.3. Variance Explained by PCA and RPCA, SSD = 10^7 SD, $p = 50$

n	type	vexpl	SSD	rvexpl	SSD	a_1
200	N	0.999998	1.958	0.999998	2.867	0.687
1000	N	0.999998	0.917	0.999998	0.971	0.944
1000	C	0.999996	161.3	0.999998	1.482	0.112
1000	L	0.999998	0.919	0.999998	1.508	0.175

CHAPTER 4

HOTELLING'S $\boldsymbol{T^2}$ TEST DIAGNOSTIC

In Multivariate Analysis, the Hotelling's T^2 is a useful tool to make inference about the center of multivariate normal data. The Hotelling's T^2 test is used to test $H_0 : \boldsymbol{\mu} = \boldsymbol{\mu}_0$ versus $H_A : \boldsymbol{\mu} \neq \boldsymbol{\mu}_0$. The test rejects H_0 if

$$T^2 = n(\overline{\boldsymbol{x}} - \boldsymbol{\mu}_0)^T \boldsymbol{S}^{-1} (\overline{\boldsymbol{x}} - \boldsymbol{\mu}_0) > \frac{(n-1)p}{n-p} F_{p,n-p,1-\alpha}$$

and the data are iid from a MVN distribution with a nonsingular covariance matrix. Note that

$$\frac{(n-1)p}{n-p} F_{p,n-p,1-\alpha} \rightarrow \chi^2_{p,1-\alpha}$$

as sample size $n \rightarrow \infty$. So for a large sample, the test can be

$$T^2 = n(\overline{\boldsymbol{x}} - \boldsymbol{\mu}_0)^T \boldsymbol{S}^{-1} (\overline{\boldsymbol{x}} - \boldsymbol{\mu}_0) > \chi^2_{p,1-\alpha}.$$

When using the classical location and dispersion estimator, the test can be adversely affected by outliers. Therefore a robust estimator should be considered to replace the classical one for a Hotelling's T^2 Test.

If a location estimator T satisfies

$$\sqrt{n}(T - \boldsymbol{\mu}) \xrightarrow{d} N_p(\boldsymbol{0}, \boldsymbol{D}),$$

then a competing test rejects H_0 if

$$T^2 = n(T - \boldsymbol{\mu}_0)^T \hat{\boldsymbol{D}}^{-1} (T - \boldsymbol{\mu}_0) > \frac{(n-1)p}{n-p} F_{p,n-p,1-\alpha} \rightarrow \chi^2_{p,1-\alpha}$$

if $\hat{\boldsymbol{D}}$ is a consistent estimator of $\boldsymbol{D}$.

Now the RMVN estimator is asymptotically equivalent to a scaled DGK estimator that uses 5 concentration steps from the start and two "reweight for efficiency" steps.

We conjecture that

$$\sqrt{n}(T_{RMVN} - \boldsymbol{\mu}) \xrightarrow{D} N_p(\mathbf{0}, \tau_p \boldsymbol{\Sigma})$$

for a wide variety of EC distributions where τ_p depends on both p and the underlying distribution. Since the test is based on a conjecture, it is ad hoc, and should be used as an outlier diagnostic rather than for inference. Willems, Pison, Rousseeuw, and Van Aelst (2002) use similar reasoning for the MCD estimator, but their actual statistic uses the FMCD estimator which is not high breakdown and may not even be consistent.

For contaminated MVN data, simulations suggest that τ_p is close to 1 and gets closer as p increases. The ad hoc test rejects H_0 if

$$T_R^2/f_{n,p} = n(T_{RMVN} - \boldsymbol{\mu}_0)^T \hat{\boldsymbol{C}}_{RMVN}^{-1}(T_{RMVN} - \boldsymbol{\mu}_0)/f_{n,p} > \frac{(n-1)p}{n-p} F_{p,n-p,1-\alpha}$$

where $f_{n,p} = 1.04 + 0.12/p + 74/n$ gave fair results in the simulations for $n \geq 15p$ and $2 \leq p \leq \infty$.

As sample size gets larger, the $\chi^2_{p,1-\alpha}$ is often recommended for robust Hotelling's T^2 Test. One reason is the distribution $\frac{(n-1)p}{n-p} F_{p,n-p,1-\alpha}$ converges to the distribution $\chi^2_{p,1-\alpha}$ as $n \to \infty$.

For the Hotelling's T_H^2 simulation, the data is $N_p(\delta \mathbf{1}, diag(1, 2, ..., p))$ where $H_0 : \boldsymbol{\mu} = \mathbf{0}$ is being tested with 5000 runs at a nominal level of 0.05. In Table 4.1, $\delta = 0$ so H_0 is true, while hcv and rhcv are the proportion of rejections by the T_H^2

test and by the ad hoc robust test. Sample sizes are $n = 15p, 20p$ and $30p$. The robust test is not recommended for $n < 15p$ and appears to be conservative except when $n = 15p$ and $75 \leq p \leq 100$. The nominal level is 0.05.

If $\delta > 0$, then H_0 is false and the proportion of rejections estimates the power of the test. Table 4.2 shows that T^2_H has more power than the robust test, but suggests that the power of both tests rapidly increases to one as δ increases. The robust T^2_R statistic tends not to be as inflated as T^2_H when outliers are present, as can be demonstrated with the `rhotsim` program referenced below. Ideally software users would make a DD plot and other checks on the model, but users of statistical software too often fail to make such checks. Since both statistics are easily computed, if $n \geq 15n$ software could produce a warning if the two statistics differ.

In www.math.siu.edu/olive/rpack.txt, there is a function rhotsim. This R function simulates an ad hod robust Hotelling's T^2 test. Need $p > 1$. Outliers $= 0$ for no outliers and $X \sim N(0, diag(1, ..., p))$; Outliers $= 1$ for outliers a tight cluster at major axis $(0, ..., 0, pm)'$; Outliers $= 2$ for outliers a tight cluster at minor axis $(pm, 0, ..., 0)'$; Outliers $= 3$ for outliers $X \sim N((pm, ..., pm)', diag(1, ..., p))$; Outliers $= 4$ for outliers $X[i, p] = pm$; Outliers $= 5$ for outliers $X[i, 1] = pm$. Power can be estimated by increasing delta so $\boldsymbol{\mu} = \boldsymbol{\delta}(1, ..., 1)$ and $\boldsymbol{\mu}_0 = 0 * \boldsymbol{\mu}$. For outliers=0, want hquant and rquant approx 1.

Simulations were done in *R*. The `MASS` library was used to compute FMCD and the `robustbase` library was used to compute OGK. Programs are in the collection of functions *rpack.txt* at (www.math.siu.edu/olive/ol-bookp.htm). Function `covrmvn` computes the FCH, RMVN and MB estimators while `covfch` computes the FCH, RFCH and MB estimators. More Hotelling simulation results are available in Appendix B.

Table 4.1. Hotelling simulation

p	n=15p	hcv	rhcv	n=20p	hcv	rhcv	n=30p	hcv	rhcv
2	30	0.0502	0.0516	40	0.0498	0.0624	60	0.0540	0.0382
5	75	0.0500	0.0456	100	0.0474	0.0250	150	0.0542	0.0310
10	150	0.0476	0.0300	200	0.0516	0.0304	300	0.0498	0.0286
15	225	0.0474	0.0318	300	0.0506	0.0308	450	0.0492	0.0320
20	300	0.0540	0.0368	400	0.0548	0.0314	600	0.0520	0.0354
25	375	0.0444	0.0334	500	0.0462	0.0296	750	0.0456	0.0288
30	450	0.0472	0.0324	600	0.0516	0.0358	900	0.0484	0.0342
35	525	0.0490	0.0384	700	0.0522	0.0358	1050	0.0502	0.0374
40	600	0.0534	0.0440	800	0.0486	0.0354	1200	0.0526	0.0336
45	675	0.0406	0.0390	900	0.0544	0.0390	1350	0.0512	0.0366
50	750	0.0498	0.0430	1000	0.0522	0.0394	1500	0.0512	0.0364
55	825	0.0504	0.0502	1100	0.0496	0.0392	1650	0.0510	0.0374
60	900	0.0482	0.0514	1200	0.0488	0.0404	1800	0.0474	0.0376
65	975	0.0568	0.0602	1300	0.0524	0.0414	1950	0.0548	0.0410
70	1050	0.0462	0.0530	1400	0.0558	0.0432	2100	0.0522	0.0424
75	1125	0.0474	0.0632	1500	0.0502	0.0486	2250	0.0490	0.0370
80	1200	0.0524	0.0620	1600	0.0524	0.0432	2400	0.0468	0.0356
85	1275	0.0482	0.0758	1700	0.0496	0.0456	2550	0.0520	0.0404
90	1350	0.0504	0.0746	1800	0.0484	0.0454	2700	0.0484	0.0398
95	1425	0.0524	0.0892	1900	0.0472	0.0506	2850	0.0538	0.0424
100	1500	0.0554	0.0808	2000	0.0452	0.0506	3000	0.0488	0.0392

Table 4.2. Hotelling power simulation

p	n	hcv	rhcv	δ	n	hcv	rhcv	δ	n	hcv	rhcv	δ
2	30	0.380	0.150	0.30	40	0.367	0.160	0.25	60	0.363	0.180	0.20
2	30	0.615	0.260	0.40	40	0.640	0.314	0.35	60	0.700	0.430	0.30
2	30	0.830	0.422	0.50	40	0.864	0.516	0.45	60	0.921	0.706	0.40
5	75	0.459	0.245	0.20	100	0.366	0.184	0.15	150	0.333	0.208	0.12
5	75	0.682	0.416	0.25	100	0.599	0.368	0.20	150	0.577	0.394	0.16
5	75	0.840	0.588	0.30	100	0.816	0.587	0.30	150	0.860	0.708	0.40
10	150	0.221	0.113	0.10	200	0.312	0.182	0.10	300	0.469	0.340	0.10
10	150	0.621	0.400	0.17	200	0.655	0.467	0.15	300	0.647	0.504	0.12
10	150	0.888	0.729	0.22	200	0.848	0.692	0.18	300	0.872	0.767	0.15
15	225	0.314	0.188	0.10	300	0.442	0.294	0.10	450	0.317	0.228	0.07
15	225	0.714	0.543	0.15	300	0.623	0.449	0.12	450	0.648	0.522	0.10
15	225	0.881	0.738	0.18	300	0.858	0.755	0.15	450	0.853	0.762	0.12
20	300	0.408	0.276	0.10	400	0.341	0.230	0.08	600	0.291	0.216	0.06
20	300	0.691	0.525	0.13	400	0.674	0.534	0.11	600	0.554	0.433	0.08
20	300	0.935	0.852	0.17	400	0.858	0.742	0.13	600	0.790	0.701	0.10
25	375	0.304	0.214	0.08	500	0.434	0.319	0.08	750	0.354	0.266	0.06
25	375	0.728	0.580	0.12	500	0.676	0.531	0.10	750	0.660	0.556	0.08
25	375	0.926	0.837	0.15	500	0.868	0.771	0.12	750	0.887	0.815	0.10
30	450	0.374	0.264	0.08	600	0.395	0.290	0.07	900	0.290	0.217	0.05
30	450	0.602	0.467	0.10	600	0.639	0.517	0.09	900	0.743	0.642	0.08
30	450	0.883	0.763	0.13	600	0.867	0.770	0.11	900	0.876	0.808	0.09

CHAPTER 5

ROBUST CANONICAL CORRELATION ANALYSIS

5.1 INTRODUCTION

Canonical correlation analysis (CCA) is a multivariate statistical method to identify and quantify the association between two sets of variables. It focuses on the correlation between a linear combination of the variables in one set and a linear combination of the variables in another set. First, a pair of linear combinations is determined by maximizing the correlation. Next, a pair of linear combinations uncorrelated to previously selected pair is determined by maximizing the correlation, and so on. The pairs of combinations are called the *canonical variables* (*canonical variates*), and their correlations are called *canonical correlations*.

Denote the first set of variables by the p-dimensional variable $\boldsymbol{x}$ and the second set of variables by the q-dimensional variable $\boldsymbol{y}$.

$$\boldsymbol{x} = [X_1, X_2, \cdots X_p]' \quad \text{and} \quad \boldsymbol{y} = [Y_1, Y_2, \cdots Y_q]'.$$

Without loss of generality, assume $p \leq q$. For the random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, let

$$\mathrm{E}(\boldsymbol{x}) = \boldsymbol{\mu}_1 \quad \text{and} \quad \mathrm{E}(\boldsymbol{y}) = \boldsymbol{\mu}_2,$$

$$\mathrm{Cov}(\boldsymbol{x}) = \boldsymbol{\Sigma}_{11} \quad \text{and} \quad \mathrm{Cov}(\boldsymbol{y}) = \boldsymbol{\Sigma}_{22},$$

$$\mathrm{Cov}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\Sigma}_{12} = \boldsymbol{\Sigma}_{21}'.$$

Considering $\boldsymbol{x}$ and $\boldsymbol{y}$ jointly in a random vector $\boldsymbol{W}$,

$$\underset{((p+q)\times 1)}{\boldsymbol{W}} = \begin{bmatrix} \boldsymbol{x} \\ \\ \boldsymbol{y} \end{bmatrix},$$

with mean vector

$$\underset{((p+q)\times 1)}{\boldsymbol{\mu}} = \mathrm{E}(\boldsymbol{W}) = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \\ \boldsymbol{\mu}_2 \end{bmatrix}$$

and covariance matrix

$$\underset{((p+q)\times(p+q))}{\boldsymbol{\Sigma}} = \mathrm{E}\big[(\boldsymbol{W}-\boldsymbol{\mu})(\boldsymbol{W}-\boldsymbol{\mu})'\big] = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix}. \tag{5.1}$$

Then the canonical coefficients of the first pair of linear combination is determined by

$$(\boldsymbol{\alpha}_1, \boldsymbol{\beta}_1) = \underset{\boldsymbol{a},\boldsymbol{b}}{\arg\max}\, \mathrm{Corr}(\boldsymbol{a}'\boldsymbol{x}, \boldsymbol{b}'\boldsymbol{y}) \tag{5.2}$$

with the restriction $\mathrm{Cov}(\boldsymbol{a}'\boldsymbol{x}) = 1$, $\mathrm{Cov}(\boldsymbol{b}'\boldsymbol{y}) = 1$ and $\mathrm{Cov}(\boldsymbol{a}'\boldsymbol{x}, \boldsymbol{b}'\boldsymbol{y}) = 0$. So the first pair of canonical variates is the pair of the linear combinations

$$U_1 = \boldsymbol{\alpha}_1'\boldsymbol{x} \quad \text{and} \quad V_1 = \boldsymbol{\beta}_1'\boldsymbol{y}$$

where $\mathrm{Cov}(U_1) = 1$, $\mathrm{Cov}(V_1) = 1$, and $\mathrm{Cov}(U_1, V_1) = 0$. Higher order kth canonical vectors is then recursively defined by

$$(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k) = \underset{\boldsymbol{a},\boldsymbol{b}}{\arg\max}\, \mathrm{Corr}(\boldsymbol{a}'\boldsymbol{x}, \boldsymbol{b}'\boldsymbol{y}) \tag{5.3}$$

with the restriction $\text{Cov}(\boldsymbol{a}'\boldsymbol{x}) = 1$, $\text{Cov}(\boldsymbol{b}'\boldsymbol{y}) = 1$, $\text{Cov}(\boldsymbol{a}'\boldsymbol{x}, \boldsymbol{b}'\boldsymbol{y}) = 0$ and $(\boldsymbol{a}'\boldsymbol{x}, \boldsymbol{b}'\boldsymbol{y})$ is uncorrelated with all previous selected canonical variates (U_i, V_i) where $1 \leq i \leq k-1$. The canonical correlation ρ_k between the canonical variates of the kth pair is

$$\rho_k = \text{Corr}(U_k, V_k).$$

Johnson and Wichern (1998, Chapter 10) gives a simple solution to compute the canonical variates. The kth pair of canonical variates, $k = 1, 2, \cdots, p$ can be computed as

$$U_k = \boldsymbol{e}_k' \boldsymbol{\Sigma}_{11}^{-1/2} \boldsymbol{x} \qquad V_k = \boldsymbol{f}_k' \boldsymbol{\Sigma}_{22}^{-1/2} \boldsymbol{y} \tag{5.4}$$

and

$$\text{Corr}(U_k, V_k) = \rho_k$$

where $\rho_1^2 \geq \rho_2^2 \geq \cdots \geq \rho_p^2$ are the eigenvalues of the matrix

$$\boldsymbol{\Sigma}_{11}^{-1/2} \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1} \boldsymbol{\Sigma}_{21} \boldsymbol{\Sigma}_{11}^{-1/2}$$

with associated eigenvectors $\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_p$. Moreover, $\rho_1 \geq \rho_2 \geq \cdots \geq \rho_p$ are also the p largest eigenvalues of

$$\boldsymbol{\Sigma}_{22}^{-1/2} \boldsymbol{\Sigma}_{21} \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1/2}$$

with associated eigenvectors $\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_p$.

When the original variables to be studied by CCA have quite different measure scales or standard deviations, they usually will be standardized for better analysis and interpretation before computing the canonical variates. Let $\sigma_{ii} = \text{Cov}(X_i)$ and $\nu_{ii} = \text{Cov}(Y_i)$. Further let $\boldsymbol{V}_{11} = \text{diag}(\sigma_{11}, \sigma_{22}, \cdots, \sigma_{pp})$ and

$\boldsymbol{V}_{22} = \text{diag}(\nu_{11}, \nu_{22}, \cdots, \nu_{qq})$. Then the standardized random vectors are

$$\boldsymbol{z}_x = \boldsymbol{V}_{11}^{-1/2}(\boldsymbol{x} - \boldsymbol{\mu}_x) \quad \text{and} \quad \boldsymbol{z}_y = \boldsymbol{V}_{22}^{-1/2}(\boldsymbol{y} - \boldsymbol{\mu}_y).$$

Consequently the canonical variates standardized vectors, $\boldsymbol{z}_x$ and $\boldsymbol{z}_y$ have the form

$$U_k^* = (\boldsymbol{\alpha}_k^*)'\boldsymbol{z}_x = \boldsymbol{e}_k^{'}\boldsymbol{\rho}_{11}^{-1/2}\boldsymbol{z}_x$$

and

$$V_k^* = (\boldsymbol{\beta}_k^*)'\boldsymbol{z}_y = \boldsymbol{f}_k^{'}\boldsymbol{\rho}_{22}^{-1/2}\boldsymbol{z}_y$$

where $\text{Cov}(\boldsymbol{z}_x) = \boldsymbol{\rho}_{11}$, $\text{Cov}(\boldsymbol{z}_y) = \boldsymbol{\rho}_{22}$, $\text{Cov}(\boldsymbol{z}_x, \boldsymbol{z}_y) = \boldsymbol{\rho}_{12} = \boldsymbol{\rho}_{21}^{'}$, and $\boldsymbol{e}_k$ and $\boldsymbol{f}_k$ are the eigenvectors of $\boldsymbol{\rho}_{11}^{-1/2}\boldsymbol{\rho}_{12}\boldsymbol{\rho}_{22}^{-1}\boldsymbol{\rho}_{21}\boldsymbol{\rho}_{11}^{-1/2}$ and $\boldsymbol{\rho}_{22}^{-1/2}\boldsymbol{\rho}_{21}\boldsymbol{\rho}_{11}^{-1}\boldsymbol{\rho}_{12}\boldsymbol{\rho}_{22}^{-1/2}$ respectively. The canonical correlations is given by

$$\text{Corr}(U_k^*, V_k^*) = \rho_k^*,$$

where $\rho_1^* \geq \rho_2^* \geq \cdots \geq \rho_p^*$ are the eigenvalues of the matrices of both $\boldsymbol{\rho}_{11}^{-1/2}\boldsymbol{\rho}_{12}\boldsymbol{\rho}_{22}^{-1}\boldsymbol{\rho}_{21}\boldsymbol{\rho}_{11}^{-1/2}$ and $\boldsymbol{\rho}_{22}^{-1/2}\boldsymbol{\rho}_{21}\boldsymbol{\rho}_{11}^{-1}\boldsymbol{\rho}_{12}\boldsymbol{\rho}_{22}^{-1/2}$.

Note that in accordance with the definition of the canonical variate,

$$\begin{aligned}(\boldsymbol{\alpha}_k^*, \boldsymbol{\beta}_k^*) &= \underset{\boldsymbol{a}^*, \boldsymbol{b}^*}{\arg\max}\, \text{Corr}[(\boldsymbol{a}^*)'\boldsymbol{z}_x, (\boldsymbol{b}^*)'\boldsymbol{z}_y] \\ &= \underset{\boldsymbol{a}^*, \boldsymbol{b}^*}{\arg\max}\, \text{Corr}\big((\boldsymbol{a}'V_{11}^{-1/2})\boldsymbol{x}, (\boldsymbol{b}'V_{22}^{-1/2})\boldsymbol{y}\big) \\ &= \underset{\boldsymbol{a}, \boldsymbol{b}}{\arg\max}\, \text{Corr}(\boldsymbol{a}'\boldsymbol{x}, \boldsymbol{b}'\boldsymbol{y}) \\ &= (V_{11}^{-1/2}\boldsymbol{a}, V_{22}^{-1/2}\boldsymbol{b}). \qquad (5.5)\end{aligned}$$

Therefore, unlike the principal component analysis, CCA has an equivariance property since the canonical correlations are unchanged by the standardization. That is, $\rho_k \equiv \rho_k^*$ for all $1 \leq k \leq p$.

Canonical variates are generally artificial and have no physical meaning. They are latent variables analogous to factors obtained in factor analysis. They often are looked as subject-matter variables. If the original variables are standardized to have zero means and unit variances, then the standardized canonical coefficients are interpreted in a similar manner to standardized regression coefficients. Being increased by one for a standardized variable is the same as being increased by one standard deviation for the corresponding original variable.

Let $\underset{(p\times p)}{\boldsymbol{A}} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_p]'$ and $\underset{(p\times p)}{\boldsymbol{B}} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_p]'$ so that the vectors of canonical variates are

$$\underset{(p\times 1)}{\boldsymbol{U}} = \boldsymbol{Ax} \quad \text{and} \quad \underset{(q\times 1)}{\boldsymbol{V}} = \boldsymbol{By}.$$

From (5.4), $\boldsymbol{A} = \boldsymbol{E}'\boldsymbol{\Sigma}_{11}^{-1/2}$ and $\boldsymbol{B} = \boldsymbol{F}'\boldsymbol{\Sigma}_{22}^{-1/2}$ where $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_p]$ and $\boldsymbol{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_q]$. So

$$\mathrm{Cov}(\boldsymbol{U}) = \mathrm{Cov}(\boldsymbol{Ax}) = \boldsymbol{A\Sigma}_{11}\boldsymbol{A}' = \boldsymbol{E}'\boldsymbol{\Sigma}_{11}^{-1/2}\boldsymbol{\Sigma}_{11}\boldsymbol{\Sigma}_{11}^{-1/2}\boldsymbol{E} = \boldsymbol{I}.$$

Likewise,

$$\mathrm{Cov}(\boldsymbol{V}) = \mathrm{Cov}(\boldsymbol{By}) = \boldsymbol{I}.$$

Decompose $\boldsymbol{\Sigma}_{11}$ to get $\boldsymbol{\Sigma}_{11} = \boldsymbol{P}_1\boldsymbol{\Lambda}_1\boldsymbol{P}_1'$. It follows that

$$\boldsymbol{U} = \boldsymbol{Ax} = \boldsymbol{E}'\boldsymbol{\Sigma}_{11}^{-1/2}\boldsymbol{x} = \boldsymbol{E}'\boldsymbol{P}_1\boldsymbol{\Lambda}_1^{-1/2}\boldsymbol{P}_1'\boldsymbol{x}.$$

Hence, the canonical variates vector $\boldsymbol{U}$ can be geometrically interpreted as the three-step transformation as follows. A similar geometrical interpretation can be made to $\boldsymbol{V}$.

(i) A transformation from $\boldsymbol{x}$ to uncorrelated standardized principal components, $\boldsymbol{\Lambda}_1^{-1/2}\boldsymbol{P'}_1\boldsymbol{x}$;

(ii) an orthogonal rotation $\boldsymbol{P}_1$;

(iii) another orthogonal rotation $\boldsymbol{E}'$.

The canonical coefficients are estimated by using sample covariance matrix instead of population covariance matrix. Denote the data matrix $\boldsymbol{X} = [X_1, X_2, \cdots, X_p]$ and $\boldsymbol{Y} = [Y_1, Y_2, \cdots, Y_q]$. (5.4) becomes

$$\hat{U}_k = \hat{\boldsymbol{e}}_k' \boldsymbol{S}_{11}^{-1/2}\boldsymbol{X} \qquad \hat{V}_k = \hat{\boldsymbol{f}}_k' \boldsymbol{S}_{22}^{-1/2}\boldsymbol{Y} \tag{5.6}$$

where $\hat{\boldsymbol{e}}_k$, for $0 \leq k \leq p$, is an eigenvector of

$$\boldsymbol{S}_{11}^{-1/2}\boldsymbol{S}_{12}\boldsymbol{S}_{22}^{-1}\boldsymbol{S}_{21}\boldsymbol{S}_{11}^{-1/2}$$

and $\hat{\boldsymbol{f}}_k$, for $0 \leq k \leq p$, is an eigenvector of

$$\boldsymbol{S}_{22}^{-1/2}\boldsymbol{S}_{21}\boldsymbol{S}_{11}^{-1}\boldsymbol{S}_{12}\boldsymbol{S}_{22}^{-1/2}.$$

Eigenvalues $r_1, r_2, \cdots, r_p$ of $\boldsymbol{S}_{11}^{-1/2}\boldsymbol{S}_{12}\boldsymbol{S}_{22}^{-1}\boldsymbol{S}_{21}\boldsymbol{S}_{11}^{-1/2}$ are the sample canonical correlations. Muirhead and Waternaux (1980) shows that if the population canonical correlation efficients are distinct and the underlying population distribution has finite fourth order cumulant, then the limit joint distribution of $\sqrt{n}(r_i^2 - \rho_i^2)$, for $i = 1, \cdots, p$, is p-variate normal. In particular, if the data are drawn from an elliptical distribution with kurtosis 3κ, then the limiting joint distribution of

$$\mu_i = \sqrt{n}\frac{r_i^2 - \rho_i^2}{2\rho_i(1-\rho_i^2)}, \quad i = 1, \cdots, p$$

is $N(\mathbf{0}, (\kappa+1)\boldsymbol{I}_p)$. As a more special case, when the data are drawn from multivariate normal distribution ($\kappa = 0$), the u_i's are asymptotically iid with a standard normal distribution.

However, these asymptotic results are nonrobust. The outliers have great distorting effect on the classical sample covariance matrix since the eigenvalues and eigenvectors are very sensitive to the presence of outliers. Replacing the classical sample covariance matrix by a robust dispersion estimator, such as RMVN, and then computing the eigenvalues and eigenvectors regularly from the robust dispersion estimator is an approach not only intuitive but also effective for a robust CCA. In the last section of this chapter, a simulation will be implemented to compare the classical CCA and robust CCA based on Fast-MCD and RMVN dispersion estimators. The next section discusses the projection pursuit (PP) approach. The idea of the PP approach is to robustify the correlation measure in (5.2) rather than robustify the classical dispersion matrix.

5.2 ROBUST CCA USING PROJECTION PURSUIT

In section 3.2, one has learned that the PCA can be looked as a PP-technique since it searches for the directions that have maximum variances. The classical PCA PP-technique uses the variance function as a projection index and robust PCA uses a robust scale. A similar idea could be applied for canonical correlation analysis. CCA can also be seen as a PP-technique since it seeks for two directions $\boldsymbol{a}$ and $\boldsymbol{b}$ in which the correlation of two projections of the variables $\boldsymbol{x}$ and $\boldsymbol{y}$, corr($\boldsymbol{a}'\boldsymbol{x}$,$\boldsymbol{b}'\boldsymbol{y}$), is maximized. The correlation measure in this case is the projection index. The robust PP-technique substitutes the classical correlation measure with a robust estimator of the correlation called robust projection index (RPI). Derivation from a robust covariance matrix of two univariate variables is a common approach to obtain a RPI. RMVN, Fast-MCD, and M-estimator robust projection indices will be compared by Monte Carlo study in next section. Muirhead and Waternaux (1980) provided a limit distribution for classical CCA when the underlying population distribution has finite fourth moment. However, so far there is still no asymptotic theory of RPP available since it is very difficulty to work out the properties of the robust CCA estimator analytically. Only simulation studies are conducted to estimate those properties. Branco, Croux, Filzmoser, and Oliveira (2005) proposed an algorithm to perform projection pursuit CCA without the backup of any rigorous and consummate theories. The algorithm starts by estimating $\boldsymbol{\Sigma}$ using a robust estimator. Then $\boldsymbol{\Sigma}$ is partitioned as

$$\underset{(p+q)\times(p+q)}{\boldsymbol{\Sigma}} = \begin{bmatrix} \underset{(p\times p)}{\boldsymbol{\Sigma}_{11}} & \underset{(p\times q)}{\boldsymbol{\Sigma}_{12}} \\ \\ \underset{(q\times p)}{\boldsymbol{\Sigma}_{21}} & \underset{(q\times q)}{\boldsymbol{\Sigma}_{22}} \end{bmatrix}.$$

Performing a spectral decomposition of $\boldsymbol{\Sigma}_{11}$ and $\boldsymbol{\Sigma}_{22}$,

$$\Sigma_{11} = \boldsymbol{AMA'} \quad \text{and} \quad \boldsymbol{\Sigma}_{22} = \boldsymbol{BNB'},$$

where $\boldsymbol{M}$, $\boldsymbol{N}$ are diagonal and $\boldsymbol{A}$, $\boldsymbol{B}$ are orthogonal matrices. Transform the original data $\boldsymbol{x}$ and $\boldsymbol{y}$ into

$$(\boldsymbol{x}^*, \boldsymbol{y}^*) = \big(\boldsymbol{M}^{-1/2}\boldsymbol{A}'\boldsymbol{x}, \boldsymbol{N}^{-1/2}\boldsymbol{B}'\boldsymbol{y}\big).$$

Note that

$$\begin{aligned}
\underset{\boldsymbol{a}^*,\boldsymbol{b}^*}{\arg\max}\, PI[(\boldsymbol{a}^*)'\boldsymbol{x}^*, (\boldsymbol{b}^*)'\boldsymbol{y}^*] &= \underset{\boldsymbol{a}^*,\boldsymbol{b}^*}{\arg\max}\, PI\left[(\boldsymbol{a}^*)'\boldsymbol{M}^{-1/2}\boldsymbol{A}'\boldsymbol{x}, (\boldsymbol{b}^*)'\boldsymbol{N}^{-1/2}\boldsymbol{B}'\boldsymbol{y}\right] \\
&= \underset{\boldsymbol{a}^*,\boldsymbol{b}^*}{\arg\max}\, PI\left[\left(\boldsymbol{A}\boldsymbol{M}^{-1/2}\boldsymbol{a}^*\right)'\boldsymbol{x}, \left(\boldsymbol{B}\boldsymbol{N}^{-1/2}\boldsymbol{b}^*\right)'\boldsymbol{y}\right] \\
&= \underset{\boldsymbol{a},\boldsymbol{b}}{\arg\max}\, PI[\boldsymbol{a}'\boldsymbol{x}, \boldsymbol{b}'\boldsymbol{y}].
\end{aligned}$$

where PI is a robust projection index. So the robust CCA has the equivariance property, meaning new data $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ have the same canonical correlation as the original data $(\boldsymbol{x}, \boldsymbol{y})$, and their canonical coefficients satisfy

$$\boldsymbol{a}_i = \boldsymbol{A}\boldsymbol{M}^{-1/2}\boldsymbol{a}_i^* \quad \text{and} \quad \boldsymbol{b}_i = \boldsymbol{B}\boldsymbol{N}^{-1/2}\boldsymbol{b}_i^*,$$

for $i = 1, \cdots, p$. Note that for any $\boldsymbol{a}$ and $\boldsymbol{b}$,

$$\begin{aligned}
\mathrm{Var}(\boldsymbol{a}'\boldsymbol{x}^*) &= \boldsymbol{a}'\,\mathrm{Var}(\boldsymbol{x})\boldsymbol{a} = \boldsymbol{a}'\,\mathrm{Var}(\boldsymbol{M}^{-1/2}\boldsymbol{A}'x)\boldsymbol{a} \\
&= \boldsymbol{a}'(\boldsymbol{M}^{-1/2}\boldsymbol{A})\,\mathrm{Var}(\boldsymbol{x})(\boldsymbol{A}'\boldsymbol{M}^{-1/2})\boldsymbol{a} \\
&= \boldsymbol{a}'(\boldsymbol{M}^{-1/2}\boldsymbol{A}')(\boldsymbol{A}\boldsymbol{M}\boldsymbol{A}')\boldsymbol{A}\boldsymbol{M}^{-1/2})\boldsymbol{a} \\
&= \boldsymbol{a}'\boldsymbol{a}.
\end{aligned}$$

Similarly, $\mathrm{Var}(\boldsymbol{b}'\boldsymbol{y}^*) = \boldsymbol{b}'\boldsymbol{b}$. So to find the first canonical coefficients $(\boldsymbol{a}_1^*, \boldsymbol{b}_1^*)$, the projection index $PI(\boldsymbol{a}'\boldsymbol{x}^*, \boldsymbol{b}'\boldsymbol{y}^*)$ must be maximized subject to $\boldsymbol{a}'\boldsymbol{a} = 1$ and $\boldsymbol{b}'\boldsymbol{b} = 1$. One can write $\boldsymbol{a}$ and $\boldsymbol{b}$ in polar coordinates with norm 1 so that the constraint $\boldsymbol{a}'\boldsymbol{a} = 1$ and $\boldsymbol{b}'\boldsymbol{b} = 1$ can be satisfied automatically. See Branco, Croux, Filzmoser and Oliveira (2005) for more details. The projection index is then maximized, over the polar angle vectors $(\theta_1, \cdots, \theta_{p-1})$, by a standard maximization routine, *mlminb* in R. Once two angle vectors are determined by *mlminb*, they will be converted back to $(\boldsymbol{a}_1^*, \boldsymbol{b}_1^*)$.

Now assume that the first $k-1$ pairs of canonical coefficients are already obtained. To get kth pair $(\boldsymbol{a}_k, \boldsymbol{b}_k)$, the projection index $PI(\boldsymbol{a}'\boldsymbol{x}^*, \boldsymbol{b}'\boldsymbol{y}^*)$ must be maximized subject to $\boldsymbol{a}'\boldsymbol{a} = 1$, $\boldsymbol{b}'\boldsymbol{b} = 1$, $\text{Cov}(\boldsymbol{a}_k\boldsymbol{x}^*, \boldsymbol{a}_i\boldsymbol{x}^*) = 0$, and $\text{Cov}(\boldsymbol{b}_k\boldsymbol{y}^*, \boldsymbol{b}_i\boldsymbol{y}^*) = 0$ for $i = 1, \cdots, (k-1)$. Note that

$$\begin{aligned}\text{Cov}(\boldsymbol{a}_k'\boldsymbol{x}^*, \boldsymbol{a}_i'\boldsymbol{x}^*) &= \boldsymbol{a}_k'\text{Cov}(\boldsymbol{x}^*, \boldsymbol{x}^*)\boldsymbol{a}_i \\ &= \boldsymbol{a}_k'\boldsymbol{I}\boldsymbol{a}_i = \boldsymbol{a}_k'\boldsymbol{a}_i\end{aligned}$$

Likewise, $\text{Cov}(\boldsymbol{b}_k\boldsymbol{y}^*, \boldsymbol{b}_i\boldsymbol{y}^*) = \boldsymbol{b}_k'\boldsymbol{b}_i$. Hence $(\boldsymbol{a}_k, \boldsymbol{b}_k)$ can be obtained by maximizing the RPI in two subspaces that are orthogonal to $\boldsymbol{a}_1, \cdots, \boldsymbol{a}_{k-1}$ and $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_{k-1}$ respectively. Using Gram-Schmidt process, one can construct two orthogonal matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ such that

$$\boldsymbol{U} = [\boldsymbol{a}_1^*, \cdots, \boldsymbol{a}_{k-1}^* | \hat{\boldsymbol{U}}] \quad \text{and} \quad \boldsymbol{V} = [\boldsymbol{b}_1^*, \cdots, \boldsymbol{b}_{k-1}^* | \hat{\boldsymbol{V}}],$$

where $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{V}}$ are orthogonal bases of the subspaces that are orthogonal to $\boldsymbol{a}_1, \cdots, \boldsymbol{a}_{k-1}$ and $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_{k-1}$ respectively. Next project the original data to these two subspaces, one gets

$$(\boldsymbol{x}^{**}, \boldsymbol{y}^{**}) = (\hat{\boldsymbol{U}}'\boldsymbol{x}^*, \hat{\boldsymbol{V}}'\boldsymbol{y}^*).$$

Now one can obtain $(\boldsymbol{a}^{**}, \boldsymbol{b}^{**})$ with the data $(\boldsymbol{x}^{**}, \boldsymbol{y}^{**})$ by maximizing $PI(\boldsymbol{a}'\boldsymbol{x}^{**}, \boldsymbol{b}'\boldsymbol{y}^{**})$ subject to $\boldsymbol{a}'\boldsymbol{a} = 1$ and $\boldsymbol{b}'\boldsymbol{b} = 1$. After $(\boldsymbol{a}^{**}, \boldsymbol{b}^{**})$ is determined, it is transformed back to get $(\boldsymbol{a}_k^*, \boldsymbol{b}_k^*)$ by

$$\boldsymbol{a}_k^* = \hat{\boldsymbol{U}}\boldsymbol{a}^{**} \quad \text{and} \quad \boldsymbol{b}_k^* = \hat{\boldsymbol{V}}\boldsymbol{a}^{**}.$$

And then

$$\boldsymbol{a}_k = \boldsymbol{A}\boldsymbol{M}^{-1/2}\boldsymbol{a}_k^* \quad \text{and} \quad \boldsymbol{b}_k = \boldsymbol{B}\boldsymbol{N}^{-1/2}\boldsymbol{b}_k^*.$$

The k-th canonical correlation is estimated by $\rho_k = PI(\boldsymbol{a}_k'\boldsymbol{x}, \boldsymbol{b}_k'\boldsymbol{y})$ for $1 \leq k \leq p$. Once the k-th canonical covariate is obtained, a robust covariance matrix with dimension 2×2 is computed based on two univariate variables $\boldsymbol{a}_k'\boldsymbol{x}$ and $\boldsymbol{b}_k'\boldsymbol{y}$. The off-diagonal entry of this matrix is then taken to be the estimator of ρ_k.

One obvious advantage of projecting onto subspaces $(\hat{\boldsymbol{U}}, \hat{\boldsymbol{V}})$ is their lower dimensions. The maximization in a lower dimensional space can be much more computationally efficient. Another advantage is that the canonical coefficient $\boldsymbol{a}_k^*$ and $\boldsymbol{b}_k^*$ are orthogonal to all previously found $\boldsymbol{a}_i^*$ and $\boldsymbol{b}_i^*$ respectively so that the constraint of PI maximization is automatically satisfied.

5.3 SIMULATION STUDY

Two simulation studies in this section are conducted to compare eight different CCA methods, based on:

1. the classical sample covariance matrix,
2. FMCD covariance matrix estimator,
3. M covariance matrix estimator,
4. RMVN covariance matrix estimator,
5. PP-C (using the classical correlation function as the PI),
6. PP-FMCD (using the FMCD correlation estimator as the PI),
7. PP-M (using the M correlation estimator as the PI),
8. PP-RMVN (using the RMVN correlation estimator as the PI).

Simulation 1

UCLA: Academic Technology Services (2011) provides a data analysis example of CCA at http://www.ats.ucla.edu/stat/R/dae/canonical.htm. The example uses a data file, mmreg.csv, available at http://www.ats.ucla.edu/stat/R/dae/mmreg.csv. The dataset consists of 600 observations on eight variables. They are *locus of control, self-concept, motivation, reading, writing, math, science,* and *female.* The first three variables are a group of psychological variables. The next four variables are a group of academic variables. The last variable *female* is a categorical indicator. The first simulation studies the canonical correlation between these two groups of variables. The *female* variable is not included in the simulation study since the FMCD is likely to be singular when some of the variables are categorical. See Olive (2004). In fact, two Fast-MCD algorithms, *cov.mcd* and *covMcd*, failed to generate a FMCD estimator when the female variable was included. The DD plot of the mmreg dataset from Figure 5.1 shows the data follows a multivariate normal distribution since all points tightly cluster about the identity line. With the absence of apparent outliers, it is reasonable to assume this dataset is "clean". Hence, the classical canonical covariates and correlations obtained from this "clean" dataset will be used as benchmarks for a comparison of different CCA methods.

Let $(T, \boldsymbol{C})$ be the sample mean and covariance matrix of the mmreg dataset. The following different types of outliers are considered:

0. No outliers are added to original "clean" dataset.

1. 30% (in probability) of the data values are trippled.

2. 10% (in probability) of the data values are trippled.

4. 30% (in probability) of the observations are replaced by the data generated from a multivariate normal distribution, $N(T, 5\boldsymbol{C})$.

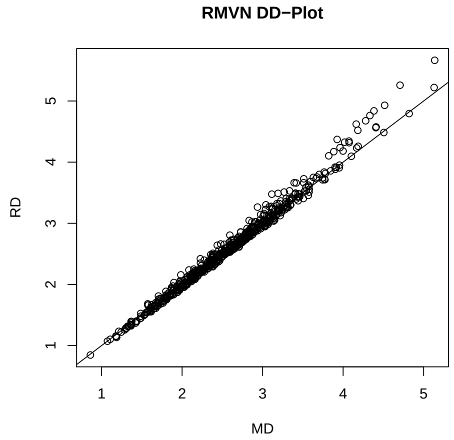

Figure 5.1. RMVN DD Plot for mmreg Data

5. 10% (in probability) of the observations are replaced by the data generated from a multivariate normal distribution, $N(T, 5\boldsymbol{C})$.

Note that when some observations are replaced by outliers, their original values of the *motivation* variable are retained on purpose since it is categorical.

Denote the k-th canonical coefficients and correlation for the i-th replication by $\hat{\boldsymbol{a}}_k^i$, $\hat{\boldsymbol{b}}_k^i$ and $\hat{\rho}_k^i$ where $k = 1, \cdots, p$ and $i = 1, \cdots, m$. Then the final estimators of k-th canonical coefficients and correlation are computed by

$$\hat{\boldsymbol{a}}_k = \frac{1}{m}\sum_i^m \hat{\boldsymbol{a}}_k^i, \quad \hat{\boldsymbol{b}}_k = \frac{1}{m}\sum_i^m \hat{\boldsymbol{b}}_k^i, \quad \text{and} \quad \hat{\rho}_k = \frac{1}{m}\sum_i^m \rho_k^i.$$

Denote the classical canonical coefficients and correlation computed from the "clean" mmreg dataset by $\boldsymbol{a}_k$, $\boldsymbol{b}_k$ and ρ_k. In the first simulation study, $\boldsymbol{a}_k$, $\boldsymbol{b}_k$ and ρ_k are used as benchmarks for a comparison of different CCA methods. The correlation, such as $\text{corr}(\hat{\boldsymbol{a}}_k, \boldsymbol{a}_k)$, between a canonical covariate and its benchmark will be used as one robustness measure. The mean squared error (MSE) of $\hat{\rho}_k$, as another robustness measure, is defined by

$$\mathrm{MSE}(\hat{\rho}_k) = \frac{1}{m}\sum_{i=1}^{m}\big(\tanh^{\text{-}1}(\hat{\rho}_k^i) - \tanh^{\text{-}1}(\rho_k)\big)^2 \tag{5.7}$$

where tanh$^{\text{-}1}$ is the inverse hyperbolic function known as the Fisher transformation in Statistics. The Fisher transformation turns the distribution of correlation coefficients toward a normal distribution. The MSE of $\boldsymbol{a}_k$ is defined by

$$\mathrm{MSE}(\hat{\boldsymbol{a}}_k) = \frac{1}{m}\sum_{i=1}^{m}\cos^{-1}\left(\frac{|\hat{\boldsymbol{a}}_k^i \boldsymbol{a}^k|}{\|\hat{\boldsymbol{a}}_k^i\| \cdot \|\boldsymbol{a}_k\|}\right), \tag{5.8}$$

and the MSE of $\boldsymbol{b}_k$ is defined in a similar manner by

$$\mathrm{MSE}(\hat{\boldsymbol{b}}_k) = \frac{1}{m}\sum_{i=1}^{m}\cos^{-1}\left(\frac{|\hat{\boldsymbol{b}}_k^i \boldsymbol{b}^k|}{\|\hat{\boldsymbol{b}}_k^i\| \cdot \|\boldsymbol{b}_k\|}\right). \tag{5.9}$$

See Branco, Croux, Filzmoser and Oliveira (2005).

The simulation program written in R language can be seen in Appendix E. The main function for the first simulation is *ccasim1*. To obtain RMVN dispersion estimator, this simulation program calls the *covrmvn* function available at http://www.math.siu.edu/olive/rpack.txt. For using the CCA PP method, the simulation program calls the routine *pp* and its subroutines which can be found at http://www.statistik.tuwien.ac.at/public/filz/programs.html. However, these original routines are not up to date and in need of many modifications as well as corrections. Furthermore, the RMVN estimator is added to the *pp* routine as one of the projection indices. Also the algorithm *cov.mcd* is substituted with *covMcd* in order to greatly increase computational speed. Finally, these modified routines are collected in a single file, ccapp.r. This file can be provided upon request. To run the simulation program, one needs to install and load some R libraries including robustbase, CCA and splus2R. The program also requires the working directory set to be "C:/work/MyDt0627/sim". One can modify the program to change the working directory setting. Before running the program, two source files, "rpack.txt" and "ccapp.r", should be included in the working directory. After running the program, the output files are put in the working directory.

The results of the simulation, with the number of replications $m = 150$, are shown in tables 5.1 and 5.2. In table 5.1, the column with header "ra1" gives the value of $\text{corr}(\hat{\boldsymbol{a}}_1, \boldsymbol{a}_1)$. All other columns to the right are similar. Table 5.1 shows all CCA methods except PP-FMCD perform well on a clean dataset ($outlier = 0$) since $\text{corr}(\hat{\boldsymbol{a}}_k, \boldsymbol{a}_k)$ and $\text{corr}(\hat{\boldsymbol{b}}_k, \boldsymbol{b}_k)$ are quite close to 1 for $k = 1, 2$. When 30% the values are trippled, the PP-FMCD and PP-M estimators failed quite badly. RMVN works well both as PI and as robust dispersion estimator. In table 5.2, the column with header "Mr1" gives the value $1000 * \text{MSE}(\hat{\rho}_1)$. The "Mr2" and "Mr3" columns are similar. The column with header "Ma1" gives the value $\text{MSE}(\hat{\boldsymbol{a}}_1)$. The rest of the columns to the right are similar. The PP-FMCD MSEs really stand out. It has larger MSEs than all other approaches for all different types of outliers. Table 5.1 and 5.2 are consistent regarding two aspects: (i) as a whole, the CCA methods using projection pursuit are not as good as the CCA methods based on robust dispersion estimators; (ii) PP-FMCD does not work well as a robust CCA technique. It was in doubt whether the performance of PP-FMCD is significantly impacted by the categorical variable, *motiviation*. So another simulation program *ccasim11* was run on the mmreg dataset with both categorical variables *motivation* and *female* removed. The result of *ccasim11* can be seen in Appendix C. Moreover, the simulation program tracking shows that the running time of the projection pursuit approach is at least 10 times longer than the approaches based on covariance matrices. Among all RPP approaches, the PP-M is the most computationally inefficient.

Simulation 2

In the second CCA simulation study, the following sampling distributions are considered:

1. normal distribution, $N_{p+q}(0, \mathbf{\Sigma})$,
2. normal mixture, $.8N_{p+q}(0, \mathbf{\Sigma}) + .2N_{p+q}(0, 8\mathbf{\Sigma})$,
3. normal mixture, $.95N_{p+q}(0, \mathbf{\Sigma}) + .05N_{p+q}(0, 8\mathbf{\Sigma})$,
4. mixture distribution, $.8N_{p+q}(0, \mathbf{\Sigma}) + .2\delta\big(tr(\mathbf{\Sigma})\mathbf{1}'\big)$,
5. mixture distribution, $.95N_{p+q}(0, \mathbf{\Sigma}) + .05\delta\big(tr(\mathbf{\Sigma})\mathbf{1}'\big)$,
6. mixture distribution, $.8N_{p+q}(0, \mathbf{\Sigma}) + .2\delta\big(tr(\mathbf{\Sigma}) * [1, 0, \cdots, 0]'\big)$,
7. mixture distribution, $.95N_{p+q}(0, \mathbf{\Sigma}) + .05\delta\big(tr(\mathbf{\Sigma}) * [1, 0, \cdots, 0]'\big)$,

where $tr(\mathbf{\Sigma})$ represents the trace of the $\mathbf{\Sigma}$ and $\delta()$ represents a point mass distribution. To form the covariance matrix $\mathbf{\Sigma}$, let $\mathbf{\Sigma}_{11} = \boldsymbol{I}_p$, $\mathbf{\Sigma}_{22} = \boldsymbol{I}_q$ and $\mathbf{\Sigma}_{12}$ be one of the following:

1. $\underset{(2\times 4)}{\mathbf{\Sigma}_{12}} = \begin{bmatrix} .9 & 0 & 0 & 0 \\ 0 & .3 & 0 & 0 \end{bmatrix}, \quad \mathbf{\Sigma}_{11} = \boldsymbol{I}_2, \quad \mathbf{\Sigma}_{22} = \boldsymbol{I}_4;$
2. $\underset{(3\times 3)}{\mathbf{\Sigma}_{12}} = \begin{bmatrix} .9 & 0 & 0 \\ 0 & .5 & 0 \\ 0 & 0 & .2 \end{bmatrix}, \quad \mathbf{\Sigma}_{11} = \boldsymbol{I}_3, \quad \mathbf{\Sigma}_{22} = \boldsymbol{I}_3;$

Table 5.1. Robust CCA with Correlation Measure

outlier	method	ra1	ra2	ra3	rb1	rb2	rb3
0	1	1.00	1.00	1.00	1.00	1.00	1.00
0	2	1.00	-1.00	-0.99	0.99	1.00	-0.73
0	3	1.00	-1.00	-0.97	-0.99	-0.98	-0.55
0	4	1.00	-1.00	0.99	-0.98	-0.98	-0.16
0	5	1.00	-1.00	1.00	-1.00	1.00	-0.62
0	6	0.60	-0.46	0.43	-0.62	0.71	0.06
0	7	1.00	-1.00	0.96	-0.99	0.99	-0.51
0	8	0.96	-0.96	-0.86	-0.85	0.99	0.00
1	1	0.99	-0.99	0.78	-0.54	0.82	0.73
1	2	0.99	-0.99	0.96	0.81	-0.99	0.48
1	3	0.97	0.99	-0.93	0.99	0.98	-0.14
1	4	0.99	-1.00	-0.97	-0.96	0.99	0.28
1	5	0.95	-0.99	0.69	-0.93	0.90	0.00
1	6	0.27	0.69	0.52	-0.67	0.73	0.00
1	7	-0.18	0.96	-0.85	-0.06	-0.98	-0.42
1	8	0.98	0.37	0.56	-0.70	0.36	0.35
2	1	0.71	-1.00	0.52	-0.19	0.92	0.17
2	2	0.96	-1.00	0.97	0.96	0.99	0.99
2	3	0.99	-1.00	-0.99	0.98	0.99	0.35
2	4	1.00	-1.00	-0.99	-0.96	-0.97	0.41
2	5	-0.30	-0.81	0.66	-0.06	0.45	0.01
2	6	0.98	0.42	0.60	-0.34	-0.86	0.38
2	7	-0.98	-1.00	-0.88	0.91	0.99	0.07
2	8	0.92	1.00	-0.97	-0.54	-0.92	0.30
3	1	0.49	-0.95	-0.65	-0.40	0.97	0.28
3	2	0.74	0.01	-0.44	0.91	-0.57	-0.55
3	3	0.95	-0.97	-0.87	-0.89	0.93	-0.86
3	4	1.00	0.98	-0.52	-0.89	0.89	0.69
3	5	-0.81	-0.99	0.06	-0.20	0.90	-0.51
3	6	-0.51	0.67	-0.45	-0.33	0.55	-0.51
3	7	-1.00	-0.76	0.57	0.52	0.70	0.63
3	8	0.98	-0.50	-0.62	-0.86	-0.21	-0.74
4	1	0.89	1.00	1.00	0.96	0.71	0.47
4	2	1.00	-1.00	-0.96	1.00	1.00	0.59
4	3	1.00	-1.00	-0.93	0.99	0.96	0.61
4	4	1.00	1.00	1.00	0.87	0.97	0.41
4	5	1.00	-1.00	0.98	-0.99	0.87	-0.84
4	6	-0.30	0.33	-0.38	-1.00	-0.94	-0.21
4	7	-0.99	1.00	-1.00	0.84	-0.90	0.93
4	8	0.75	0.94	0.96	-0.55	-0.98	-0.35

Table 5.2. Robust CCA with MSE Measure

outlier	method	Mr1	Mr2	Mr3	Ma1	Ma2	Ma3	Mb1	Mb2	Mb3
0	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	2	0.05	0.01	0.03	0.00	0.00	0.76	0.00	0.00	0.76
0	3	0.75	0.69	0.45	0.09	0.20	1.01	0.09	0.20	1.01
0	4	0.40	0.46	0.13	0.00	0.04	1.41	0.00	0.04	1.41
0	5	0.00	0.00	0.13	0.00	0.00	0.98	0.00	0.00	0.98
0	6	158.65	23.27	25.13	0.68	1.14	1.44	0.68	1.14	1.44
0	7	0.10	0.01	0.29	0.25	0.00	1.06	0.25	0.00	1.06
0	8	0.17	1.19	0.49	0.55	0.00	1.40	0.55	0.00	1.40
1	1	51.41	1.04	1.23	1.10	1.12	1.23	1.10	1.12	1.23
1	2	30.84	0.64	0.26	1.08	0.72	1.02	1.08	0.72	1.02
1	3	1.19	1.06	0.58	0.53	0.59	1.06	0.53	0.59	1.06
1	4	1.26	1.47	1.06	0.13	0.17	1.03	0.13	0.17	1.03
1	5	54.21	1.37	0.30	1.06	1.14	1.45	1.06	1.14	1.45
1	6	122.09	34.32	25.89	0.74	1.26	1.38	0.74	1.26	1.38
1	7	27.64	2.92	1.34	1.10	0.93	1.42	1.10	0.93	1.42
1	8	35.43	25.91	9.35	0.27	1.01	1.32	0.27	1.01	1.32
2	1	41.87	1.69	1.88	0.85	0.96	1.12	0.85	0.96	1.12
2	2	27.12	0.19	0.08	0.34	0.28	0.83	0.34	0.28	0.83
2	3	1.70	0.71	0.33	0.28	0.37	0.99	0.28	0.37	0.99
2	4	0.60	0.77	0.42	0.07	0.06	1.19	0.07	0.06	1.19
2	5	42.18	1.28	0.41	0.86	0.94	1.40	0.86	0.94	1.40
2	6	175.89	19.64	37.28	0.70	1.19	1.44	0.70	1.19	1.44
2	7	24.08	1.71	0.97	0.77	0.41	1.29	0.77	0.41	1.29
2	8	27.41	15.95	9.63	0.26	0.93	1.33	0.26	0.93	1.33
3	1	3.08	2.10	1.00	0.94	1.05	1.19	0.94	1.05	1.19
3	2	1.95	1.76	1.14	0.78	0.80	1.03	0.78	0.80	1.03
3	3	2.27	1.53	0.97	0.58	0.65	1.02	0.58	0.65	1.02
3	4	1.93	1.81	0.93	0.26	0.40	0.94	0.26	0.40	0.94
3	5	2.81	2.25	0.29	0.96	1.04	1.40	0.96	1.04	1.40
3	6	246.93	23.08	31.15	0.87	1.19	1.42	0.87	1.19	1.42
3	7	2.32	2.13	0.93	0.80	0.93	1.35	0.80	0.93	1.35
3	8	33.57	23.21	11.04	0.58	1.00	1.30	0.58	1.00	1.30
4	1	1.42	1.81	1.08	0.64	0.72	0.98	0.64	0.72	0.98
4	2	0.56	0.75	0.38	0.38	0.38	0.85	0.38	0.38	0.85
4	3	1.85	0.74	0.39	0.32	0.39	0.90	0.32	0.39	0.90
4	4	0.62	0.66	0.52	0.10	0.13	1.13	0.10	0.13	1.13
4	5	1.85	1.26	0.24	0.67	0.69	1.37	0.67	0.69	1.37
4	6	225.93	24.20	33.37	0.76	1.12	1.45	0.76	1.12	1.45
4	7	1.61	1.41	0.44	0.46	0.49	1.26	0.46	0.49	1.26
4	8	31.97	18.19	8.31	0.37	0.93	1.31	0.37	0.93	1.31

$$3.\ \underset{(5\times5)}{\boldsymbol{\Sigma}_{12}} = \begin{bmatrix} .9 & 0 & 0 & 0 & 0 \\ 0 & .7 & 0 & 0 & 0 \\ 0 & 0 & .4 & 0 & 0 \\ 0 & 0 & 0 & .3 & 0 \\ 0 & 0 & 0 & 0 & .1 \end{bmatrix}, \quad \boldsymbol{\Sigma}_{11} = \boldsymbol{I}_5, \quad \boldsymbol{\Sigma}_{22} = \boldsymbol{I}_5.$$

$\boldsymbol{\Sigma}_{11}$ and $\boldsymbol{\Sigma}_{22}$ are set to be identity matrices due to the equivariant property of CCA. The sample size of the simulation is $n = 1000$ and the number of replications is $m = 200$. The benchmarks in simulation 2 are the true values of ρ_k, $\boldsymbol{a}_k$ and $\boldsymbol{b}_k$ computed from the matrix $\boldsymbol{\Sigma}$. The main function for the simulation is *ccasim2*. See Appendix E.

The result of simulation 2 when $\boldsymbol{\Sigma}$ is formed by the third choice above is presented in table 5.3. The "cov" column indicates the choice of $\boldsymbol{\Sigma}$, the "std" column indicates the type of sampling distribution, and the "mdt" column indicates the CCA methods. Although $p = q = 5$ in this case, only the results of first two canonical covariates are listed due to the limit of the space. Table 5.3 shows that the MSE(ρ_k) of classical CCA (as well as classical PP) increases rapidly when the point mass outliers are introduced. For the normal mixture sampling distribution, only PP-MCD does not work well. For the mixture distribution $.8N_{p+q}(0, \boldsymbol{\Sigma}) + .2\delta\big(tr(\boldsymbol{\Sigma})\mathbf{1}'\big)$, only RMVN and PP-RMVN CCA perform well. The result of the mixture distribution $.8N_{p+q}(0, \boldsymbol{\Sigma}) + .2\delta\big(tr(\boldsymbol{\Sigma}) * [1, 0, \cdots, 0]'\big)$ is quite similar. In general, it is observed that RMVN and PP-RMVN have the best performance when the underlying distribution has the multivariate normality. Between them, the CCA based on RMVN approach should be adopted since it has the computational efficiency advantage. The results of simulation 2 with $\boldsymbol{\Sigma}$ using the other two choices are very similar and can be seen in Appendix C.

Table 5.3. Robust CCA Simulation 2, cov type=3

cov	sdt	mdt	ra1	ra2	rb1	rb2	Mr1	Mr2	Ma1	Ma2	Mb1	Mb2
3	1	1	1.00	1.00	1.00	1.00	3.21	1.35	0.32	0.22	0.32	0.22
3	1	2	1.00	1.00	1.00	1.00	2.96	1.25	0.30	0.22	0.30	0.22
3	1	3	1.00	1.00	1.00	0.99	2.92	0.48	0.28	0.27	0.28	0.27
3	1	4	1.00	1.00	1.00	0.99	2.24	0.63	0.26	0.25	0.26	0.25
3	1	5	1.00	1.00	1.00	1.00	3.21	1.35	0.32	0.22	0.32	0.22
3	1	6	0.83	0.81	0.84	0.80	1755.97	791.31	0.53	0.58	0.53	0.58
3	1	7	1.00	1.00	1.00	1.00	2.42	1.04	0.30	0.22	0.30	0.22
3	1	8	1.00	1.00	1.00	1.00	1.46	0.22	0.27	0.24	0.27	0.24
3	2	1	1.00	1.00	1.00	0.99	0.65	0.51	0.91	0.91	0.91	0.91
3	2	2	1.00	1.00	1.00	1.00	0.19	0.41	0.47	0.47	0.47	0.47
3	2	3	1.00	1.00	1.00	1.00	0.89	2.50	0.48	0.44	0.48	0.44
3	2	4	1.00	1.00	1.00	1.00	1.64	3.91	0.15	0.09	0.15	0.09
3	2	5	1.00	1.00	1.00	0.99	0.65	0.51	0.91	0.91	0.91	0.91
3	2	6	0.86	0.86	0.90	0.81	932.23	323.75	0.62	0.72	0.62	0.72
3	2	7	1.00	1.00	1.00	1.00	2.10	0.02	0.53	0.53	0.53	0.53
3	2	8	1.00	1.00	1.00	1.00	3.12	0.28	0.14	0.09	0.14	0.09
3	3	1	0.99	0.99	0.99	0.98	1.49	1.28	0.51	0.57	0.51	0.57
3	3	2	1.00	1.00	1.00	0.99	1.20	4.75	0.25	0.31	0.25	0.31
3	3	3	1.00	1.00	1.00	1.00	1.32	2.21	0.26	0.30	0.26	0.30
3	3	4	1.00	1.00	1.00	1.00	1.61	1.90	0.21	0.21	0.21	0.21
3	3	5	0.99	0.99	0.99	0.98	1.49	1.28	0.51	0.57	0.51	0.57
3	3	6	0.82	0.78	0.77	0.69	1235.39	1014.15	0.60	0.77	0.60	0.77
3	3	7	1.00	1.00	1.00	0.99	1.18	6.41	0.26	0.33	0.26	0.33
3	3	8	1.00	1.00	1.00	1.00	2.75	10.68	0.21	0.20	0.21	0.20
3	4	1	0.97	0.62	0.98	0.63	3154.60	124.85	1.43	0.90	1.43	0.90
3	4	2	0.97	0.62	0.98	0.62	3281.19	133.00	1.43	0.88	1.43	0.88
3	4	3	1.00	0.68	0.99	0.62	4684.50	223.34	1.44	0.81	1.44	0.81
3	4	4	1.00	0.99	1.00	1.00	1.50	0.23	0.27	0.00	0.27	0.00
3	4	5	0.97	0.62	0.98	0.63	3154.60	124.85	1.43	0.90	1.43	0.90
3	4	6	0.83	0.72	0.85	0.60	108.47	723.71	1.14	0.88	1.14	0.88
3	4	7	0.98	0.63	0.98	0.61	3252.78	133.14	1.42	0.90	1.42	0.90
3	4	8	1.00	0.99	1.00	1.00	0.60	0.08	0.27	0.00	0.27	0.00
3	5	1	0.97	0.61	0.97	0.59	1416.06	89.40	1.32	0.99	1.32	0.99
3	5	2	0.98	0.68	0.98	0.65	404.06	69.29	1.09	0.90	1.09	0.90
3	5	3	1.00	1.00	1.00	0.99	1.35	0.30	0.33	0.18	0.33	0.18
3	5	4	1.00	1.00	1.00	0.99	1.30	0.40	0.22	0.00	0.22	0.00
3	5	5	0.97	0.61	0.97	0.59	1416.06	89.40	1.32	0.99	1.32	0.99
3	5	6	1.00	0.86	1.00	0.87	4.72	425.08	0.33	0.45	0.33	0.45
3	5	7	1.00	0.62	1.00	0.60	62.54	96.29	1.17	0.98	1.17	0.98
3	5	8	1.00	1.00	1.00	1.00	2.30	0.79	0.22	0.00	0.22	0.00
3	6	1	0.30	0.46	0.31	0.46	384.04	159.29	1.53	1.43	1.53	1.43
3	6	2	0.30	0.45	0.31	0.45	389.42	156.69	1.52	1.43	1.52	1.43
3	6	3	1.00	1.00	1.00	1.00	1.49	0.34	0.43	0.34	0.43	0.34
3	6	4	1.00	1.00	1.00	1.00	1.61	0.68	0.20	0.00	0.20	0.00
3	6	5	0.30	0.46	0.31	0.46	384.04	159.29	1.53	1.43	1.53	1.43
3	6	6	0.87	0.80	0.83	0.80	564.07	362.96	0.62	0.66	0.62	0.66
3	6	7	0.30	0.45	0.29	0.43	375.30	152.88	1.54	1.44	1.54	1.44
3	6	8	1.00	0.95	1.00	0.98	1.23	6.55	0.19	0.15	0.19	0.15
3	7	1	0.23	0.35	0.29	0.43	395.83	231.61	1.55	1.48	1.55	1.48
3	7	2	0.24	0.39	0.32	0.49	412.43	223.62	1.54	1.45	1.54	1.45
3	7	3	1.00	0.99	1.00	0.99	0.55	2.30	0.13	0.31	0.13	0.31
3	7	4	1.00	1.00	1.00	0.99	1.11	1.83	0.00	0.02	0.00	0.02
3	7	5	0.23	0.35	0.29	0.43	395.83	231.61	1.55	1.48	1.55	1.48
3	7	6	0.75	0.77	0.79	0.77	2121.97	552.10	0.51	0.66	0.51	0.66
3	7	7	0.23	0.38	0.30	0.44	397.27	217.56	1.54	1.50	1.54	1.50
3	7	8	1.00	1.00	1.00	1.00	0.03	1.53	0.00	0.00	0.00	0.00

CHAPTER 6

CONCLUSIONS

Robust outlier resistant estimators of MLD should be i) $\sqrt{n}$ consistent for a large class of distributions, ii) easy to compute, iii) effective at detecting certain types of outliers and iv) outlier resistant. Although Hawkins and Olive (2002) showed that almost all of the literature focuses either on i) and iv) or on ii) and iii), Olive and Hawkins (2010) shows that it is simple to construct estimators satisfying i)–iv) provided that $n > 20p$ and $p \leq 40$. These results represent both a computational and theoretical breakthrough in the field of robust MLD.

The new FCH, RFCH and RMVN estimators use information from both location and dispersion criteria and are more effective at screening attractors than estimators such as MBA and FMCD that only use the MCD dispersion criterion. The new estimators are roughly two orders of magnitude faster than FMCD.

The collection of easily computed "robust estimators" for MLD that have not been shown to be both HB and consistent is enormous, but without theory the methods should be classified as outlier diagnostics rather than robust statistics.

Examine the estimator on many "benchmark data sets." FCH was examined on 30 such data sets. Outlier performance was competitive with estimators such as FMCD. For any given estimator, it is easy to find outlier configurations where the estimator fails. For the modified wood data of Rousseeuw (1984), MB detected the planted outliers but FCH used DGK. For another data set, 2 clean cases had larger MB distances than 4 of 5 planted outliers that FMCD can detect. For small p, elemental methods can be used as outlier diagnostics.

Simulations were done in *R*. Majority of the programs are in the collection of functions *rpack.txt* at (www.math.siu.edu/olive/ol-bookp.htm). The `robustbase` library was used to compute FMCD. Function `covrmvn` computes the FCH, RMVN and MB estimators while `covfch` computes the FCH, RFCH and MB estimators. Function `covesim` computes the $\hat{\boldsymbol{\Sigma}}$ on contaminated normal data. Function `rhotsim` does the robust Hotelling's T^2 test and function `pcasim` does the robust PCA based on RMVN. The robust CCA simulation program is attached in Appendix E.

REFERENCES

[1] Bernholt, T., and Fischer, P. (2004), “The Complexity of Computing the MCD-Estimator,” *Theoretical Computer Science,* 326, 383-398.

[2] Bhatia, R., Elsner, L., and Krause, G. M. (1990), “Bounds for the variation of the roots of a polynomial and the eigenvalues of a matrix,” *Linear Algebra and its Applications,* 142, 195-209

[3] Boente, G., and Fraiman, R. (1999), “Comment on Robust principal component analysis for functional data by N. Locantoe, J. Marron, D. Simpson, N. Tripoli, J. Zhang and K. Cohen,” Test, Spain, 8, 28-35.

[4] Branco, J. A., Croux, C., Filzmoser, P., and Oliveira, M. R. (2005), “Robust Canonical Correlation: A Comparative Study,” *Computational Statistics*, 20, 203-229.

[5] Butler, R. W., Davies, P. L., and Jhun, M. (1993), “Asymptotics for the Minimum Covariance Determinant Estimator,” *The Annals of Statistics,* 21, 1385-1400.

[6] Buxton, L. H .D. (1920), “The Anthropology of Cyprus,” *The Journal of the Royal Anthropological Institute of Great Britain and Ireland,* 50, 183-235.

[7] Casella, G., and Berger, R. L. (2002), *Statistical Inference,* 2nd ed. CA: Duxbury Press.

[8] Cator, E., and Lopuhaa, H. (2009), “Central limit theorem and influence function for the MCD estimators at general multivariate distributions,” online text available from (http://arxiv.org/PS_cache/arxiv/pdf/0907/0907.0079v2.pdf).

[9] Cattell, R. B. (1966), “The Scree Test for the Number of Factors,” *Multivariate Behavioral Research,* 1, 245-276.

[10] Croux, C., Filzmoser, P., and Oliveira, M. R. (2007), “Algorithms for Projection-Pursuit Robust Principal Component Analysis,” *Chemometrics and*

Intelligent Laboratory Systems, 87, 218-225.

[11] Croux, C., and Haesbroeck, G. (2000), “Principal Component Analysis based on Robust Estimators of the Covariance or Correlation Matrix: Influence Functions and Efficiencies,” *Biometrika,* 87,603-61.

[12] Datta, B. N. (1995), *Numerical Linear Algebra and Applications,* Pacific Grove, CA: Brooks/Cole.

[13] Devlin, S. J., Gnanadesikan, R., and Kettenring, J. R. (1975), “Robust Estimation and Outlier Detection with Correlation Coefficients,” *Biometrika,* 62, 531-545.

[14] Devlin, S. J., Gnanadesikan, R., and Kettenring, J. R. (1981), “Robust Estimation of Dispersion Matrices and Principal Components,” *Journal of the American Statistical Association,* 76, 354-362.

[15] Eaton, M. L., and Tyler, D. E. (1991) “On Wielandt’s Inequality and Its Application to the Asymptotic Distribution of the Eigenvalues of a Random Sysmmetric Matrix,” *The Annals of Statistics,* 19, 260-271.

[16] Gladstone, R. J. (1905-6), “A Study of the Relations of the Brain to the Size of the Head,” *Biometrika,* 4, 105-123.

[17] Gnanadesikan, R., and Kettenring, J. (1972), “Robust Estimates, Residuals, and Outlier Detection with Multiresponse Data,” *Biometrics,* 28, 81-124.

[18] Hawkins, D. M., and Olive, D. J. (1999) “Improved Feasible Solution Algorithms for High Breakdown Estimation,” *Computational Statsistics and Data Analysis,* 30, 1-11

[19] Huber, P. J. (1981), *Robust Statistics.* NY, John Wiley & Sons.

[20] Huber, P. J. (1985), “Projection Pursuit,” *The Annals of Statistics*, 13, 435-475.

[21] Huber, P. J., and Ronchetti, E. M. (2009), *Robust Statistics,* 2nd ed. Hoboken, NJ: John Wiley & Sons.

[22] Hubert, M., Rousseeuw, P. J., and Van Aelst, S. (2008), “High Breakdown

Multivariate Methods," *Statistical Science,* 23, 92-119.

[23] Johnson, M. E. (1987), *Multivariate Statistical Simulation,* Canada: John Wiley & Sons.

[24] Johnson, R. A., and Wichern, D. W. (1998), *Applied Multivariate Statistical Analysis,* 4th ed. NJ: Prentice Hall.

[25] Leon, S. J. (1986), *Linear Algebra with Applications,* 2nd ed. NY: Macmillan Publishing Company.

[26] Li, G., and Chen, Z. (1985), "Projection-Pursuit Approach to Robust Dispersion matrices and Principal Components: Primary Theory and Monte Carlo," *American Statistical Association,* 80, 759-766.

[27] Locantore, N., Marron, J., Simpson, D., Tripoli, N., Zhang, J., and Cohen, K. (1999), "Robust Principal Components for Functional Data," *Test,* 8, 1-28.

[28] Lopuhaä, H. P. (1999), "Asymptotics of Reweighted Estimators of Multivariate Location and Scatter," *The Annals of Statistics,* 27, 1638-1665.

[29] Maronna, R. A., Martin R. D., and Yohai V. J. (2006), *Robust Statistics - Theory and Methods.* England: John Wiley & Sons.

[30] Maronna, R. A., and Zamar, R. H. (2002), "Robust Estimates of Location and Dispersion for High-Dimensional Datasets," *Technometrics,* 50, 295-304.

[31] Muirhead, R. J., and Waternaux, C. M. (1980), "Asymptotic Distribution in Canonical Correlation Analysis and Other Multivariate Procedures for Nonnormal Populations," *Biometrika,* 67, 31-43.

[32] Olive, D. J. (2002), "Applications of Robust Distances for Regression," *Technometrics,* 44, 64-71.

[33] Olive, D. J. (2004), "A Resistant Estimator of Multivariate Location and Dispersion," *Computational Statistics and Data Analysis,* 46, 99-102.

[34] Olive, D. J. (2008), *Applied Robust Statistics,* unpublished online text available from (www.math.siu.edu/olive/ol-bookp.htm).

[35] Olive, D. J., and Hawkins, D. M. (2008), "High Breakdown Multivariate Estimators," Preprint, from (www.math.siu.edu/olive/pphbrs.pdf).

[36] Olive, D. J., and Hawkins, D. M. (2010), "Robust Multivariate Location and Dispersion," unpublished manuscript available from (http://www.math.siu.edu/olive/pphbmld.pdf).

[37] Pratt, J. W. (1959), "On a General Concept of 'in Probability'," *The Annals of Mathematical Statistics,* 30, 549-558.

[38] Rocke, D. M., and Woodruff, D.L. (1996), "Identification of Outliers in Multivariate Data," *American Statistical Association,* 91, 1047-1061.

[39] Rousseeuw, P. J. (1984), "Least Median of Squares Regression," *Journal of the American Statistical Association,* 79, 871-880.

[40] Rousseeuw, P. J., and Leroy, A.M. (1987), *Robust Regression and Outlier Detection.* NY: John Wiley & Sons.

[41] Rousseeuw, P. J., and Van Driessen, K. (1999), "A Fast Algorithm for the Minimum Covariance Determinant Estimator," *Technometrics,* 41, 212-223.

[42] Stuart, A., Ord, J. K., and Arnold, S. (1999). *Advanced Theory of Statistics,* Volume II, 5th ed. New York: Oxford University Press.

[43] Tremearne, A. J. N. (1911), "Notes on Some Nigerian Tribal Marks," *Journal of the Royal Anthropological Institute of Great Britain and Ireland,* 41, 162-178.

[44] UCLA: Academic Technology Services (2011), "R Data Analysis Examples, Canonical Correlation Analysis," available from (http://www.ats.ucla.edu/stat/R/dae/canonical.htm).

[45] Willems, G., Pison, G., Rousseeuw, P.J., and Van Aelst, S. (2002), "A Robust Hotelling Test," *Metrika,* 55, 125-138.

APPENDIX A: MEAN “VARIANCE EXPLAINED”’ BY PCA AND RPCA

Table 6.1: Variance Explained by PCA and RPCA, p=3

n	vexpl			rvexpl			a1	a2	a3
60	0.5258	0.3146	0.1596	0.5359	0.3146	0.1495	0.951	0.918	0.975
	0.0444	0.0376	0.0296	0.0563	0.0457	0.0341			
100	0.5174	0.3250	0.1576	0.5241	0.3214	0.1545	0.958	0.954	0.988
	0.0429	0.0343	0.0252	0.0491	0.0399	0.0278			
200	0.5084	0.3323	0.1592	0.5085	0.3289	0.1626	0.952	0.937	0.986
	0.0372	0.0344	0.0234	0.0396	0.0333	0.0235			
300	0.5113	0.3291	0.1596	0.5098	0.3276	0.1627	0.948	0.953	0.983
	0.0334	0.0318	0.0191	0.0334	0.0302	0.0210			
400	0.5038	0.3297	0.1665	0.5107	0.3301	0.1592	0.949	0.929	0.987
	0.0265	0.0249	0.0131	0.0484	0.0413	0.0261			
500	0.5140	0.3254	0.1606	0.5065	0.3297	0.1638	0.954	0.955	0.987
	0.0457	0.0389	0.0234	0.0227	0.0215	0.0116			
350	0.5177	0.3197	0.1626	0.5108	0.3258	0.1634	0.951	0.944	0.986
	0.0407	0.0356	0.0212	0.0280	0.0224	0.0179			
500	0.5265	0.3144	0.1591	0.5086	0.3279	0.1635	0.950	0.923	0.979
	0.0650	0.0475	0.0324	0.0287	0.0252	0.0165			
200	0.5084	0.3323	0.1592	0.5085	0.3289	0.1626	0.952	0.937	0.986
	0.0372	0.0344	0.0234	0.0396	0.0333	0.0235			
1500	0.5119	0.3223	0.1658	0.5005	0.3322	0.1673	0.949	0.892	0.988
	0.0654	0.0531	0.0361	0.0372	0.0334	0.0203			
2000	0.7711	0.1749	0.0540	0.4995	0.3328	0.1677	0.746	0.691	0.796
	0.1541	0.1190	0.0497	0.0175	0.0158	0.0095			

Table 6.2: Variance Explained by PCA and RPCA, p=5

n	vexpl			rvexpl			a1	a2	a3
100	0.3566	0.2653	0.1897	0.3617	0.2678	0.1873	0.953	0.930	0.928
	0.0284	0.0244	0.0225	0.0326	0.0257	0.0227			
1000	0.3381	0.2649	0.1992	0.3359	0.2663	0.1979	0.958	0.931	0.957
	0.0155	0.0129	0.0110	0.0162	0.0132	0.0126			
1600	0.3381	0.2665	0.1988	0.3405	0.2684	0.1981	0.968	0.953	0.971
	0.0105	0.0100	0.0071	0.0161	0.0160	0.0134			
1600	0.3400	0.2652	0.1977	0.3356	0.2654	0.1997	0.952	0.920	0.952
	0.0190	0.0172	0.0144	0.0093	0.0093	0.0074			
1500	0.3392	0.2663	0.1966	0.3351	0.2654	0.2000	0.944	0.917	0.943
	0.0198	0.0170	0.0139	0.0096	0.0096	0.0079			
1600	0.3374	0.2669	0.2002	0.3346	0.2674	0.1982	0.946	0.943	0.973
	0.0147	0.0121	0.0108	0.0126	0.0096	0.0079			
8000	0.3482	0.2624	0.1948	0.3338	0.2664	0.1992	0.952	0.934	0.959
	0.0328	0.0162	0.0129	0.0061	0.0046	0.0043			
1200	0.3351	0.2680	0.1985	0.3346	0.2662	0.1996	0.948	0.936	0.974
	0.0160	0.0147	0.0098	0.0108	0.0118	0.0098			
200	0.3464	0.2655	0.1958	0.3496	0.2642	0.1947	0.951	0.912	0.949
	0.0213	0.0180	0.0163	0.0238	0.0214	0.0179			
20000	0.7708	0.1390	0.0536	0.3337	0.2669	0.1993	0.538	0.474	0.462
	0.1734	0.1001	0.0528	0.0043	0.0035	0.0028			
9000	0.3330	0.2682	0.1970	0.3327	0.2678	0.1996	0.950	0.920	0.973
	0.0220	0.0216	0.0156	0.0108	0.0102	0.0082			

Table 6.3: Variance Explained by PCA and RPCA, p=8

n	vexpl			rvexpl			a1	a2	a3
200	0.2372	0.1959	0.1644	0.2398	0.1968	0.1634	0.948	0.894	0.887
	0.0149	0.0109	0.0102	0.0157	0.0116	0.0104			
4000	0.2240	0.1945	0.1670	0.2240	0.1951	0.1668	0.959	0.937	0.956
	0.0055	0.0051	0.0046	0.0054	0.0044	0.0044			
3300	0.2232	0.1951	0.1673	0.2257	0.1957	0.1672	0.951	0.923	0.946
	0.0056	0.0052	0.0047	0.0077	0.0078	0.0070			
5000	0.2256	0.1959	0.1656	0.2225	0.1954	0.1666	0.948	0.912	0.925
	0.0072	0.0074	0.0073	0.0040	0.0034	0.0036			
4500	0.2244	0.1947	0.1666	0.2234	0.1949	0.1660	0.949	0.931	0.954
	0.0059	0.0060	0.0053	0.0045	0.0040	0.0036			
30000	0.2328	0.1940	0.1645	0.2223	0.1945	0.1667	0.920	0.907	0.905
	0.0355	0.0108	0.0104	0.0024	0.0021	0.0020			
2800	0.2255	0.1947	0.1664	0.2237	0.1954	0.1659	0.955	0.926	0.936
	0.0067	0.0066	0.0061	0.0056	0.0060	0.0050			
600	0.2300	0.1951	0.1660	0.2297	0.1966	0.1661	0.950	0.930	0.941
	0.0112	0.0080	0.0069	0.0121	0.0089	0.0082			
60000	0.6703	0.1650	0.0737	0.2223	0.1946	0.1668	0.409	0.384	0.401
	0.1934	0.0899	0.0560	0.0021	0.0016	0.0017			
25000	0.2218	0.1946	0.1675	0.2231	0.1944	0.1659	0.946	0.885	0.910
	0.0099	0.0088	0.0088	0.0053	0.0049	0.0039			

Table 6.4: Variance Explained by PCA and RPCA, p=12

n	vexpl			rvexpl			a1	a2	a3
500	0.1610	0.1444	0.1294	0.1621	0.1449	0.1296	0.964	0.936	0.934
	0.0060	0.0054	0.0055	0.0061	0.0058	0.0056			
10000	0.1553	0.1411	0.1282	0.1545	0.1411	0.1283	0.957	0.941	0.949
	0.0026	0.0027	0.0021	0.0024	0.0027	0.0022			
5000	0.1550	0.1409	0.1284	0.1564	0.1415	0.1284	0.948	0.903	0.918
	0.0031	0.0028	0.0029	0.0041	0.0037	0.0036			
15000	0.1557	0.1418	0.1281	0.1543	0.1409	0.1283	0.957	0.934	0.944
	0.0036	0.0033	0.0030	0.0019	0.0018	0.0013			
15000	0.1547	0.1409	0.1285	0.1542	0.1411	0.1283	0.954	0.940	0.954
	0.0028	0.0025	0.0022	0.0019	0.0018	0.0017			
60000	0.1681	0.1406	0.1261	0.1542	0.1411	0.1281	0.869	0.841	0.863
	0.0498	0.0093	0.0079	0.0013	0.0009	0.0011			
8000	0.1551	0.1417	0.1283	0.1551	0.1408	0.1284	0.952	0.935	0.941
	0.0037	0.0031	0.0027	0.0030	0.0026	0.0023			
2000	0.1554	0.1406	0.1280	0.1562	0.1410	0.1279	0.958	0.931	0.932
	0.0048	0.0040	0.0033	0.0051	0.0040	0.0036			
80000	0.6984	0.1448	0.0641	0.1540	0.1410	0.1282	0.365	0.315	0.321
	0.2065	0.1026	0.0541	0.0013	0.0011	0.0012			
70000	0.1534	0.1412	0.1284	0.1543	0.1412	0.1283	0.918	0.853	0.901
	0.0059	0.0043	0.0045	0.0031	0.0026	0.0021			

Table 6.5: Variance Explained by PCA and RPCA, p=20

n	vexpl			rvexpl			a1	a2	a3
800	0.1014	0.0943	0.0874	0.1022	0.0945	0.0876	0.956	0.922	0.915
	0.0029	0.0029	0.0026	0.0029	0.0028	0.0024			
30000	.0957	0.0906	0.0859	0.0957	0.0907	0.0857	0.958	0.937	0.946
	0.0010	0.0010	0.0009	0.0009	0.0009	0.0009			
10000	0.0962	0.0910	0.0860	0.0970	0.0913	0.0863	0.948	0.895	0.879
	0.0015	0.0013	0.0013	0.0018	0.0016	0.0017			
40000	0.0959	0.0909	0.0858	0.0954	0.0905	0.0857	0.946	0.900	0.901
	0.0013	0.0015	0.0012	0.0008	0.0007	0.0006			
40000	0.0957	0.0907	0.0859	0.0954	0.0905	0.0857	0.961	0.944	0.950
	0.0012	0.0010	0.0009	0.0007	0.0008	0.0007			
60000	0.1076	0.0913	0.0856	0.0954	0.0906	0.0858	0.727	0.686	0.671
	0.0319	0.0038	0.0035	0.0006	0.0007	0.0008			
80000	0.6557	0.1546	0.0617	0.0953	0.0905	0.0858	0.236	0.246	0.251
	0.2381	0.1096	0.0502	0.0007	0.0007	0.0007			
80000	0.0956	0.0905	0.0859	0.0956	0.0906	0.0859	0.813	0.656	0.723
	0.0030	0.0023	0.0023	0.0016	0.0014	0.0012			

APPENDIX B: HOTELLING T TEST SIMULATION

Table 6.6: Hotelling Simulation (i)

p	n	hcv	rhcvr	p	n	hcv	rhcvr
5	40	0.06	0.04	60	660	0.06	0.16
5	41	0.08	0.12	60	700	0.07	0.14
5	42	0.07	0.05	60	750	0.10	0.16
5	43	0.02	0.04	60	800	0.05	0.08
10	40	0.04	0.21	60	840	0.05	0.06
10	80	0.03	0.08	60	850	0.04	0.11
10	90	0.06	0.05	60	900	0.07	0.06
10	100	0.02	0.01	60	950	0.04	0.06
10	95	0.10	0.06	60	980	0.03	0.04
10	97	0.04	0.04	60	970	0.05	0.05
10	96	0.04	0.03	65	1100	0.07	0.12
15	120	0.06	0.08	65	1200	0.05	0.06
15	150	0.05	0.04	65	1300	0.02	0.02
15	140	0.08	0.05	65	1240	0.04	0.07
15	145	0.08	0.05	65	1250	0.03	0.03
15	149	0.03	0.06	70	1300	0.07	0.05
20	180	0.05	0.06	70	1350	0.08	0.07
20	200	0.03	0.05	70	1400	0.11	0.12
20	190	0.06	0.03	70	1500	0.07	0.07
20	195	0.02	0.03	70	1600	0.03	0.06
20	194	0.06	0.05	70	1650	0.07	0.07
25	230	0.08	0.06	70	1700	0.05	0.06
25	250	0.09	0.08	70	1740	0.08	0.06
25	250	0.04	0.04	70	1750	0.03	0.02
30	290	0.09	0.07	75	1750	0.04	0.06
30	300	0.06	0.06	75	1760	0.04	0.04
30	300	0.07	0.06	80	1800	0.02	0.03
30	310	0.05	0.05	80	1750	0.04	0.11

35	340	0.03	0.08	85	1900	0.05	0.04
35	350	0.08	0.10	85	1850	0.04	0.05
35	360	0.08	0.06	85	1840	0.08	0.06
35	370	0.05	0.10	90	1900	0.06	0.12
35	380	0.05	0.05	90	2000	0.05	0.06
40	400	0.03	0.09	90	2040	0.06	0.07
40	440	0.07	0.07	90	2050	0.04	0.01
40	460	0.07	0.04	95	2050	0.07	0.06
40	470	0.03	0.02	95	2100	0.03	0.07
45	500	0.05	0.11	95	2120	0.08	0.06
45	550	0.03	0.07	95	2200	0.07	0.11
45	560	0.07	0.09	95	2250	0.07	0.06
45	570	0.01	0.02	95	2300	0.09	0.10
50	500	0.01	0.07	95	2350	0.04	0.05
50	550	0.06	0.04	95	2340	0.04	0.05
50	560	0.09	0.11	95	2330	0.06	0.06
50	600	0.04	0.03	100	2400	0.04	0.08
50	590	0.05	0.08	100	2450	0.03	0.07
55	600	0.05	0.13	100	2500	0.05	0.07
55	800	0.08	0.06	100	2550	0.08	0.12
55	840	0.08	0.07	100	2600	0.06	0.05
55	850	0.04	0.05	100	2610	0.01	0.03

Table 6.7: Hotelling Simulation (ii)

p	n=15p	hcv	rhcvr	n=20p	hcv	rhcvr	n=30p	hcv	rhcvr
2	30	0.0502	0.0516	40	0.0498	0.0624	60	0.0540	0.0382
5	75	0.0500	0.0456	100	0.0474	0.0250	150	0.0542	0.0310
10	150	0.0476	0.0300	200	0.0516	0.0304	300	0.0498	0.0286
15	225	0.0474	0.0318	300	0.0506	0.0308	450	0.0492	0.0320
20	300	0.0540	0.0368	400	0.0548	0.0314	600	0.0520	0.0354
25	375	0.0444	0.0334	500	0.0462	0.0296	750	0.0456	0.0288
30	450	0.0472	0.0324	600	0.0516	0.0358	900	0.0484	0.0342
35	525	0.0490	0.0384	700	0.0522	0.0358	1050	0.0502	0.0374
40	600	0.0534	0.0440	800	0.0486	0.0354	1200	0.0526	0.0336
45	675	0.0406	0.0390	900	0.0544	0.0390	1350	0.0512	0.0366
50	750	0.0498	0.0430	1000	0.0522	0.0394	1500	0.0512	0.0364
55	825	0.0504	0.0502	1100	0.0496	0.0392	1650	0.0510	0.0374
60	900	0.0482	0.0514	1200	0.0488	0.0404	1800	0.0474	0.0376
65	975	0.0568	0.0602	1300	0.0524	0.0414	1950	0.0548	0.0410
70	1050	0.0462	0.0530	1400	0.0558	0.0432	2100	0.0522	0.0424
75	1125	0.0474	0.0632	1500	0.0502	0.0486	2250	0.0490	0.0370
80	1200	0.0524	0.0620	1600	0.0524	0.0432	2400	0.0468	0.0356
85	1275	0.0482	0.0758	1700	0.0496	0.0456	2550	0.0520	0.0404
90	1350	0.0504	0.0746	1800	0.0484	0.0454	2700	0.0484	0.0398
95	1425	0.0524	0.0892	1900	0.0472	0.0506	2850	0.0538	0.0424
100	1500	0.0554	0.0808	2000	0.0452	0.0506	3000	0.0488	0.0392

Table 6.8: Hotelling Simulation (iii)

p	n=15p	δ	hcv	n=20p	δ	hcv	n=30p	δ	hcv
2	30	0.30	0.32	40	0.20	0.30	60	0.20	0.41
2	30	0.40	0.57	40	0.40	0.79	60	0.30	0.75
2	30	0.60	0.90	40	0.50	0.93	60	0.12	0.94
5	75	0.20	0.33	100	0.15	0.25	150	0.10	0.27
5	75	0.22	0.60	100	0.20	0.66	150	0.15	0.56
5	75	0.30	0.82	100	0.30	0.94	150	0.20	0.87
10	150	0.10	0.32	200	0.10	0.29	300	0.10	0.38
10	150	0.15	0.62	200	0.15	0.76	300	0.13	0.73
10	150	0.20	0.89	200	0.18	0.84	300	0.15	0.86
15	225	0.10	0.27	300	0.10	0.36	450	0.07	0.38
15	225	0.15	0.61	300	0.12	0.69	450	0.10	0.74
15	225	0.18	0.90	300	0.15	0.91	450	0.12	0.82
20	300	0.10	0.34	400	0.10	0.46	600	0.06	0.36
20	300	0.13	0.62	400	0.12	0.78	600	0.08	0.56
20	300	0.17	0.92	400	0.13	0.87	600	0.10	0.82
25	375	0.08	0.39	500	0.08	0.39	750	0.06	0.43
25	375	0.10	0.54	500	0.10	0.77	750	0.08	0.68
25	375	0.15	0.90	500	0.12	0.87	750	0.10	0.94
30	450	0.08	0.40	600	0.08	0.48	900	0.05	0.32
30	450	0.10	0.65	600	0.10	0.70	900	0.08	0.72
30	450	0.12	0.82	600	0.12	0.94	900	0.09	0.90

Table 6.9: Hotelling Simulation (iv)

p	n=15p	hcv	rhcvr	δ	n=20p	hcv	rhcvr	δ	n=30p	hcv	rhcvr	δ
2	30	0.380	0.150	0.30	40	0.367	0.160	0.25	60	0.363	0.180	0.20
2	30	0.615	0.260	0.40	40	0.640	0.314	0.35	60	0.700	0.430	0.30
2	30	0.830	0.422	0.50	40	0.864	0.516	0.45	60	0.921	0.706	0.40
5	75	0.459	0.245	0.20	100	0.366	0.184	0.15	150	0.333	0.208	0.12
5	75	0.682	0.416	0.25	100	0.599	0.368	0.20	150	0.577	0.394	0.16
5	75	0.840	0.588	0.30	100	0.816	0.587	0.30	150	0.860	0.708	0.40
10	150	0.221	0.113	0.10	200	0.312	0.182	0.10	300	0.469	0.340	0.10
10	150	0.621	0.400	0.17	200	0.655	0.467	0.15	300	0.647	0.504	0.12
10	150	0.888	0.729	0.22	200	0.848	0.692	0.18	300	0.872	0.767	0.15
15	225	0.314	0.188	0.10	300	0.442	0.294	0.10	450	0.317	0.228	0.07
15	225	0.714	0.543	0.15	300	0.623	0.449	0.12	450	0.648	0.522	0.10
15	225	0.881	0.738	0.18	300	0.858	0.755	0.15	450	0.853	0.762	0.12
20	300	0.408	0.276	0.10	400	0.341	0.230	0.08	600	0.291	0.216	0.06
20	300	0.691	0.525	0.13	400	0.674	0.534	0.11	600	0.554	0.433	0.08
20	300	0.935	0.852	0.17	400	0.858	0.742	0.13	600	0.790	0.701	0.10
25	375	0.304	0.214	0.08	500	0.434	0.319	0.08	750	0.354	0.266	0.06
25	375	0.728	0.580	0.12	500	0.676	0.531	0.10	750	0.660	0.556	0.08
25	375	0.926	0.837	0.15	500	0.868	0.771	0.12	750	0.887	0.815	0.10
30	450	0.374	0.264	0.08	600	0.395	0.290	0.07	900	0.290	0.217	0.05
30	450	0.602	0.467	0.10	600	0.639	0.517	0.09	900	0.743	0.642	0.08
30	450	0.883	0.763	0.13	600	0.867	0.770	0.11	900	0.876	0.808	0.09

APPENDIX C: ROBUST CCA SIMULATION RESULTS

Table 6.10: Robust CCA Simulation 1 (Categorical Variables Removed)

outlier	method	ra1	ra2	rb1	rb2	Mr1	Mr2	Ma1	Ma2	Mb1	Mb2
0	1	1	1	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
0	2	1	1	0.99	0.95	0.07	0.14	0.00	0.29	0.00	0.29
0	3	1	1	0.99	0.89	1.09	3.60	0.01	0.48	0.01	0.48
0	4	1	1	0.99	0.81	0.12	2.21	0.00	0.56	0.00	0.56
0	5	1	1	1.00	0.91	0.00	0.06	0.00	0.47	0.00	0.47
0	6	1	1	0.55	0.92	207.71	22.22	0.62	0.72	0.62	0.72
0	7	1	1	0.99	0.90	0.08	0.77	0.02	0.42	0.02	0.42
0	8	1	1	0.23	0.55	53.53	46.70	0.96	1.37	0.96	1.37
1	1	1	1	0.92	0.92	17.04	4.15	1.13	1.07	1.13	1.07
1	2	1	1	0.90	0.96	6.73	6.14	1.09	0.68	1.09	0.68
1	3	1	1	0.81	0.67	0.03	0.00	0.68	0.91	0.68	0.91
1	4	1	1	0.70	0.39	0.03	0.02	0.40	1.11	0.40	1.11
1	5	1	1	0.92	0.71	17.04	0.61	1.13	1.17	1.13	1.17
1	6	1	1	0.71	0.69	0.57	0.62	0.73	1.07	0.73	1.07
1	7	1	1	0.99	0.72	3.64	2.16	1.11	1.01	1.11	1.01
1	8	1	1	0.54	0.69	0.00	0.18	0.49	0.91	0.49	0.91
2	1	1	1	0.21	0.72	28.44	1.35	1.27	0.90	1.27	0.90
2	2	1	1	0.90	0.97	10.67	0.21	0.72	0.20	0.72	0.20
2	3	1	1	0.99	0.90	5.58	2.84	0.08	0.55	0.08	0.55
2	4	1	1	0.96	0.91	2.70	2.66	0.00	0.37	0.00	0.37
2	5	1	1	0.21	0.82	28.44	0.03	1.27	0.86	1.27	0.86
2	6	1	1	0.53	0.89	320.79	24.33	0.72	0.83	0.72	0.83
2	7	1	1	0.92	0.91	7.21	0.13	0.91	0.41	0.91	0.41
2	8	1	1	0.97	0.91	0.91	9.29	0.00	0.33	0.00	0.33
3	1	1	1	0.93	0.72	0.69	2.60	1.05	1.05	1.05	1.05
3	2	1	1	0.81	0.93	0.92	4.07	0.89	0.74	0.89	0.74
3	3	1	1	0.81	0.71	3.52	10.60	0.73	0.90	0.73	0.90
3	4	1	1	0.68	0.70	4.95	2.73	0.47	0.79	0.47	0.79

3	5	1	1	0.93	0.68	0.69	0.58	1.05	1.12	1.05	1.12
3	6	1	1	0.49	0.64	317.72	55.97	0.92	1.16	0.92	1.16
3	7	1	1	0.92	0.71	1.35	3.19	1.09	1.09	1.09	1.09
3	8	1	1	0.87	0.75	3.22	16.17	0.96	0.83	0.96	0.83
4	1	1	1	0.43	0.59	0.02	0.77	0.56	1.04	0.56	1.04
4	2	1	1	0.77	0.47	0.13	0.34	0.22	1.08	0.22	1.08
4	3	1	1	0.99	0.81	0.66	2.87	0.26	0.67	0.26	0.67
4	4	1	1	0.92	0.68	0.00	1.29	0.00	0.77	0.00	0.77
4	5	1	1	0.43	0.80	0.02	3.54	0.56	0.97	0.56	0.97
4	6	1	1	0.51	0.77	331.07	38.52	0.79	1.03	0.79	1.03
4	7	1	1	0.66	0.79	0.23	0.09	0.34	0.75	0.34	0.75
4	8	1	1	0.43	0.86	107.15	10.70	1.07	0.87	1.07	0.87

Table 6.11: Robust CCA Simulation 2, cov type=1

cov	sdt	mdt	ra1	ra2	rb1	rb2	Mr1	Mr2	Ma1	Ma2	Mb1	Mb2
1	1	1	1	1	1.00	0.99	0.03	1.10	0.11	0.29	0.11	0.29
1	1	2	1	1	1.00	0.99	0.07	0.99	0.15	0.30	0.15	0.30
1	1	3	1	1	1.00	1.00	0.20	0.39	0.16	0.30	0.16	0.30
1	1	4	1	1	1.00	0.99	0.13	0.62	0.17	0.31	0.17	0.31
1	1	5	1	1	1.00	0.99	0.03	1.09	0.11	0.29	0.11	0.29
1	1	6	1	1	0.88	0.87	4068.37	72.51	0.44	0.50	0.44	0.50
1	1	7	1	1	1.00	0.99	0.13	0.88	0.13	0.30	0.13	0.30
1	1	8	1	1	1.00	0.99	0.75	0.40	0.17	0.31	0.17	0.31
1	2	1	1	1	1.00	0.95	1.55	0.65	0.86	0.88	0.86	0.88
1	2	2	1	1	1.00	0.97	4.42	0.25	0.55	0.51	0.55	0.51
1	2	3	1	1	1.00	0.98	7.59	1.18	0.55	0.44	0.55	0.44
1	2	4	1	1	1.00	0.97	6.05	0.57	0.28	0.00	0.28	0.00
1	2	5	1	1	1.00	0.98	1.55	0.22	0.86	0.88	0.86	0.88
1	2	6	1	1	0.98	0.96	363.32	6.06	0.58	0.46	0.58	0.46
1	2	7	1	1	1.00	0.99	4.47	0.90	0.61	0.58	0.61	0.58
1	2	8	1	1	1.00	0.98	19.03	0.11	0.29	0.00	0.29	0.00
1	3	1	1	1	1.00	0.99	0.35	1.97	0.51	0.51	0.51	0.51
1	3	2	1	1	1.00	1.00	0.07	0.00	0.20	0.00	0.20	0.00
1	3	3	1	1	1.00	0.99	0.09	1.04	0.24	0.02	0.24	0.02
1	3	4	1	1	1.00	1.00	0.03	0.64	0.12	0.00	0.12	0.00
1	3	5	1	1	1.00	0.98	0.35	0.41	0.51	0.47	0.51	0.47
1	3	6	1	1	0.67	0.58	2413.04	280.64	0.89	0.91	0.89	0.91
1	3	7	1	1	1.00	0.99	0.35	0.37	0.22	0.15	0.22	0.15
1	3	8	1	1	1.00	0.99	0.07	0.73	0.12	0.00	0.12	0.00
1	4	1	1	1	1.00	0.66	1430.24	187.04	1.24	0.87	1.24	0.87
1	4	2	1	1	1.00	0.66	1506.14	216.81	1.25	0.85	1.25	0.85
1	4	3	1	1	1.00	0.69	2096.80	362.33	1.32	0.80	1.32	0.80
1	4	4	1	1	1.00	0.99	0.96	5.46	0.22	0.00	0.22	0.00
1	4	5	1	1	1.00	0.68	1430.24	173.17	1.24	0.86	1.24	0.86
1	4	6	1	1	0.96	0.70	73.49	786.32	1.23	0.83	1.23	0.83
1	4	7	1	1	1.00	0.68	1484.25	204.07	1.24	0.85	1.24	0.85

1	4	8	1	1	1.00	0.98	3.83	15.45	0.22	0.94	0.22	0.94
1	5	1	1	1	1.00	0.70	392.18	170.29	1.02	0.89	1.02	0.89
1	5	2	1	1	1.00	0.87	49.81	65.45	0.63	0.65	0.63	0.65
1	5	3	1	1	1.00	1.00	2.93	4.39	0.43	0.28	0.43	0.28
1	5	4	1	1	1.00	1.00	3.28	4.59	0.32	0.00	0.32	0.00
1	5	5	1	1	1.00	0.71	392.18	92.11	1.02	0.93	1.02	0.93
1	5	6	1	1	1.00	1.00	1.37	0.51	0.43	0.28	0.43	0.28
1	5	7	1	1	1.00	0.86	78.96	6.75	0.70	0.74	0.70	0.74
1	5	8	1	1	1.00	1.00	1.89	1.32	0.32	0.00	0.32	0.00
1	6	1	1	1	0.32	0.45	1538.62	27.72	1.45	1.42	1.45	1.42
1	6	2	1	1	0.33	0.48	1531.15	26.87	1.45	1.41	1.45	1.41
1	6	3	1	1	1.00	0.98	0.02	4.01	0.41	0.54	0.41	0.54
1	6	4	1	1	1.00	0.99	0.21	2.46	0.00	0.24	0.00	0.24
1	6	5	1	1	0.32	0.43	1538.62	30.89	1.45	1.44	1.45	1.44
1	6	6	1	1	0.99	0.96	8149.64	24.24	0.42	0.55	0.42	0.55
1	6	7	1	1	0.35	0.37	1550.74	26.78	1.50	1.54	1.50	1.54
1	6	8	1	1	1.00	0.99	0.48	11.49	0.00	0.29	0.00	0.29
1	7	1	1	1	0.98	0.97	1254.89	1.04	0.19	0.13	0.19	0.13
1	7	2	1	1	1.00	1.00	196.15	1.40	0.00	0.00	0.00	0.00
1	7	3	1	1	1.00	1.00	0.10	2.53	0.10	0.27	0.10	0.27
1	7	4	1	1	1.00	1.00	0.32	2.57	0.00	0.12	0.00	0.12
1	7	5	1	1	0.98	0.98	1254.89	1.20	0.19	0.22	0.19	0.22
1	7	6	1	1	0.81	0.75	2506.06	99.60	0.54	0.68	0.54	0.68
1	7	7	1	1	1.00	0.99	49.39	1.06	0.00	0.00	0.00	0.00
1	7	8	1	1	1.00	0.99	0.16	0.92	0.00	0.18	0.00	0.18

Table 6.12: Robust CCA Simulation 2, cov type=2

cov	sdt	mdt	ra1	ra2	rb1	rb2	Mr1	Mr2	Ma1	Ma2	Mb1	Mb2
2	1	1	1.00	1.00	1.00	1.00	0.03	0.58	0.00	0.00	0.00	0.00
2	1	2	1.00	1.00	1.00	1.00	0.05	0.18	0.00	0.00	0.00	0.00
2	1	3	1.00	1.00	1.00	1.00	0.32	0.01	0.00	0.00	0.00	0.00
2	1	4	1.00	1.00	1.00	1.00	0.10	0.00	0.00	0.00	0.00	0.00
2	1	5	1.00	1.00	1.00	1.00	0.03	0.58	0.00	0.00	0.00	0.00
2	1	6	1.00	0.73	1.00	0.81	0.77	277.79	0.00	0.65	0.00	0.65
2	1	7	1.00	1.00	1.00	1.00	0.13	0.02	0.00	0.00	0.00	0.00
2	1	8	1.00	1.00	1.00	1.00	0.94	1.02	0.00	0.00	0.00	0.00
2	2	1	1.00	0.98	1.00	0.97	3.05	0.03	0.87	0.94	0.87	0.94
2	2	2	1.00	1.00	1.00	1.00	5.72	0.13	0.53	0.56	0.53	0.56
2	2	3	1.00	1.00	1.00	1.00	3.62	0.46	0.52	0.52	0.52	0.52
2	2	4	1.00	1.00	1.00	1.00	4.54	0.07	0.29	0.29	0.29	0.29
2	2	5	1.00	0.98	1.00	0.97	3.05	0.03	0.87	0.94	0.87	0.94
2	2	6	0.78	0.84	0.82	0.83	1675.48	96.58	0.88	0.77	0.88	0.77
2	2	7	1.00	1.00	1.00	0.99	11.86	1.20	0.59	0.61	0.59	0.61
2	2	8	1.00	1.00	1.00	0.99	2.13	0.31	0.29	0.34	0.29	0.34
2	3	1	1.00	1.00	1.00	1.00	5.28	0.40	0.60	0.58	0.60	0.58
2	3	2	1.00	1.00	1.00	1.00	0.18	0.12	0.36	0.27	0.36	0.27
2	3	3	1.00	1.00	1.00	0.99	1.05	0.49	0.33	0.29	0.33	0.29
2	3	4	1.00	1.00	1.00	0.99	0.26	0.67	0.23	0.00	0.23	0.00
2	3	5	1.00	1.00	1.00	1.00	5.28	0.40	0.60	0.58	0.60	0.58
2	3	6	0.98	0.92	0.97	0.91	701.73	155.87	0.36	0.43	0.36	0.43
2	3	7	1.00	0.99	1.00	1.00	0.43	0.18	0.38	0.34	0.38	0.34
2	3	8	1.00	1.00	1.00	0.99	0.12	0.61	0.23	0.08	0.23	0.08
2	4	1	1.00	0.73	1.00	0.72	1327.37	130.05	1.24	0.84	1.24	0.84
2	4	2	1.00	0.72	1.00	0.71	1408.11	127.17	1.25	0.84	1.25	0.84
2	4	3	1.00	0.90	1.00	0.89	747.55	37.11	0.74	0.63	0.74	0.63
2	4	4	1.00	1.00	1.00	1.00	0.00	2.20	0.00	0.30	0.00	0.30
2	4	5	1.00	0.73	1.00	0.72	1327.37	130.05	1.24	0.84	1.24	0.84
2	4	6	0.98	0.89	0.98	0.85	30.21	517.26	0.70	0.64	0.70	0.64
2	4	7	1.00	0.74	1.00	0.73	1383.14	126.77	1.24	0.83	1.24	0.83

2	4	8	1.00	1.00	1.00	1.00	0.05	1.35	0.00	0.30	0.00	0.30
2	5	1	1.00	0.70	1.00	0.72	340.35	106.30	0.99	0.87	0.99	0.87
2	5	2	1.00	0.86	1.00	0.86	28.46	22.08	0.52	0.62	0.52	0.62
2	5	3	1.00	1.00	1.00	0.99	0.25	0.59	0.25	0.31	0.25	0.31
2	5	4	1.00	1.00	1.00	1.00	0.37	0.94	0.00	0.05	0.00	0.05
2	5	5	1.00	0.70	1.00	0.72	340.35	106.30	0.99	0.87	0.99	0.87
2	5	6	0.85	0.91	0.84	0.90	1556.93	57.06	0.57	0.45	0.57	0.45
2	5	7	0.95	0.19	0.95	0.30	315.15	770.22	0.98	1.33	0.98	1.33
2	5	8	1.00	1.00	1.00	1.00	0.09	0.00	0.00	0.00	0.00	0.00
2	6	1	0.51	0.51	0.48	0.52	825.86	95.73	1.54	1.53	1.54	1.53
2	6	2	0.50	0.51	0.47	0.51	819.97	97.72	1.55	1.52	1.55	1.52
2	6	3	1.00	1.00	1.00	1.00	0.66	0.34	0.50	0.50	0.50	0.50
2	6	4	1.00	1.00	1.00	1.00	0.54	0.84	0.14	0.18	0.14	0.18
2	6	5	0.51	0.51	0.48	0.52	825.86	95.73	1.54	1.53	1.54	1.53
2	6	6	0.94	0.94	0.93	0.95	349.30	35.51	0.62	0.58	0.62	0.58
2	6	7	0.51	0.52	0.45	0.57	821.94	91.78	1.57	1.53	1.57	1.53
2	6	8	1.00	1.00	1.00	1.00	1.27	0.16	0.14	0.18	0.14	0.18
2	7	1	0.68	0.42	0.74	0.11	874.71	66.67	1.41	1.32	1.41	1.32
2	7	2	1.00	0.99	1.00	0.98	172.37	0.20	0.00	0.00	0.00	0.00
2	7	3	1.00	0.98	1.00	0.97	3.67	0.01	0.00	0.30	0.00	0.30
2	7	4	1.00	0.99	1.00	0.98	3.16	0.08	0.00	0.21	0.00	0.21
2	7	5	0.68	0.42	0.74	0.11	874.71	66.67	1.41	1.32	1.41	1.32
2	7	6	1.00	1.00	1.00	0.99	2.65	1.35	0.00	0.25	0.00	0.25
2	7	7	1.00	0.98	1.00	0.98	54.44	0.32	0.00	0.00	0.00	0.00
2	7	8	1.00	1.00	1.00	0.99	5.49	0.47	0.00	0.14	0.00	0.14

APPENDIX D: OUTLIER RESISTANCE SIMULATION RESULT

dt: outlier type, F: FCH, RF: RFCH, C: CMVE, RC: RCMVE

Table 6.13: Outlier Resistance Simulation

n	p	dt	pm	F	RF	C	RC	MB	n	p	dt	pm	F	RF	C	RC	MB
50	2	1	6	74	34	74	34	97	200	50	3	10	1	1	28	28	95
50	2	1	7	96	74	96	74	99	200	50	3	33	92	92	88	88	100
50	5	1	13	92	90	92	90	93	200	55	3	11	0	0	31	31	96
50	10	1	22	91	91	91	91	91	200	55	3	35	84	84	90	90	100
50	15	1	31	90	87	90	87	90	200	60	3	11	0	0	36	36	95
100	20	1	36	92	90	92	90	92	200	60	3	37	92	92	96	96	100
100	25	1	45	91	91	91	91	91	200	65	3	11	0	0	27	27	91
100	30	1	55	92	89	92	89	92	200	65	3	38	82	82	90	90	100
150	35	1	60	87	90	87	90	87	250	70	3	11	0	0	26	26	91
150	40	1	65	93	91	93	91	93	250	70	3	38	76	76	91	91	100
150	45	1	77	95	96	95	96	95	250	75	3	12	0	0	30	30	93
200	50	1	78	92	89	92	89	92	250	75	3	39	72	72	91	91	100
200	55	1	89	92	91	92	91	92	250	80	3	12	0	0	40	40	99
200	60	1	99	92	92	92	92	92	250	80	3	40	75	75	90	90	100
200	65	1	106	90	87	90	87	90	300	85	3	12	0	0	32	32	94
250	75	1	117	92	92	92	92	92	300	90	3	12	0	0	32	32	90
250	85	1	138	92	92	92	92	92	300	95	3	13	0	0	26	26	99
300	90	1	139	92	91	92	91	92	350	100	3	13	0	0	38	38	95
350	100	1	148	92	93	92	93	92	50	2	4	6	72	41	45	35	94
50	2	2	6	80	62	80	62	98	50	2	4	7	91	61	61	52	98
50	2	2	7	100	99	100	99	100	50	5	4	12	43	40	19	18	91
50	5	2	7	25	25	28	28	97	50	5	4	17	92	94	44	44	98
50	5	2	9	100	100	100	100	100	50	10	4	24	30	32	15	15	92
50	10	2	11	2	2	96	96	96	50	10	4	35	90	90	73	73	100
50	15	2	15	0	0	99	99	99	50	15	4	35	32	30	24	25	92
100	20	2	19	0	0	100	100	100	50	15	4	55	91	91	89	89	98
100	25	2	23	0	0	100	100	100	100	20	4	38	8	8	8	8	92

100	30	2	27	0	0	99	99	99
150	35	2	30	0	0	100	100	100
150	40	2	33	0	0	96	96	96
150	45	2	38	0	0	98	98	98
200	55	2	46	0	0	99	99	99
200	60	2	49	0	0	97	97	97
200	65	2	53	0	0	99	99	99
250	70	2	57	0	0	99	99	99
250	75	2	60	0	0	95	95	95
250	80	2	64	0	0	99	99	99
300	85	2	68	0	0	100	100	100
300	90	2	71	0	0	94	94	94
300	95	2	75	0	0	99	99	99
350	100	2	78	0	0	91	91	91
50	2	3	5	93	65	87	64	94
50	5	3	6	95	95	49	49	99
50	10	3	10	90	90	42	42	100
50	15	3	8	47	47	39	39	98
50	15	3	16	90	90	71	71	100
100	20	3	16	93	93	60	60	100
100	25	3	9	26	26	32	32	99
100	25	3	22	91	91	75	75	100
100	30	3	9	9	9	33	33	91
100	30	3	26	86	86	90	90	100
100	35	3	29	94	94	90	90	100
150	35	3	9	10	10	22	22	96
150	35	3	26	90	90	82	82	100
150	40	3	10	5	5	35	35	98
150	40	3	30	93	93	91	91	100
150	45	3	10	1	1	29	29	91
150	45	3	32	90	90	89	89	100
100	20	4	65	94	94	88	88	100
100	25	4	47	5	5	13	11	92
100	25	4	77	91	91	84	84	100
100	30	4	60	7	7	20	20	93
150	35	4	57	1	1	9	9	91
150	35	4	105	89	89	91	91	100
150	40	4	69	0	0	14	14	93
150	40	4	111	86	86	91	91	100
150	45	4	81	0	0	17	16	92
150	45	4	125	85	95	92	92	100
200	50	4	80	0	0	11	11	93
200	50	4	135	91	91	92	92	100
200	55	4	85	0	0	9	9	91
200	55	4	149	90	90	93	93	100
200	60	4	101	0	0	14	14	94
200	60	4	158	87	87	93	93	100
200	65	4	109	0	0	24	24	93
200	65	4	166	88	88	90	90	100
250	70	4	108	0	0	11	11	93
250	75	4	116	0	0	13	13	92
250	75	4	190	87	87	90	90	100
250	80	4	128	0	0	13	12	93
250	80	4	201	88	88	92	92	100
300	85	4	126	0	0	12	12	96
300	90	4	133	0	0	18	18	92
300	90	4	223	89	89	90	90	100
300	95	4	144	0	0	14	14	91
300	95	4	235	91	91	91	91	100
350	100	4	138	0	0	10	10	97
350	100	4	240	85	85	90	90	100

APPENDIX E: PROGRAM FOR ROBUST CCA SIMULATION

```
########################################################################
# Funtion: invtanh                                                    #
# Description:  Fisher Transformation. turn a distribution of          #
#               correlation coeffcients towards normality              #
# Variables: rho <- correlation                                        #
########################################################################
"invtanh" <- function (rho)
{
    0.5 * log((1 + rho)/(1 - rho))
}


########################################################################
# Function: ccacov                                                     #
# Description: compute canonical correlation and covariats from        #
#              a covariance matrix                                     #
# Variables:  mx <- covariance matrix                                  #
#             p <- number of first group of variables                  #
#             q <- number of second group of variables                 #
########################################################################
"ccacov" <- function(mx=mx, p=1, q=1) {
  mx <- as.matrix(mx)
  if (dim(mx)[1]!=dim(mx)[2]) {
    cat ("must input a positive definite covariance matrix\n")
    return(-1)
  }
  else if (dim(mx)[1]!= (p+q)) {
    cat ("dimensions not match\n")
    return(-1)
  }
  ##partition mx
  S11 <- mx[1:p, 1:p]
  S22 <- mx[(p+1):dim(mx)[1], (p+1):dim(mx)[1]]
  S12 <- mx[1:p, (p+1):dim(mx)[1]]
  S21 <- t(S12)
  ##get eigenvalues and eigen vectors
  EigS11 <- eigen(S11)
  EigS22 <- eigen(S22)
  InvS11 <- solve(S11)     #S11^(-1)
  InvS22 <- solve(S22)
  ##get S11^(-1/2) and S22^(-1/2)
  InvRtS11 <- EigS11$vec %*% diag(1/sqrt(EigS11$val)) %*% t(EigS11$vec)
  InvRtS22 <- EigS22$vec %*% diag(1/sqrt(EigS22$val)) %*% t(EigS22$vec)
  Ems <- InvRtS11 %*% S12 %*% InvS22 %*% S21 %*% InvRtS11
  Fms <- InvRtS22 %*% S21 %*% InvS11 %*% S12 %*% InvRtS22
  CcaE <- eigen(Ems)
  CcaF <- eigen(Fms)
  CcaCor <- sqrt(CcaE$val)
  CcaCovE <- t(InvRtS11)  %*%  CcaE$vec
  CcaCovF <- t(InvRtS22)  %*%  CcaF$vec
  list(cor=CcaCor, xcoef=CcaCovE, ycoef=CcaCovF)
}


########################################################################
# Function:  ccadata1                                                  #
# Description: add outliers to "ucla" clean dataset                    #
# Variables:  outlier <- outlier types                                 #
#             x <- original clean data                                 #
# Comment:  no outliers added if the outlier type is not 1,2,3,4       #
########################################################################
"ccadata1" <- function(x, outlier=outlier) {
x <- as.matrix(x)
xcov <- cov(x)
xm <- apply(x, 2, mean)
unf <- runif(dim(x)[1])            #get uniform dist
dx <- x

if (outlier == 1)
  dx[unf < .3, ] <- x[unf < .3, ]*3
else if (outlier == 2)
  dx[unf < .1, ] <- x[unf < .1, ]*3
else if (outlier == 3) {
  outnorm <- mvrnorm(n=600, xm, 5*xcov)
  dx[unf < .3, ] <- outnorm[unf < .3, ]
}
else if ( outlier == 4) {
  outnorm <- mvrnorm(n=600, xm, 5*xcov)
  dx[unf < .1, ] <- outnorm[unf < .1, ]
}
return(dx)
}


########################################################################
# Function:  ccadata2                                                  #
# Description:   generate data based on covariance matrices            #
# Variables:    n <- sample size                                       #
#               stp <- covariance matrix type                          #
#               dtp <- distribution type                               #
# Comment:    output includes the data and benchmark                   #
########################################################################
"ccadata2" <- function(n=n, stp=stp, dtp=dtp) {
if (stp == 1) {
  sgm12 <- cbind(diag(c(.9, .3)), matrix(rep(0,4),nrow=2, ncol=2))
  sgm11 <- diag(2)
  sgm22 <- diag(4)
  p <- 2
  q <- 4
}
else if (stp==2) {
  sgm12 <- diag(c(.9, .5, .2))
  sgm11 <- diag(3)
  sgm22 <- diag(3)
  p <- 3
  q <- 3
}
else {
  sgm12 <- diag(c(.9, .7, .4, .3, .1))
  sgm11 <- diag(5)
  sgm22 <- diag(5)
  p <- 5
  q <- 5
}
sgm <- rbind(cbind(sgm11, sgm12), cbind(t(sgm12), sgm22))
bm <- ccacov(sgm, p=p, q=q)
dx <- mvrnorm(n=n, rep(0,p+q), sgm)
unf <- runif(n)

if (dtp == 2) {
  xo <- mvrnorm(n=n, rep(0,p+q), 8*sgm)
  dx[unf < .2, ] <- xo[unf < .2, ]
}
else if (dtp == 3) {
  xo <- mvrnorm(n=n, rep(0,p+q), 8*sgm)
  dx[unf < .05, ] <- xo[unf < .05, ]
}
else if (dtp == 4)
  dx[unf < .2, ] <- sum(diag(sgm))
else if (dtp == 5)
  dx[unf < .05, ] <- sum(diag(sgm))
else if (dtp == 6)
  dx[unf < .2, 1] <- 2*sum(diag(sgm))
else if (dtp == 7)
  dx[unf < .05, 1] <- 2*sum(diag(sgm))
list(dx=dx, p=p, q=q, bm=bm)
}


#########################################################################
# Function:  ccarun                                                     #
# Description: output canonical covariates and coefficients             #
#                   based on robust cca methods                         #
# Variables:  dx <- contaminated dataset                                #
#             est <- type of robust cca estimator                       #
#             p <- number of first group of variables                   #
#             q <- number of second group of variables                  #
# Comment:  there are eight different robust cca methods                #
#########################################################################
```

```
"ccarun" <- function(dx=dx, est=est, p=p, q=q) {
dx <- as.matrix(dx)

if ( est == 1) {
  dxcov <- cov(dx)            #classical cov estimator
  outdx <- ccacov(dxcov, p=p, q=q)
}
else if (est == 2) {
  dxcov <- mesthub(dx)$cov     #Huber cov M-estimator
  outdx <- ccacov(dxcov, p=p, q=q)
}
else if (est == 3) {
  dxcov <- covMcd(dx)$cov     #F-MCD cov estimator
  outdx <- ccacov(dxcov, p=p, q=q)
}
else if (est == 4) {
  dxcov <- covrmvn(dx)$cov     #Huber cov M-estimator
  outdx <- ccacov(dxcov, p=p, q=q)
}
else if (est == 5)
  outdx <- pp(p=p,q=q,dx, inds=1,indp=1)      #PP-Classical
else if (est ==6)
  outdx <- pp(p=p,q=q,dx, inds=2,indp=2)      #PP-FMCD
else if (est == 7)
  outdx <- pp(p=p,q=q,dx, inds=3,indp=3)      #PP-M
else if (est == 8)
  outdx <- pp(p=p,q=q,dx, inds=4,indp=4)      #PP-RMVN

return(outdx)
}


#########################################################################
# Function: ccaind                                                      #
# Description: compute MSE of canonical correlation and covariates,    #
#                   compute correlation as robustness measure          #
# Variables:  dx <- contaminated dataset                                #
#             nrun <- number of replications                            #
#             est <- type of robust cca estimator                      #
#             phrb <- Fisher Transformation of correlation benchmark   #
#             p <- number of first group of variables                  #
#             q <- number of second group of variables                 #
#             bm <- benchmark ("ture value")                           #
# Comment:  MSE and correlation are both used as measures of robustness#
#               and correlation measure only works when p>2            #
########################################################################
"ccaind" <- function(dx=dx, nrun=nrun, est=est,
                    phrb=phrb, p=p, q=q, bm=bm)
{
MSEr <- rep(0, p)
alfr <- MSEr
betr <- MSEr
alfrt <- MSEr
betrt <- MSEr
MSEca <- MSEr
MSEcb <- MSEr
MSEcat <- MSEr
MSEcbt <- MSEr
bmx <- bm$xcoef
bmy <- bm$ycoef
for (i in 1:nrun) {
  outdx <- ccarun(dx=dx, est=est, p=p, q=q)
  phri <- invtanh(outdx$cor)
  MSEr <- MSEr + (phri - phrb)^2
  cvcx <- outdx$xcoef
  cvcy <- outdx$ycoef
  for (k in 1:p) {
    alfr[k] <- cor(bmx[,k], cvcx[,k])
    betr[k] <- cor(bmy[,k], cvcy[,k])
    knanx <- abs(t(bmy[,k]) %*% cvcy[,k]) /
              (vecnorm(bmy[,k]) * vecnorm(bmy[,k]))
    knany <- abs(t(bmy[,k]) %*% cvcy[,k]) /
              (vecnorm(bmy[,k]) * vecnorm(bmy[,k]))
    if (knanx > 1)  MSEca[k]<-0
    else  MSEca[k] <- acos(knanx)
    if (knany > 1)  MSEcb[k]<-0
    else MSEcb[k] <- acos(knany)
  }
  MSEcat <- MSEcat + MSEca
  MSEcbt <- MSEcbt + MSEcb
  alfrt <- alfrt + abs(alfr)
  betrt <- betrt + abs(betr)
}
MSEr <- MSEr / nrun
MSEcat <- MSEcat / nrun
MSEcbt <- MSEcbt / nrun
alfrt <- alfrt / nrun
betrt <- betrt / nrun
list(MSEr=MSEr, alfr=alfrt, betr=betrt, Mcat=MSEcat, Mcbt=MSEcbt)
}


########################################################################
# Function: ccasim1                                                    #
# Description: input dataset and output results                        #
# Variables:  nrun <- number of replications                           #
#             nout <- number of types of outliers                      #
#             nest <- number of robust cca methods                     #
#             p <- number of first group of variables                  #
#             q <- number of second group of variables                 #
# Comment:    categorical variable, gender, is removed for             #
#                    a better performance of the F-MCD estimator       #
########################################################################
"ccasim1" <- function(nrun=nrun, nout=4, nest=8, p=3, q=4) {
setwd("c:/work/MyDt0627/sim")
source("rpack.txt")
source("ccapp.r")

mm <- read.table("mmreg.csv", sep = ",", header = TRUE)

x <- mm[, 1:7]
bm <- ccarun(x, est=1, p=p, q=q)                #get benchmark
phrb <- invtanh(bm$cor)
```

```
cat("--------------------", date(), "--------------------------\n",
      file="ccasim1.txt", append=T)
for (i in 0:nout)
{
  dx <- ccadata1(x, outlier=i)                  #get contaminated data
  for (j in 1:nest)
  {
    out <- ccaind(dx=dx, nrun=nrun, est=j, phrb=phrb,
              p=p, q=q, bm=bm)                  #get covariates and corrs
    cat("outlier=", i, "  est=", j, "  MSEr=", out$MSEr, "\n")
    cat(i, j, " MSEr ", out$MSEr, " alfr ", out$alfr, " betr ", out$betr,
      " MSEca", out$Mcat, "MSEcb", out$Mcbt, "\n",
      file="ccasim1.txt", append=T)
  }
}
cat("--------------------", date(), "--------------------------\n",
     file="ccasim1.txt", append=T)
}


########################################################################
# Function: ccasim11                                                   #
# Description: input dataset and output results                        #
# Variables:  nrun <- number of replications                           #
#             nout <- number of types of outliers                      #
#             nest <- number of robust cca methods                     #
#             p <- number of first group of variables                  #
#             q <- number of second group of variables                 #
# Comment:  two categorical variables, gender and motivation are       #
#             removed for a better performance of the F-MCD estimator  #
########################################################################
"ccasim11" <- function(nrun=nrun, nout=4, nest=8, p=2, q=4) {
setwd("c:/work/MyDt0627/sim")
source("rpack.txt")
source("ccapp.r")

mm <- read.table("mmreg.csv", sep = ",", header = TRUE)
```

```
x <- mm[, c(1,2,4,5,6,7)]
bm <- ccarun(x, est=1, p=p, q=q)                #get the benchmark
phrb <- invtanh(bm$cor)

cat("--------------------", date(), "--------------------------\n",
      file="ccasim11.txt", append=T)
for (i in 0:nout)
{
  dx <- ccadata1(x, outlier=i)                  #get contaminated data
  for (j in 1:nest)
  {
    out <- ccaind(dx=dx, nrun=nrun, est=j, phrb=phrb,
              p=p, q=q, bm=bm)                  #get covariates and corrs
    cat("outlier=", i, "  est=", j, "  MSEr=", out$MSEr, "\n")
    cat(i, j, " MSEr ", out$MSEr, " alfr ", out$alfr, " betr ", out$betr,
      " MSEca", out$Mcat, "MSEcb", out$Mcbt, "\n",
      file="ccasim11.txt", append=T)
  }
}
cat("--------------------", date(), "--------------------------\n",
     file="ccasim11.txt", append=T)
}


########################################################################
# Function: ccasim2                                                    #
# Description: Data are created based on given convariance matirx,     #
#                then apply robust cca on the generated data           #
# Variables:  nrun <- number of replications                           #
#             n <- sample size                                         #
# Comment: Monte Carlo study (3 covriance matrices,                    #
#                   7 distributions, and 8 robust cca methods          #
########################################################################
"ccasim2" <- function(nrun=1, n=100) {
setwd("c:/work/MyDt0627/sim")
source("rpack.txt")
source("ccapp.r")
cat("--------------------", date(), "--------------------------\n",
      file="ccasim2.txt", append=T)
for (i in 1:3)
{
  for (j in 1:7) {
    out <- ccadata2(n=n, stp=i, dtp=j)
    bm <- out$bm          #get the benchmark
    phrb <- invtanh(bm$cor)
    dx <- out$dx          #get contaminated data
    p <- out$p
    q <- out$q
    for (k in 1:8)
    {
      out <- ccaind(dx=dx, nrun=nrun, est=k, phrb=phrb,
              p=p, q=q, bm=bm)                  #get covariates and corrs
      cat("stp=", i, "  dtp=", j, "  est=", k, "  MSEr=", out$MSEr, "\n")
      cat(i, j, k, " MSEr ", out$MSEr, " alfr ", out$alfr, " betr ",
          out$betr, " MSEca", out$Mcat, "MSEcb", out$Mcbt, "\n",
          file="ccasim2.txt", append=T)
    }
  }
}
cat("--------------------", date(), "--------------------------\n",
          file="ccasim2.txt", append=T)
}
```

VITA

Graduate School
Southern Illinois University

JIANFENG ZHANG Date of Birth: Oct 28, 1973

4103 Dayton Blvd, Chattanooga, TN 37415

Tennessee Technological University
Master of Science, Mathematics, Dec 2005

Special Honors and Awards: Stanley Dolzychi Memorial Scholarship

Research Thesis Title:
Statistical Data Mining Methods in Adverse Drug Reaction Database

Beijing Institute of Technology
Bachelor of Science, Applied Mathematics, July 1996

Research Paper Title:
Applications of Central Limit Theorem