5-11-2018

Outlier Dectection For High Dimensional Data

Handong Wang
wanghandong1992@siu.edu

OUTLIER DETECTION FOR HIGH DIMENSIONAL DATA

by

Handong Wang

B.A., Southern Illinois University Carbondale, 2015

A Research Paper
Submitted in Partial Fulfillment of the Requirements for the
Master of Science

Department of Mathematics
in the Graduate School
Southern Illinois University Carbondale
May, 2018

RESEARCH PAPER APPROVAL

OUTLIER DETECTION FOR HIGH DIMENSIONAL DATA

by

Handong Wang

A Research Paper Submitted in Partial

Fulfillment of the Requirements

for the Degree of

Master of Science

in the field of Mathematics

Approved by:

David J. Olive

Jerzy Kocik

Yaser Samadi

Graduate School
Southern Illinois University Carbondale
March 5, 2018

AN ABSTRACT OF THE RESEARCH PAPER OF

HANDONG WANG, for the Master of Science degree in MATHEMATICS, presented on MARCH 5, 2018, at Southern Illinois University Carbondale.

TITLE: OUTLIER DETECTION FOR HIGH DIMENSIONAL DATA

MAJOR PROFESSOR: Dr. David J. Olive

This paper presents outlier detection for a data set $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ where $\boldsymbol{x}_i$ is $p \times 1$ and p could be larger than n.

KEY WORDS: FCH, Outliers, RMVN.

ACKNOWLEDGMENTS

First of all, I would like to take this chance to thank Prof. David Olive help me complete my master research paper. Thank you for your assistance and patience. I also want to thank Prof. Kocik and Prof. Samadi for being my committee. Secondly, I want to thank all the professors from math department for their academic instructions and care over the past six years. Finally I want to say, I can not have this achievement without my family's support and encouragement. Thank you all again from the bottom of my heart!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Suppose the data set is $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ where $\boldsymbol{x}_i$ is $p \times 1$. Outliers are cases that lie far away from the bulk of the data, and outliers can ruin a statistical analysis. This paper discusses a technique for outlier detection that works well for certain outlier configurations provided the bulk of the data consists of more than $n/2$ cases. The technique could fail if there are $g > 2$ groups of about n/g cases per group. First we need to define Mahalanobis distances and the coordinatewise median. Some univariate estimators will be defined first.

The *location model* is

$$Y_i = \mu + e_i, \quad i = 1, \ldots, n \tag{1.1}$$

where $e_1, ..., e_n$ are error random variables, often independent and identically distributed (iid) with zero mean. The location model is used when there is one variable Y, such as height, of interest. The location model is a special case of the multivariate location and dispersion model, where there are p variables $x_1, ..., x_p$ of interest, such as height and weight if $p = 2$.

The location model is often summarized by obtaining point estimates and confidence intervals for a location parameter and a scale parameter. Assume that there is a sample $Y_1, \ldots, Y_n$ of size n where the Y_i are iid from a distribution with median MED(Y), mean $E(Y)$, and variance $V(Y)$ if they exist. The location parameter μ is often the population mean or median while the scale parameter is often the population standard deviation $\sqrt{V(Y)}$. The ith *case* is Y_i.

Point estimation is one of the oldest problems in statistics and four important statistics for the location model are the sample mean, median, variance, and the median absolute deviation (MAD). Let $Y_1, \ldots, Y_n$ be the random sample; i.e., assume that $Y_1, ..., Y_n$ are iid. The sample mean is a measure of location and estimates the popula-

tion mean (expected value) $\mu = E(Y)$.

The *sample mean*

$$\overline{Y} = \frac{\sum_{i=1}^{n} Y_i}{n}. \tag{1.2}$$

The *sample median*

$$\text{MED}(n) = Y_{((n+1)/2)} \text{ if n is odd,} \tag{1.3}$$

$$\text{MED}(n) = \frac{Y_{(n/2)} + Y_{((n/2)+1)}}{2} \text{ if n is even.}$$

The notation $\text{MED}(n) = \text{MED}(Y_1, ..., Y_n)$ will also be used.

The *sample variance*

$$S_n^2 = \frac{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}{n-1} = \frac{\sum_{i=1}^{n} Y_i^2 - n(\overline{Y})^2}{n-1}, \tag{1.4}$$

and the *sample standard deviation* $S_n = \sqrt{S_n^2}$.

If the data $Y_1, ..., Y_n$ is arranged in ascending order from smallest to largest and written as $Y_{(1)} \leq \cdots \leq Y_{(n)}$, then $Y_{(i)}$ is the ith order statistic and the $Y_{(i)}$'s are called the *order statistics*. If the data $Y_1 = 1, Y_2 = 4, Y_3 = 2, Y_4 = 5$, and $Y_5 = 3$, then $\overline{Y} = 3$, $Y_{(i)} = i$ for $i = 1, ..., 5$ and $\text{MED}(n) = 3$ where the sample size $n = 5$. The sample median is a measure of location while the sample standard deviation is a measure of scale. The sample mean and standard deviation are vulnerable to outliers, while the sample median and MAD, defined below, are outlier resistant.

The *sample median absolute deviation* is

$$\text{MAD}(n) = \text{MED}(|Y_i - \text{MED}(n)|, \ i = 1, \ldots, n). \tag{1.5}$$

Since $\text{MAD}(n)$ is the median of n distances, at least half of the observations are within a distance $\text{MAD}(n)$ of $\text{MED}(n)$ and at least half of the observations are a distance of $\text{MAD}(n)$ or more away from $\text{MED}(n)$.

Example 1. Let the data be $1, 2, 3, 4, 5, 6, 7, 8, 9$. Then $\text{MED}(n) = 5$ and $\text{MAD}(n) = 2 = \text{MED}\{0, 1, 1, 2, 2, 3, 3, 4, 4\}$.

CHAPTER 2

OUTLIER DETECTION WITH MAHALANOBIS DISTANCE

Now suppose the multivariate data has been collected into an $n \times p$ matrix

$$\boldsymbol{W} = \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_n^T \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,p} \\ x_{2,1} & x_{2,2} & \dots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \dots & x_{n,p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_p \end{bmatrix}$$

where the ith row of $\boldsymbol{W}$ is the ith case $\boldsymbol{x}_i^T$ and the jth column $\boldsymbol{v}_j$ of $\boldsymbol{W}$ corresponds to n measurements of the jth random variable X_j for $j = 1, ..., p$. Hence the n rows of the data matrix $\boldsymbol{W}$ correspond to the n cases, while the p columns correspond to measurements on the p random variables $X_1, ..., X_p$. For example, the data may consist of n visitors to a hospital where the $p = 2$ variables *height* and *weight* of each individual were measured.

The *coordinatewise median* $\text{MED}(\boldsymbol{W}) = (\text{MED}(X_1), ..., \text{MED}(X_p))^T$ where $\text{MED}(X_i)$ is the sample median of the data in column i corresponding to variable X_i and $\boldsymbol{v}_i$.

Example 2. Let the data for X_1 be $1, 2, 3, 4, 5, 6, 7, 8, 9$ while the data for X_2 is $7, 17, 3, 8, 6, 13, 4, 2, 1$. Then $\text{MED}(\boldsymbol{W}) = (\text{MED}(X_1), \text{MED}(X_2))^T = (5, 6)^T$.

For multivariate data, sample Mahalanobis distances play a role similar to that of residuals in multiple linear regression. Let the observed training data be collected in an $n \times p$ matrix $\boldsymbol{W}$. Let the $p \times 1$ column vector $T = T(\boldsymbol{W})$ be a multivariate location estimator, and let the $p \times p$ symmetric positive definite matrix $\boldsymbol{C} = \boldsymbol{C}(\boldsymbol{W})$ be a dispersion estimator.

Let $x_{1j}, ..., x_{nj}$ be measurements on the ith random variable X_j corresponding to

the jth column of the data matrix $\boldsymbol{W}$. The jth *sample mean* is $\overline{x}_j = \frac{1}{n}\sum_{k=1}^{n} x_{kj}$. The *sample covariance* S_{ij} estimates $\mathrm{Cov}(X_i, X_j) = \sigma_{ij}$, and

$$S_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}(x_{ki} - \overline{x}_i)(x_{kj} - \overline{x}_j).$$

$S_{ii} = S_i^2$ is the *sample variance* that estimates the population variance $\sigma_{ii} = \sigma_i^2$. The *sample correlation* r_{ij} estimates the population correlation $\mathrm{Cor}(X_i, X_j) = \rho_{ij}$, and

$$r_{ij} = \frac{S_{ij}}{S_i S_j} = \frac{S_{ij}}{\sqrt{S_{ii}S_{jj}}} = \frac{\sum_{k=1}^{n}(x_{ki} - \overline{x}_i)(x_{kj} - \overline{x}_j)}{\sqrt{\sum_{k=1}^{n}(x_{ki} - \overline{x}_i)^2}\sqrt{\sum_{k=1}^{n}(x_{kj} - \overline{x}_j)^2}}.$$

The sample mean or *sample mean vector*

$$\overline{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i = (\overline{x}_1, ..., \overline{x}_p)^T = \frac{1}{n}\boldsymbol{W}^T\mathbf{1}$$

where $\mathbf{1}$ is the $n \times 1$ vector of ones. The sample covariance matrix

$$\boldsymbol{S} = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T = (S_{ij}).$$

That is, the ij entry of $\boldsymbol{S}$ is the sample covariance S_{ij}. The *classical estimator of multivariate location and dispersion* is $(T, \boldsymbol{C}) = (\overline{\boldsymbol{x}}, \boldsymbol{S})$.

Rule of Thumb. Multivariate procedures start to give good results for $n \geq 10p$, especially if the distribution is close to multivariate normal. In particular, we want $n \geq 10p$ for the sample covariance and matrix. For procedures with large sample theory on a large class of distributions, for any value of n, there are always distributions where the results will be poor, but will eventually be good for larger sample sizes. This rule of thumb is much like the rule of thumb that says the central limit theorem normal approximation for $\overline{Y}$ starts to be good for many distributions for $n \geq 30$.

The ith *Mahalanobis distance* $D_i = \sqrt{D_i^2}$ where the ith *squared Mahalanobis distance* is

$$D_i^2 = D_i^2(T(\boldsymbol{W}), \boldsymbol{C}(\boldsymbol{W})) = (\boldsymbol{x}_i - T(\boldsymbol{W}))^T\boldsymbol{C}^{-1}(\boldsymbol{W})(\boldsymbol{x}_i - T(\boldsymbol{W})) \tag{2.1}$$

for each point $\boldsymbol{x}_i$. Notice that D_i^2 is a random variable (scalar valued).

Let $(T, \boldsymbol{C}) = (T(\boldsymbol{W}), \boldsymbol{C}(\boldsymbol{W}))$. Then

$$D^2_{\boldsymbol{x}}(T, \boldsymbol{C}) = (\boldsymbol{x} - T)^T \boldsymbol{C}^{-1} (\boldsymbol{x} - T).$$

Hence D_i^2 uses $\boldsymbol{x} = \boldsymbol{x}_i$.

Notice that if $\boldsymbol{x}$ is a random vector, then the population squared Mahalanobis distance is

$$D^2_{\boldsymbol{x}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \tag{2.2}$$

and that the term $\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{x} - \boldsymbol{\mu})$ is the $p-$dimensional analog to the z-score used to transform a univariate $N(\mu, \sigma^2)$ random variable into a $N(0, 1)$ random variable. Hence the sample Mahalanobis distance $D_i = \sqrt{D_i^2}$ is an analog of the absolute value $|Z_i|$ of the sample Z-score $Z_i = (X_i - \overline{X})/\hat{\sigma}$. Also notice that the Euclidean distance of $\boldsymbol{x}_i$ from the estimate of center $T(\boldsymbol{W})$ is $D_i(T(\boldsymbol{W}), \boldsymbol{I}_p)$ where $\boldsymbol{I}_p$ is the $p \times p$ identity matrix.

Most outlier detection methods work best if $n \geq 20p$, and often robust estimators $(T, \boldsymbol{C})$ are used with Mahalanobis distances. Olive (2017a) is a good reference. The FCH and RMVN estimators are fairly fast and have some large sample theory.

Often data sets have $p > n$, and outliers are a major problem. The Olive (2017a, § 4.7) `covmb2` estimator is useful and defined below. Also see Olive (2017b, § 1.3). One of the simplest outlier detection methods uses the squared Euclidean distances of the $\boldsymbol{x}_i$ from the coordinatewise median $D_i^2 = D_i^2(\text{MED}(\boldsymbol{W}), \boldsymbol{I}_p)$. Concentration type steps compute the weighted median MED_j, the coordinatewise median computed from the cases $\boldsymbol{x}_i$ with $D_i^2 \leq \text{MED}(D_i^2(\text{MED}_{j-1}, \boldsymbol{I}_p))$ where $\text{MED}_0 = \text{MED}(\boldsymbol{W})$. We often used $j = 0$ (no concentration type steps) or $j = 9$. Let $D_i = D_i(\text{MED}_j, \boldsymbol{I}_p)$. Let $W_i = 1$ if $D_i \leq \text{MED}(D_1, ..., D_n) + k\text{MAD}(D_1, ..., D_n)$ where $k \geq 0$ and $k = 5$ is the default choice. Let $W_i = 0$, otherwise. Using $k \geq 1$ insures that at least half of the cases get weight 1. This weighting corresponds to the weighting that would be used in a one sided metrically trimmed mean (Huber type skipped mean) of the distances.

Application 1. This outlier resistant regression method uses terms from the follow-

ing definition.

Let the ith case $\boldsymbol{w}_i = (Y_i, \boldsymbol{x}_i)^T$ where the continuous predictors from $\boldsymbol{x}_i$ are denoted by $\boldsymbol{u}_i$ for $i = 1, ..., n$. Apply the `covmb2` estimator to the $\boldsymbol{u}_i$, and then run the regression method on the m cases $\boldsymbol{w}_i$ corresponding to the `covmb2` set B indices $i_1, ...i_m$, where $m \geq n/2$.

Definition 1. Let the `covmb2 set` B of at least $n/2$ cases correspond to the cases with weight $W_i = 1$. Then the `covmb2` estimator $(T, \boldsymbol{C})$ is the sample mean and sample covariance matrix applied to the cases in set B. Hence

$$T = \frac{\sum_{i=1}^n W_i \boldsymbol{x}_i}{\sum_{i=1}^n W_i} \text{ and } \boldsymbol{C} = \frac{\sum_{i=1}^n W_i (\boldsymbol{x}_i - T)(\boldsymbol{x}_i - T)^T}{\sum_{i=1}^n W_i - 1}.$$

Example 3. Let the clean data (nonoutliers) be i $\mathbf{1}$ for $i = 1, 2, 3, 4$, and 5 while the outliers are j $\mathbf{1}$ for $j = 16, 17, 18$, and 19. Here $n = 9$ and $\mathbf{1}$ is $p \times 1$. Making a plot of the data for $p = 2$ may be useful. Then the coordinatewise median $\text{MED}_0 = \text{MED}(\boldsymbol{W}) = 5$ $\mathbf{1}$. The median Euclidean distance of the data is the Euclidean distance of 5 $\mathbf{1}$ from 1 $\mathbf{1}$ = the Euclidean distance of 5 $\mathbf{1}$ from 9 $\mathbf{1}$. The *median ball* is the hypersphere centered at the coordinatewise median with radius $r = \text{MED}(D_i(\text{MED}(\boldsymbol{W}), \boldsymbol{I}_p))$ that tends to contain $(n+1)/2$ of the cases if n is odd. Hence the clean data are in the median ball and the outliers are outside of the median ball. The coordinatewise median of the cases with the 5 smallest distances is the coordinatewise median of the clean data: $\text{MED}_1 = 3$ $\mathbf{1}$. Then the median Euclidean distance of the data from MED_1 is the Euclidean distance of 3 $\mathbf{1}$ from 1 $\mathbf{1}$ = the Euclidean distance of 3 $\mathbf{1}$ from 5 $\mathbf{1}$. Again the clean cases are the cases with the 5 smallest Euclidean distances. Hence $\text{MED}_j = 3$ $\mathbf{1}$ for $j \geq 1$. For $j \geq 1$, if $\boldsymbol{x}_i = j$ $\mathbf{1}$, then $D_i = |j - 3|\sqrt{p}$. Thus $D_{(1)} = 0$, $D_{(2)} = D_{(3)} = \sqrt{p}$, and $D_{(4)} = D_{(5)} = 2\sqrt{p}$. Hence $\text{MED}(D_1, ..., D_n) = D_{(5)} = 2\sqrt{p} = \text{MAD}(D_1, ..., D_n)$ since the median distance of the D_i from $D_{(5)}$ is $2\sqrt{p} - 0 = 2\sqrt{p}$. Note that the 5 smallest absolute distances $|D_i - D_{(5)}|$ are $0, 0, \sqrt{p}, \sqrt{p}$, and $2\sqrt{p}$. Hence $W_i = 1$ if $D_i \leq 2\sqrt{p} + 10\sqrt{p} = 12\sqrt{p}$. The clean

data get weight 1 while the outliers get weight 0 since the smallest distance D_i for the [7]
outliers is the Euclidean distance of 3 $\mathbf{1}$ from 16 $\mathbf{1}$ with a $D_i = \|16\ \mathbf{1} - 3\ \mathbf{1}\| = 13\sqrt{p}$.

Hence the `covmb2` estimator $(T, \boldsymbol{C})$ is the sample mean and sample covariance matrix of the clean data. Note that the distance for the outliers to get zero weight is proportional to the square root of the dimension $\sqrt{p}$.

The `covmb2` estimator attempts to give a robust dispersion estimator that reduces the bias by using a big ball about the MED_j that will often contain more than half of the cases, instead of a ball that contains "half" of the cases ($(n+1)/2$ of the cases). The weighting is the default method, but you can also plot the squared Euclidean distances and estimate the number $m \geq n/2$ of cases with the smallest distances to be used. Olive (2017b) uses a collection of R functions *slpack*. The *slpack* function `medout` makes the plot, and the *slpack* function `getB` gives the set B of cases that got weight 1 along with the index `indx` of the case numbers that got weight 1. The function `vecw` stacks the columns of the dispersion matrix $\boldsymbol{C}$ into a vector. Then the elements of the matrix can be plotted.

The function `ddplot5` plots the Euclidean distances from the coordinatewise median versus the Euclidean distances from the `covmb2` location estimator. Typically the plotted points in this DD plot cluster about the identity line, and outliers appear in the upper right corner of the plot with a gap between the bulk of the data and the outliers. An alternative for outlier detection is to replace $\boldsymbol{C}$ by $\boldsymbol{C}_d = diag(\hat{\sigma}_{11}, ..., \hat{\sigma}_{pp})$. For example, use $\hat{\sigma}_{ii} = \boldsymbol{C}_{ii}$. See Olive (2017a, ch. 4), Ro et al. (2015) and Tarr et al. (2016) for references.

CHAPTER 3

EXAMPLES AND SIMULATIONS

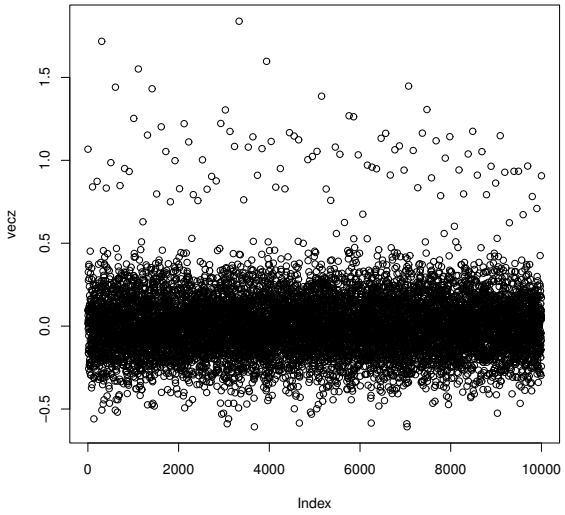

Figure 3.1. Elements of $\boldsymbol{C}$ for outlier data.

Example 4. This example helps illustrate the effect of outliers on classical methods. The artificial data set had $n = 50, p = 100$, and the clean data was iid $N_p(\mathbf{0}, \boldsymbol{I}_p)$. Hence the diagonal elements of the population covariance matrix are 0 and the diagonal elements are 1. Plots of the elements of the sample covariance matrix S and the `covmb2` estimator $\boldsymbol{C}$ are not shown, but were similar to Figure 1. Then the first ten cases were contaminated: $\boldsymbol{x}_i \sim N_p(\boldsymbol{\mu}, 100\boldsymbol{I}_p)$ where $\boldsymbol{\mu} = (10, 0, ..., 0)^T$. Figure 3.1 shows that the `covmb2` dispersion matrix $\boldsymbol{C}$ was not much effected by the outliers. The diagonal elements are near 1 and the off diagonal elements are near 0. Figure 3.2 shows that the sample covariance matrix $\boldsymbol{S}$ was greatly effected by the outliers. Several sample covariances are less than -20 and several sample variances are over 40.

R code to used to produce Figures 3.1 and 3.2 is shown below.

```
#n = 50, p = 100
```

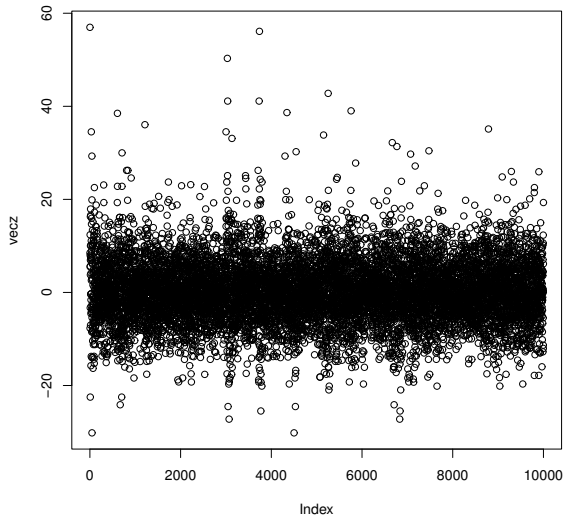

Figure 3.2. Elements of the classical covariance matrix $\boldsymbol{S}$ for outlier data.

```
x<-matrix(rnorm(5000),nrow=50,ncol=100)
out<-medout(x) #no outliers, try ddplot5(x)
out <- covmb2(x,msteps=0)
z<-out$cov
plot(diag(z))  #plot the diagonal elements of C
plot(out$center) #plot the elements of  T
vecz <- vecw(z)$vecz
plot(vecz)

out<-covmb2(x,m=45)
plot(out$center)
plot(diag(out$cov))

#outliers
x[1:10,] <- 10*x[1:10,]
```

```
x[1:10,1] <- x[1:10]+10
medout(x) #The 10 outliers are easily detected in
#the plot of the distances from the MED(X).
ddplot5(x) #two widely separated clusters of data
tem <- getB(x,msteps=0)
tem$indx #all 40 clean cases were used
dim(tem$B) #40 by 100
out<-covmb2(x,msteps=0)
z<-out$cov
plot(diag(z))
plot(out$center)
vecz <- vecw(z)$vecz
plot(vecz) #plot the elements of C
#Figure 1

#examine the sample covariance matrix and mean
plot(diag(var(x)))
plot(apply(x,2,mean)) #plot elements of xbar
zc <- var(x)
vecz <- vecw(zc)$vecz
plot(vecz) #plot the elements of S
#Figure 2

out<-medout(x) #10 outliers
out<-covmb2(x,m=40)
plot(out$center)
plot(diag(out$cov))
```

The `covmb2` estimator can also be used for $n > p$. The *slpack* function `mldsim6` [11] compares 7 estimators: FCH, RFCH, CMVE, RCMVE, RMVN, `covmb2`, and MB described in Olive (2017a, ch. 4). Most of these estimators need a nonsingular dispersion matrix, and work best with $n > 10p$. The function generates data sets and counts how many times the minimum Mahalanobis distance $D_i(T, \boldsymbol{C})$ of the outliers is larger than the maximum distance of the clean data. The simulation suggests that for 40% outliers, the outliers need to be further away from the bulk of the data (`covmb2(k=5)` needs a larger value of pm) than for the other six estimators. As the value pm increases, the distance of the outliers from the clean data increases. The value of $\gamma < 0.5$ gives the proportion of outliers.

For data sets with $p > n$ possible, the function `mldsim7` used the Euclidean distances $D_i(T, \boldsymbol{I}_p)$ and the Mahalanobis distances $D_i(T, \boldsymbol{C}_d)$ where $\boldsymbol{C}_d$ is the diagonal matrix with the same diagonal entries as $\boldsymbol{C}$ where $(T, \boldsymbol{C})$ is the `covmb2` estimator using j concentration type steps. Dispersion matrices are effected more by outliers than good robust location estimators, so when the outlier proportion is high, it is expected that the Euclidean distances $D_i(T, \boldsymbol{I}_p)$ will outperform the Mahalanobis distance $D_i(T, \boldsymbol{C}_d)$. Again the function counts the number of times the minimum outlier distance is larger than the maximum distance of the clean data.

Both functions used several outlier types. The simulations generated 100 data sets. The clean data had $\boldsymbol{x}_i \sim N_p(\boldsymbol{0}, diag(1, ..., p))$. Type 1 had outliers in a tight cluster (near point mass) at the major axis $(0, ..., 0, pm)^T$. Type 2 had outliers in a tight cluster at the minor axis $(pm, 0, ..., 0)^T$. Type 3 had mean shift outliers $\boldsymbol{x}_i \sim N_p((pm, ..., pm)^T, diag(1, ..., p))$. Type 4 changed the pth coordinate of the outliers to pm. Type 5 changed the 1st coordinate of the outliers to pm. (If the outlier $\boldsymbol{x}_i = (x_{1i}, ..., x_{pi})^T$, then $x_{i1} = pm$.)

CHAPTER 4

OUTLIER TYPE 1 EXAMPLES

Table 4.1. Number of Times All Outlier Distances > Clean Distances, otype=1 (runs = 100)

n	p	γ	steps	pm	FCH	RFCH	CMVE	RCMVE	RMVN	covmb2	MB
100	10	0.25	0	20	79	82	79	82	81	68	87
100	10	0.25	0	19	60	63	60	63	62	38	86
100	45	0.25	0	119.5	80	80	80	80	80	100	80
100	45	0.25	0	80	41	41	42	42	42	100	42
100	10	0.25	1	20	76	83	77	84	82	86	85
100	10	0.25	1	19	62	65	62	65	61	56	86
100	45	0.25	1	115	80	80	80	80	80	100	80
100	45	0.25	1	60	0	0	17	17	0	93	17
100	10	0.25	9	20	75	82	75	82	81	88	83
100	10	0.25	9	18	31	35	32	36	34	47	80
100	45	0.25	9	120	80	80	80	80	80	100	80
100	45	0.25	9	60	0	0	17	17	0	91	17
100	10	0.4	0	35	100	100	100	100	100	74	100
100	10	0.4	0	18	84	72	84	72	72	0	85
100	45	0.4	0	110	81	81	81	81	81	100	81
100	45	0.4	0	90	68	68	68	68	68	91	68
100	10	0.4	1	20	92	96	92	96	96	0	92

Table 4.2. Number of Times All Outlier Distances > Clean Distances, otype=1 (runs = 100)

n	p	γ	steps	pm	covmb2	diag
100	10	0.25	0	20	86	66
100	50	0.25	0	65	84	65
100	100	0.25	0	113	84	43
100	500	0.25	0	447	92	6
100	10	0.25	1	19	81	67
100	50	0.25	1	65	90	68
100	100	0.25	1	113	91	44
100	500	0.25	1	454	86	2
100	10	0.25	9	19	82	74
100	50	0.25	9	64	81	58
100	100	0.25	9	113	85	36
100	500	0.25	9	455	83	3
100	10	0.4	0	35	81	79
100	50	0.4	0	92.2	80	79
100	100	0.4	0	150.2	81	79
100	500	0.4	0	550	96	66

CHAPTER 5

OUTLIER TYPE 2 EXAMPLES

Table 5.1. Number of Times All Outlier Distances > Clean Distances, otype=2 (runs = 100)

n	p	γ	steps	pm	FCH	RFCH	CMVE	RCMVE	RMVN	covmb2	MB
100	10	0.25	0	18	100	100	100	100	100	50	100
100	10	0.25	0	15	0	0	35	35	0	1	100
100	45	0.25	0	63.5	0	0	100	100	0	100	100
100	45	0.25	0	35.5	0	0	99	99	0	0	99
100	10	0.25	1	18	99	99	99	99	99	40	100
100	10	0.25	1	15	1	1	36	36	1	3	100
100	45	0.25	1	68	64	64	100	100	64	100	100
100	45	0.25	1	35	0	0	93	93	0	0	93
100	10	0.25	9	18	98	98	99	99	98	45	100
100	10	0.25	9	9	0	0	11	12	0	0	99
100	45	0.25	9	65	1	1	100	100	1	100	100
100	45	0.25	9	35	0	0	95	95	0	0	95
100	10	0.4	0	25	100	100	100	100	100	16	100
100	10	0.4	0	10	0	0	2	2	0	0	85
100	45	0.4	0	50	100	100	100	100	100	0	100
100	45	0.4	0	37	0	0	80	80	0	0	80
100	10	0.4	1	13	82	82	82	82	82	0	100

Table 5.2. Number of Times All Outlier Distances > Clean Distances, otype=2 (runs = 100)

n	p	γ	steps	pm	covmb2	diag
100	10	0.25	0	18	90	52
100	50	0.25	0	62	82	79
100	100	0.25	0	107.5	81	79
100	500	0.25	0	435.5	82	78
100	10	0.25	1	18	84	51
100	50	0.25	1	63	88	79
100	100	0.25	1	101.5	81	79
100	500	0.25	1	452.5	83	78
100	10	0.25	9	18	88	46
100	50	0.25	9	63	82	79
100	100	0.25	9	110.4	83	78
100	500	0.25	9	452.5	82	79
100	10	0.4	0	29	91	76
100	50	0.4	0	84.3	81	79
100	100	0.4	0	140.1	81	77
100	500	0.4	0	506.5	81	78

CHAPTER 6

OUTLIER TYPE 3 EXAMPLES

Table 6.1. Number of Times All Outlier Distances > Clean Distances, otype=3 (runs = 100)

n	p	γ	steps	pm	FCH	RFCH	CMVE	RCMVE	RMVN	covmb2	MB
100	10	0.25	0	7	100	100	93	93	100	21	100
100	10	0.25	0	4.5	74	84	60	69	84	0	73
100	45	0.25	0	32	86	86	68	68	86	100	100
100	45	0.25	0	11	28	28	44	44	28	96	77
100	10	0.25	1	5.5	97	100	85	88	99	7	97
100	10	0.25	1	5	90	97	80	86	96	1	92
100	45	0.25	1	32	84	84	75	75	84	100	100
100	45	0.25	1	9	16	16	33	33	16	91	53
100	10	0.25	9	6	99	100	87	88	100	37	99
100	10	0.25	9	4.3	66	84	54	67	79	0	67
100	45	0.25	9	33	90	90	78	78	90	100	100
100	45	0.25	9	9	6	6	29	29	6	95	51
100	10	0.4	0	14	100	100	91	91	100	67	100
100	10	0.4	0	5.5	66	67	13	13	67	0	83
100	45	0.4	0	32	84	84	79	79	84	100	100
100	45	0.4	0	15	16	16	26	26	16	0	97
100	10	0.4	1	13	100	100	75	75	100	100	100

Table 6.2. Number of Times All Outlier Distances > Clean Distances, otype=3 (runs = 100)

n	p	γ	steps	pm	covmb2	diag
100	10	0.25	0	6	55	82
100	50	0.25	0	8	63	98
100	100	0.25	0	8	5	86
100	500	0.25	0	10	0	87
100	10	0.25	1	5.5	50	86
100	50	0.25	1	7	50	95
100	100	0.25	1	8	49	96
100	500	0.25	1	10	0	90
100	10	0.25	9	5.5	60	81
100	50	0.25	9	7	49	96
100	100	0.25	9	8	61	94
100	500	0.25	9	10	1	92
100	10	0.4	0	12.7	79	80
100	50	0.4	0	17	74	89
100	100	0.4	0	18	54	86
100	500	0.4	0	20	5	80

CHAPTER 7

OUTLIER TYPE 4 EXAMPLES

Table 7.1. Number of Times All Outlier Distances > Clean Distances, otype=4 (runs = 100)

n	p	γ	steps	pm	FCH	RFCH	CMVE	RCMVE	RMVN	covmb2	MB
100	10	0.25	0	25	99	99	71	71	99	100	100
100	10	0.25	0	20	70	87	44	53	88	78	74
100	45	0.25	0	185	81	81	79	79	81	100	92
100	45	0.25	0	80	3	3	9	9	3	100	14
100	10	0.25	1	21	80	94	58	68	92	95	83
100	10	0.25	1	19	67	80	38	41	77	69	73
100	45	0.25	1	180	81	81	76	76	81	100	93
100	45	0.25	1	54	0	0	0	0	0	80	0
100	10	0.25	9	21	85	92	61	63	93	98	91
100	10	0.25	9	19	63	75	44	53	75	80	69
100	45	0.25	9	180	80	80	78	78	80	100	93
100	45	0.25	9	55	0	0	2	2	0	88	2
100	10	0.4	0	35	98	98	64	64	98	82	100
100	10	0.4	0	20	9	9	0	0	9	0	86
100	45	0.4	0	141	79	79	81	81	81	100	84
100	45	0.4	0	85	0	0	6	6	0	91	24
100	10	0.4	1	35	97	97	63	63	97	100	100

Table 7.2. Number of Times All Outlier Distances > Clean Distances, otype=4 (runs = 100)

n	p	γ	steps	pm	covmb2	diag
100	10	0.25	0	19	89	54
100	50	0.25	0	53	93	24
100	100	0.25	0	93	82	11
100	500	0.25	0	260	89	10
100	10	0.25	1	19	97	73
100	50	0.25	1	52	87	16
100	100	0.25	1	84	88	11
100	500	0.25	1	270	86	8
100	10	0.25	9	18	86	52
100	50	0.25	9	52	88	16
100	100	0.25	9	84	84	9
100	500	0.25	9	270	80	3
100	10	0.4	0	35	85	76
100	50	0.4	0	85	84	64
100	100	0.4	0	130	82	52
100	500	0.4	0	380	85	35

CHAPTER 8

OUTLIER TYPE 5 EXAMPLES

Table 8.1. Number of Times All Outlier Distances > Clean Distances, otype=5 (runs = 100)

n	p	γ	steps	pm	FCH	RFCH	CMVE	RCMVE	RMVN	covmb2	MB
100	10	0.25	0	15	100	100	99	99	100	4	100
100	10	0.25	0	7	77	93	62	73	93	0	78
100	45	0.25	0	150	96	96	76	76	96	100	100
100	45	0.25	0	27	44	44	43	43	44	0	90
100	10	0.25	1	15	100	100	99	99	100	3	100
100	10	0.25	1	7	68	90	38	52	91	0	70
100	45	0.25	1	77	80	80	54	54	80	100	100
100	45	0.25	1	26	27	27	44	44	27	0	81
100	10	0.25	9	9	100	100	84	84	100	0	100
100	10	0.25	9	7	76	95	61	73	100	100	100
100	45	0.25	9	85	85	85	61	61	85	100	100
100	45	0.25	9	30	38	38	45	45	38	0	92
100	10	0.4	0	25	100	100	86	86	100	19	100
100	10	0.4	0	9	71	71	14	14	71	0	85
100	45	0.4	0	90	81	81	61	61	81	100	100
100	45	0.4	0	30	22	22	16	16	22	0	90
100	10	0.4	1	25	100	100	85	85	100	20	100

Table 8.2. Number of Times All Outlier Distances > Clean Distances, otype=5 (runs = 100)

n	p	γ	steps	pm	covmb2	diag
100	10	0.25	0	17	92	32
100	50	0.25	0	49	89	25
100	100	0.25	0	80	88	29
100	500	0.25	0	251	89	4
100	10	0.25	1	17	91	29
100	50	0.25	1	50	88	21
100	100	0.25	1	82	91	15
100	500	0.25	1	269	85	4
100	10	0.25	9	17	92	45
100	50	0.25	9	50	87	17
100	100	0.25	9	85	91	13
100	500	0.25	9	270	80	2
100	10	0.4	0	27	92	50
100	50	0.4	0	75	90	51
100	100	0.4	0	119	89	52
100	500	0.4	0	360	89	37

CHAPTER 9

CONCLUSION

For my simulations, the number of generated outlier datasets was one hundred. For the mldsim6 function, we want to have two kinds of output: first, we want just one or few counts over 80. To obtain this output, want to make the value of pm smaller. Second, we want just one or few results under 80. To obtain this output, we want to make the value of pm larger. Based on the results, as the the value of p is changing, the value of pm is also changing. The value of gamma does affect the output. For the mldsim7 function, we want to have one or few counts greater than 80. Based on the output, as the value of p is increasing, the value of pm is also increasing. The value of gamma does not effect the whole outputs. For these two functions, they have same specific pattern that we can track. We can detect the outliers by using these two functions.

The simulations were done in *R*. See R Core Team (2016). The collection of *R* functions *slpack*, available from (http://lagrange.math.siu.edu/Olive/slpack.txt), has some useful functions. The functions `mldsim6` and `mldsim7` were used to do the simulation.

[1] Olive, D.J. (2017a), *Robust Multivariate Analysis*, Springer, New York, NY, to appear.

[2] Olive, D.J. (2017b), *Prediction and Statistical Learning*, online course notes, see (http://lagrange.math.siu.edu/Olive/slearnbk.htm).

[3] R Core Team (2016), "R: a Language and Environment for Statistical Computing," R Foundation for Statistical Computing, Vienna, Austria, (www.R-project.org).

[4] Ro, K., Zou, C., Wang, W., and Yin, G. (2015), "Outlier Detection for High–Dimensional Data," *Biometrika*, 102, 589-599.

[5] Tarr, G., Müller, S., and Weber, N.C. (2016), "Robust Estimation of Precision Matrices Under Cellwise Contamination," *Computational Statistics & Data Analysis,* 93, 404-420.

Graduate School
Southern Illinois University

Handong Wang

wanghandong1992@gmail.com

Southern Illinois University Carbondale
Bachelor of Arts, Mathematics, May 2015

Research Paper Title:
Outlier Detection for High Dimensional Data

Major Professor: Dr. David J. Olive