Outlier Detection in High Dimensional Data using PCA

Subhrajyoty Roy (MB1911)

September 20, 2019

Introduction

Principal Component Analysis is a very useful dimension reduction technique which can effectively find out the dispersion of the dataset from its center cluster in different direction in the order of magnitude of the dispersion, so that, the most significant part of the dispersion is comprised in a very few principal components. In this assignment, we shall discuss the use Principal components to detect outliers in high dimensional data, where there is not simple way to visualize the data and detect the outliers by mere inspection. Applying the PCA would allow us to capture the variablity in the data in two or three dimensions, thereby allowing us to have a visualization of the data.

Creation of Contaminated Dataset

We first consider a 5-variate normal distribution whose mean vector and covariance matrix is generated as follows. For the mean vector, each component is chosen from a uniform $(-1, 1)$ distribution, while the covariance matrix is generated according to a Wishart distribution. We also generate mean vector for the contaminated distribution where each component is chosen from a uniform $(-5, 5)$ distribution.

```
set.seed(1911)  # set my roll number as seed for reproducibilty
mu1 <- runif(5, min = -1, max= 1)  # mean for regular distribution
mu2 <- runif(5, min = -5, max = 5)  # mean for contaminated distribution

mat <- diag(5)  + matrix(1, nrow = 5, ncol = 5)
```

```r
x <- rWishart(2, 5, mat)  # generate 5x5 covariance matrix
Sigma1 <- x[,,1]  # sigma for regular distribution
Sigma2 <- x[,,2]  # sigma for contamination distribution
```

Therefore, we have the following,

- The regular distribution is a 5-variate normal distribution with mean vector

$$\begin{bmatrix} ,0.31 \\ -0.03 \\ -0.57 \\ -0.22 \\ -0.46, \end{bmatrix}$$

  and covariance matrix

$$\begin{bmatrix} ,2.95 & 6.25 & 3.24 & 2.99 & 3.11 \\ ,6.25 & 21.49 & 8.29 & 9.34 & 0.67 \\ ,3.24 & 8.29 & 6.95 & 6.22 & 1.67 \\ ,2.99 & 9.34 & 6.22 & 7.52 & 0.19 \\ ,3.11 & 0.67 & 1.67 & 0.19 & 8.59 \\ , & & & & \end{bmatrix}$$

- The contaminated distribution is a 5-variate normal distribution with mean vector

$$\begin{bmatrix} ,2.64 \\ -4.44 \\ 1.5 \\ 4.85 \\ -4.58, \end{bmatrix}$$

  and covariance matrix

$$\begin{bmatrix} ,7.89 & 3.48 & 5.08 & 2.73 & 5.68 \\ ,3.48 & 8.06 & 4.89 & 4.15 & 5.36 \\ ,5.08 & 4.89 & 6.37 & 2.35 & 5.54 \\ ,2.73 & 4.15 & 2.35 & 5 & 2.73 \\ ,5.68 & 5.36 & 5.54 & 2.73 & 5.89 \\ , & & & & \end{bmatrix}$$

Now, we generate 100 observations from each of distributions, the regular and the contaminated one. Then, we generate 100 bernoulli random variables with success probability 0.05. The observed dataset is taken as;

$$X_i = B_i C_i + (1 - B_i) R_i \qquad i = 1, 2, \ldots 100$$

where C_i is the i-th sample from Contamined distribution, B_i is the i-th bernoulli trial result, and R_i is the i-th sample from Regular distribution.

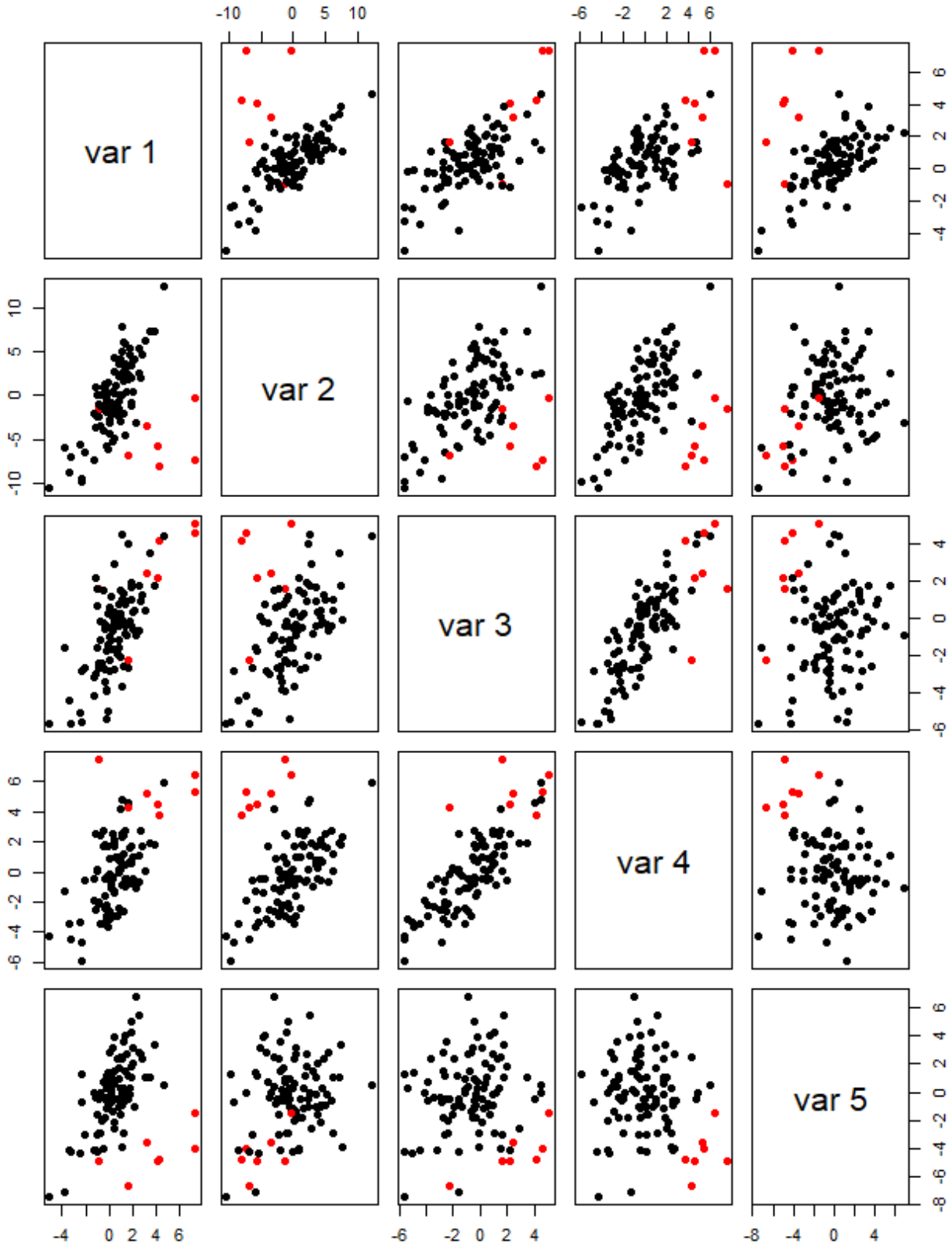
```
library(MASS)
set.seed(1911)  # my roll number
regular_samples <- mvrnorm(n = 100, mu = mu1, Sigma = Sigma1)
contaminated_samples <- mvrnorm(n = 100, mu = mu2, Sigma = Sigma2)

bernoulli_trials <- rbinom(n = 100, size = 1, prob = 0.05)

final_samples <- ((1 - bernoulli_trials)*regular_samples) +
    (bernoulli_trials * contaminated_samples)

print(paste0("There are ", sum(bernoulli_trials), " many outliers"))
```

```
[1] "There are 7 many outliers"
```

Visualization of the Data

Since the data is 5-variate, hence we cannot visualize it in 5-dimensional space. However, we can look at pairwise set of predictors and plot the datapoints on the basis of those variables only. We also color the outliers to see whether the plots does allow one to detect the outliers.

```
pairs(final_samples, col = ifelse(bernoulli_trials,
                                  "red", "black"), pch = 19)
```



Note that, no two variables can detect the outliers with confidence. Hence, we need to specifically look at the directions at which there is most variability in the data, hence effectively look at the few principle components.

Computation of Principal Component

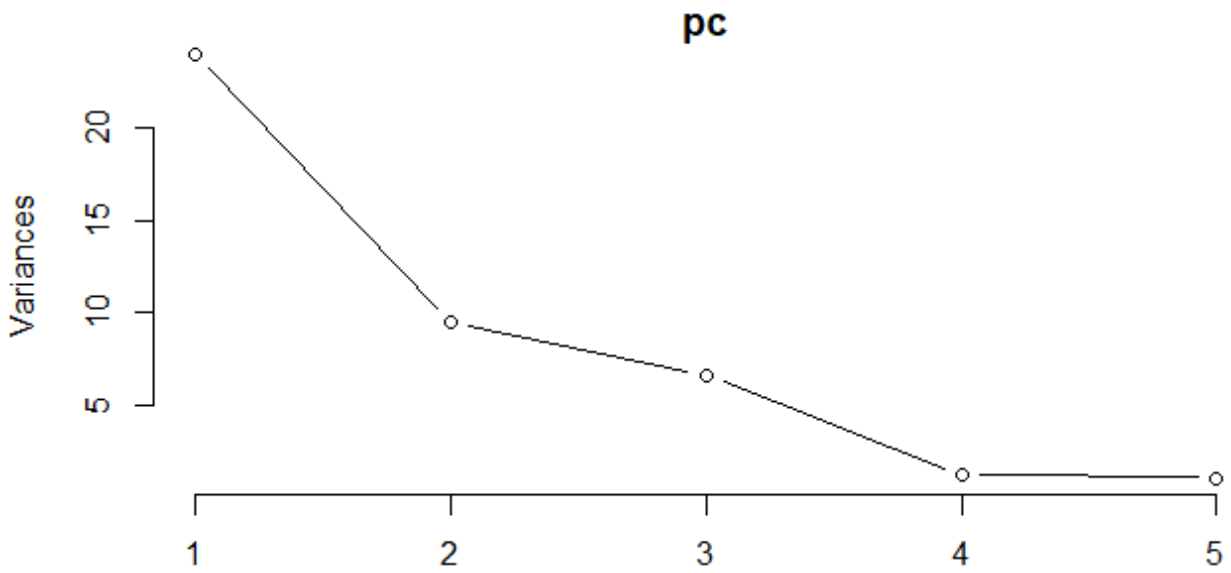
We compute the principal components of the above contaminated dataset.

```
pc <- prcomp(final_samples)  # computes the principal components
pc
```

```
Standard deviations (1, .., p=5):
[1] 4.895583 3.071227 2.563416 1.070416 1.009443

Rotation (n x k) = (5 x 5):
           PC1         PC2        PC3         PC4         PC5
[1,] 0.2766466 -0.08996917  0.3888470  0.82683435 -0.28375205
[2,] 0.8079288  0.32795332 -0.4895327 -0.00595139 -0.00447248
[3,] 0.3523608 -0.37463454  0.3422770 -0.52157662 -0.58846843
[4,] 0.3546252 -0.56854213  0.1982074 -0.02843372  0.71477703
[5,] 0.1442242  0.64866091  0.6728418 -0.20846795  0.24952650
```

We only consider the first two principal components, as these are the only ones will be used to find out the outliers. Before we compute the linear combination of the variables denoted by principal components, we make the screeplot in order to find out how much of the total variation is explained by the principal components.

```
par(mar = c(2, 4, 1, 2))
screeplot(pc, type = "lines")
```

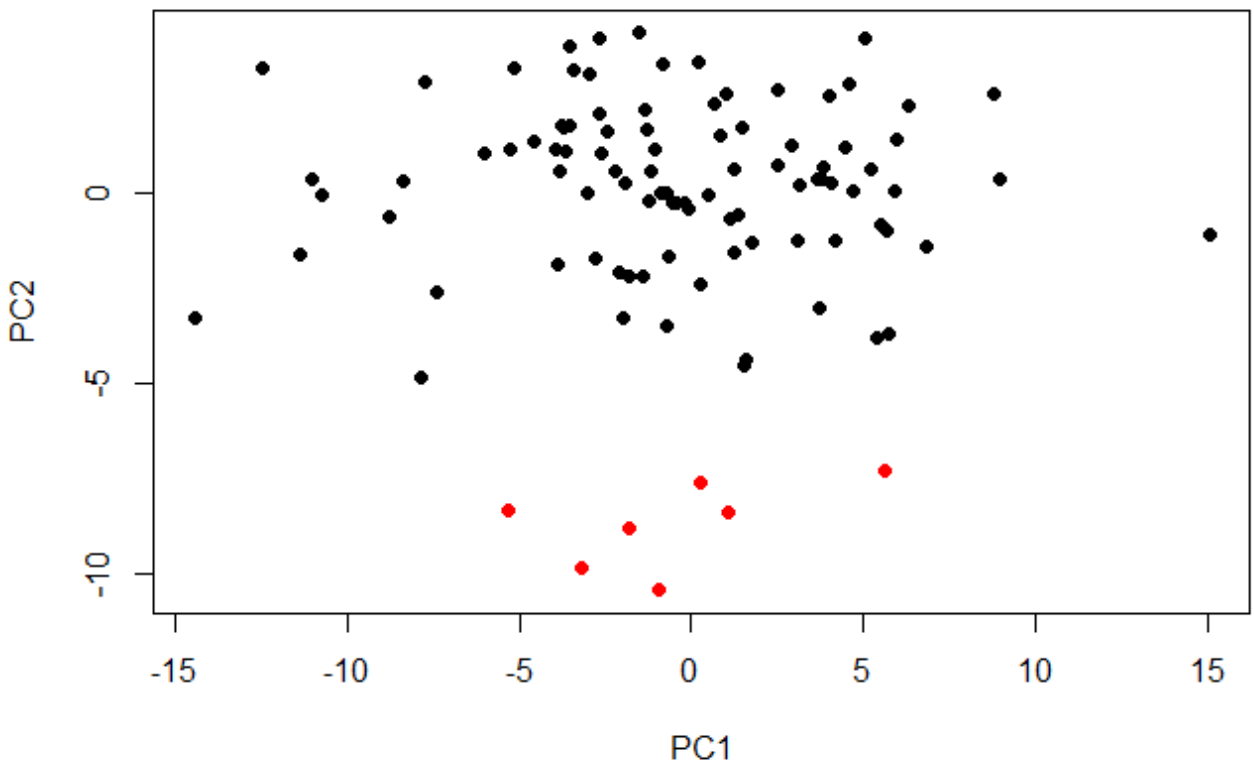


Now, to compute the principal components, we multiply the data matrix with the rotation matrix obtained from PCA.

```
# computes the PC1 and PC2
pcdata <- final_samples %*% pc$rotation[,c(1,2)]
```

Now, we make a plot of those principal components to see whether we can actually detect the outlier out of this.

```
plot(pcdata, col = ifelse(bernoulli_trials, "red", "black"), pch = 19)
```



clearly, as we see from the above plot, the outliers are much more prominent in principal component space.

Outlier Detection

Let, C be a robust measure of dispersion matrix of 1st and 2nd principal component, while T be a robust measure of center of 1st and 2nd principal

components. Also, let Y be the vector of 1st and 2nd principal components. Then, the Mahalanobis type Distance squared;

$$F = (Y - T)^\top C^{-1} (Y - T)$$

should follow a chi-sqaure distribution with 2 degrees of freedom (since C is 2×2 matrix). We consider any point as outlier if its mahalanobis distance exceeds the 0.99-th quantile of central χ^2_2 distribution.

```
x <- cov.rob(pcdata)
dists <- mahalanobis(pcdata, center = x$center, cov = x$cov)

detected_outliers <- (dists > qchisq(p = 0.99, df = 2))
print(which(detected_outliers))
```

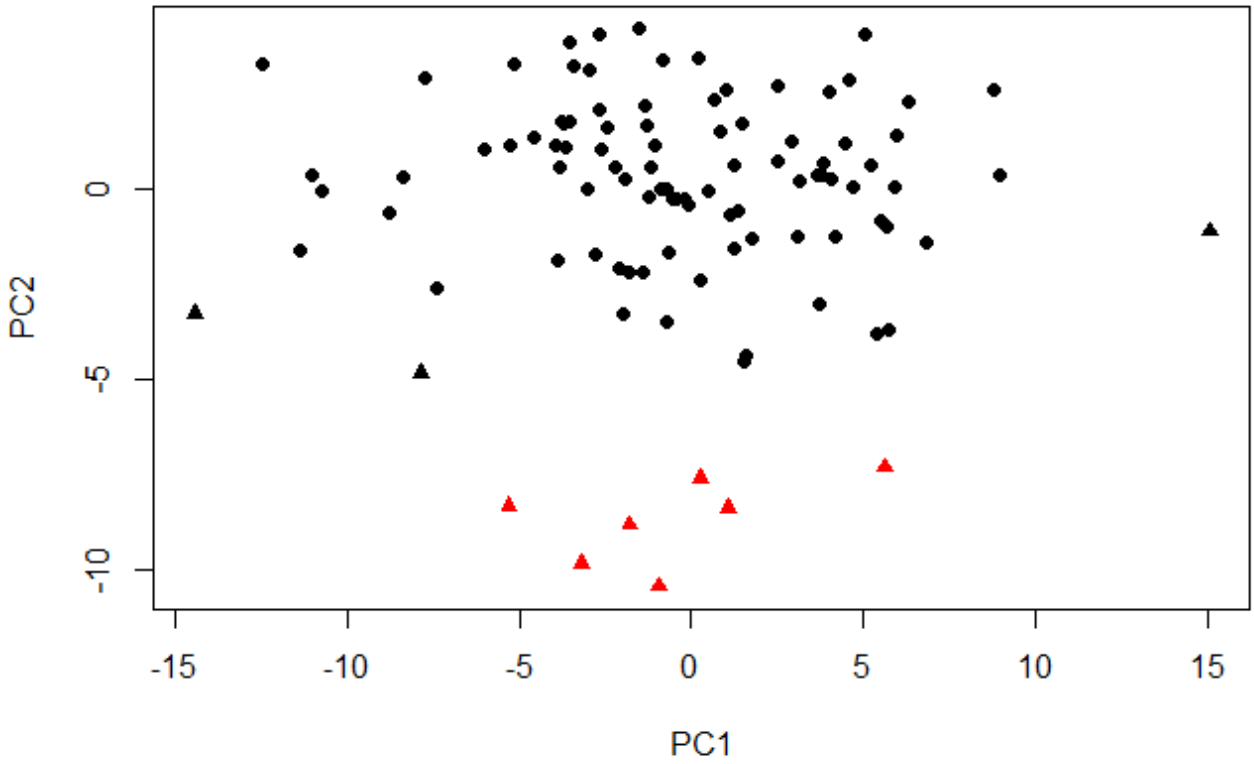
```
 [1]  5  9 12 13 22 68 69 82 89 90
```

```
true_outliers <- which(bernoulli_trials == 1)
print(true_outliers)
```

```
[1]  5 12 68 69 82 89 90
```

We find that there are some difference between detected outliers and the true outliers. If we look back to the plot, we get the following:

```
plot(pcdata, col = ifelse(bernoulli_trials, "red", "black"),
     pch = ifelse(detected_outliers, 17, 19))
```



In the above plot, the detected outliers are drawn using triangles, while the true outliers are coloured in red. Note that, the above procedure using principal component space to detect outliers correctly detected all the outliers. However, it also detected 3 more datapoints as outliers which are not truly obtained from the contaminated distribution.

THANK YOU