Business Analytics - ETC3250 2018 - Lab 9 Solutions

Principal Component Analysis

Souhaib Ben Taieb

26 April 2018

Question 1

Supppose that the columns of $X \in \mathbb{R}^{n \times p}$ have sample mean zero. Prove that $Xv \in \mathbb{R}^n$ has sample mean zero for any vector $v \in \mathbb{R}^p$.

Let $z = Xv$. The sample mean of z is given by

$$\frac{1}{n}\sum_{i=1}^{n} z_i = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p} X_{ij}v_j \tag{1}$$

$$= \frac{1}{n}\sum_{j=1}^{p} v_j \sum_{i=1}^{n} X_{ij} \tag{2}$$

$$= 0 \tag{3}$$

Question 2

Suppose that the columns of $X \in \mathbb{R}^{n \times p}$ have been centered (i.e., they have sample mean zero). The total sample variance of X is defined as $\frac{1}{n}\text{Trace}(X^{'}X)$.

Let $X = UDV^{'}$ be the singular value decomposition of X where the columns of $U \in \mathbb{R}^{n \times p}$ and $V \in \mathbb{R}^{p \times p}$ are orthonormal and the matrix $D \in \mathbb{R}^{p \times p}$ is diagonal with positive real entries, $D = \text{diag}(d_1, ...d_p)$, where $d_1 \geq d_2 \geq \cdots \geq d_p \geq 0$.

Prove that the total sample variance of X is given by $\frac{1}{n}\sum_{j=1}^{p} d_j^2$.

(Hint: note that $\text{Trace}(AB) = \text{Trace}(BA)$)

$$\frac{1}{n}\text{Trace}(X^{'}X) \tag{4}$$

$$= \frac{1}{n}\text{Trace}(VD^2V^{'}) \tag{5}$$

$$= \frac{1}{n}\text{Trace}(D^2V^{'}V) \tag{6}$$

$$= \frac{1}{n}\text{Trace}(D^2) \tag{7}$$

$$= \frac{1}{n}\sum_{j=1}^{p} d_j^2 \tag{8}$$

Question 3

If again $X = UDV^{'}$, with centered columns, and $V_k \in \mathbb{R}^{p \times k}$ denotes the first k columns of V, prove that $XV_kV_k^{'}$ has total sample variance $\frac{1}{n}\sum_{j=1}^{k} d_j^2$.

(Hint: Use the fact that $V_k^{'}V = \begin{bmatrix} I_k & 0 \end{bmatrix}$ where I_k is a $k \times k$ identity matrix.)

$$\frac{1}{n}\text{Trace}(V_k V_k^{'} X^{'} X V_k V_k^{'}) \tag{9}$$
$$= \frac{1}{n}\text{Trace}(V_k V_k^{'} V D^2 V^{'} V_k V_k^{'}) \tag{10}$$
$$= \frac{1}{n}\text{Trace}(V_k \begin{bmatrix} I_k & 0 \end{bmatrix} D^2 \begin{bmatrix} I_k \\ 0 \end{bmatrix} V_k^{'}) \tag{11}$$
$$= \frac{1}{n}\text{Trace}(\begin{bmatrix} V_k & 0 \end{bmatrix} D^2 \begin{bmatrix} V_k^{'} \\ 0 \end{bmatrix}) \tag{12}$$
$$= \frac{1}{n}\text{Trace}(\begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix} D^2) \tag{13}$$
$$= \frac{1}{n}\sum_{j=1}^{k} d_j^2 \tag{14}$$

Question 4

Download the file *digits.Rdata* from https://github.com/bsouhaib/BA2018/blob/master/data/digits.rdata which contains a matrix *threes* that has dimension 658×256. Each row of the matrix corresponds to an image of a *3* that was written by a different person. Hence each row vector is of length 256, corresponding to a 16×16 pixels image that has been unraveled into a vector, and each pixel takes grayscale values between -1 and 1.

You can plot any of the images, i.e., any row of the matrix using the following code

```
plot.digit <- function(x,zlim=c(-1,1)) {
  cols = gray.colors(100)[100:1]
  image(matrix(x,nrow=16)[,16:1],col=cols,
        zlim=zlim,axes=FALSE)
}

# Example
# plot.digit(threes[1,])
```

1. Compute the principal component directions and principal component scores of the data using (i) the *prcomp* package, (ii) eigenvalue decomposition of the covariance matrix, and (ii) SVD of the matrix.

```
load("../../data/digits.Rdata")

# Compute PCs using built-in function
z <- prcomp(threes)

#Compute via covariance:
X <- scale(threes)
C <- t(X) %*% X
z1 <- eigen(C)
pc.cv <- X %*% z1$vectors

# Compute via svd
z2 <- svd(X)
pc.svd <- X %*% z2$v
```

2. Plot the first two principal component scores (the x-axis being the first score and the y-axis being the second score). Note that each point in this plot corresponds to an image of a *3*.

3. For each of the first two principal component scores, compute the following percentiles: 5%, 25%, 50%, 75%, 95%. Draw these values as vertical and horizontal lines on top of your plot (i.e., vertical for the percentiles of the first principal component score, and horizontal for those of the second.)

(Hint: use *quantile* for the percentiles, and *abline* to draw the lines.)

4. Plot the images that are closest to each of the vertices of the grid on your plot, i.e. 25 images in total. To do so, identify a point (i.e., an image of a *3*) close to each of the vertices of the grid on your plot. This can be done by using the *identify* function with $n = 25$, which allows you to click on the plot 25 times (since there are 25 vertices). Each time you click, it will print the index of the point that is closest to your click's location. Make sure you click left-to-right, and top-to-bottom, and record the indices in that order.

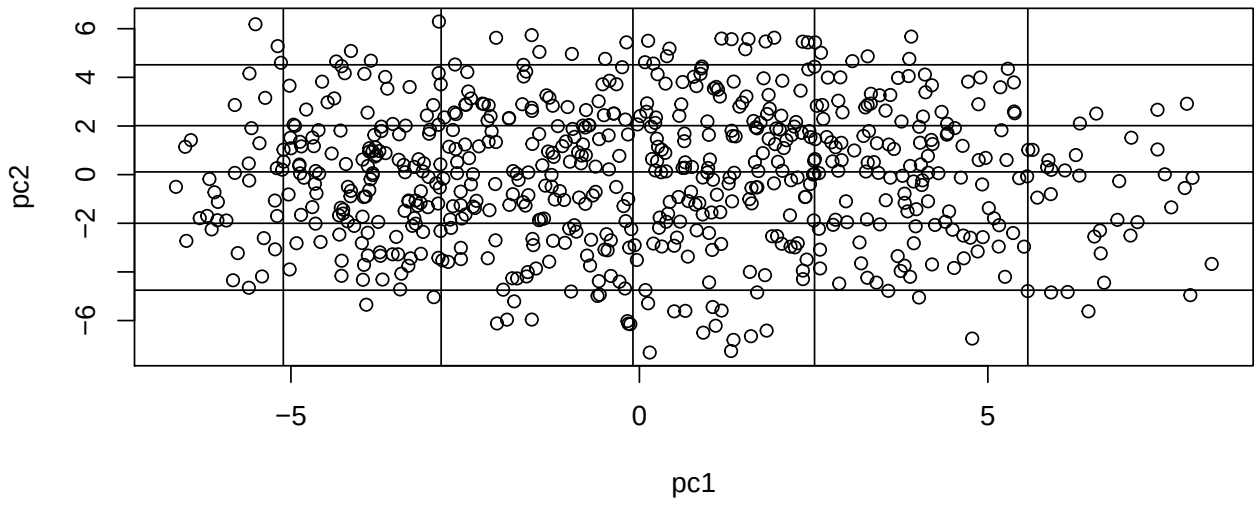
(Note: The identify function returns a vector of indices in sorted order. This isn't what you want—you want them in the order that you clicked, so you may have to build this vector manually using the output from the identify function)

For example, if you saved the vector of indices that you built previously as *inds*, and you built them by clicking left-to-right and top-to-bottom as instructed, you can plot the images with:

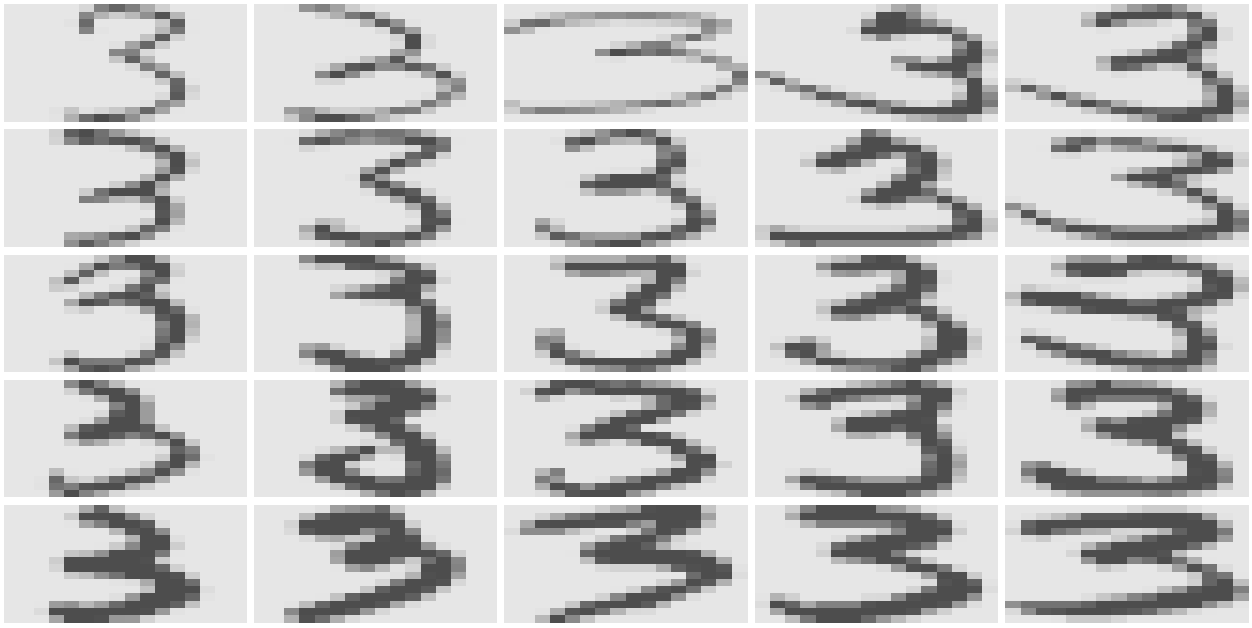
```
par(mfrow=c(5,5)) # allow for 5 x 5 plots
par(mar=c(0.2,0.2,0.2,0.2)) # set small margins
for (i in inds) {
  plot.digit(threes[i,])
}
```

5. Looking at these digits, what can be said about the nature of the first two principal component scores? (The first principal component score is increasing as you move from left-to-right in any of the rows. The second principal component score is decreasing as you move from top-to-bottom in any of the columns.) In other words, I'm asking you to explain what changes with respect to changes in each of the component scores.

```
load("../../data/digits.Rdata")

res <- prcomp(threes)
pc1 <- res$x[, 1]
pc2 <- res$x[, 2]

plot(pc1, pc2)
abline(v = quantile(pc1, probs = c(0.05, 0.25, 0.5, 0.75, 0.95)))
abline(h = quantile(pc2, probs = c(0.05, 0.25, 0.5, 0.75, 0.95)))
```

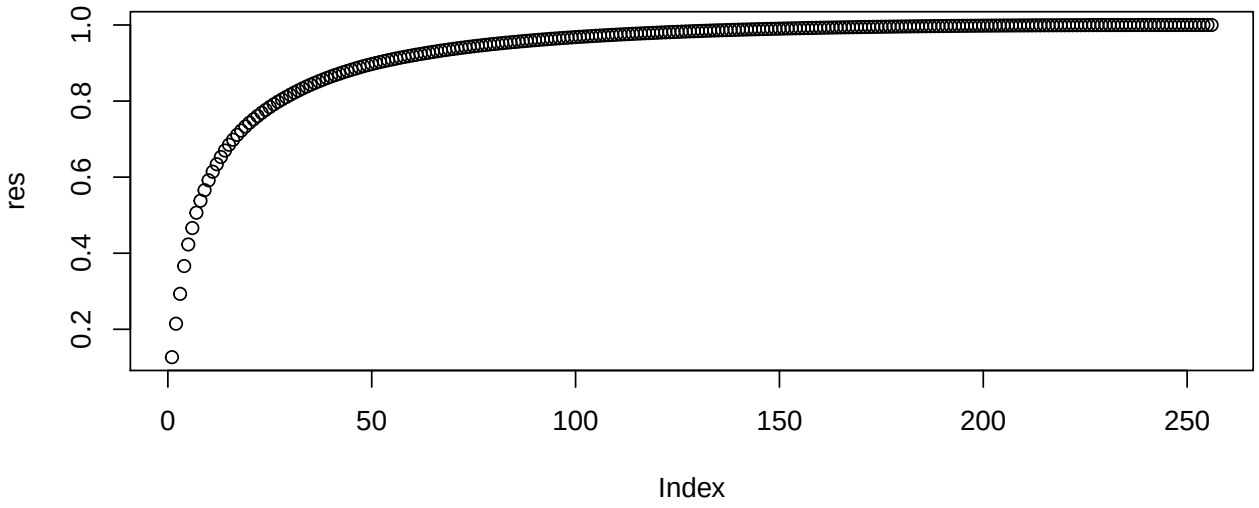


```
#inds <- identify(pc1, pc2, n = 25)
#plot(pc1[inds], pc2[inds], type = 'n')
#text(pc1[inds], pc2[inds], inds, col = "red")

newinds <- c(73, 345, 550, 317, 640,
  284, 84, 519, 51, 396,
  392, 241, 645, 610, 182,
  247, 312, 142, 405, 260,
  194, 149, 431, 633, 298)

par(mfrow=c(5,5)) # allow for 5 x 5 plots
par(mar=c(0.2,0.2,0.2,0.2)) # set small margins
for (i in newinds) {
  plot.digit(threes[i,])
}
```



6. Plot the proportion of variance explained by the first k principal component directions, as a function of $k = 1, \ldots, 256$. How many principal component directions would we need to explain 50% of the variance? How many to explain 90% of the variance?

```
res <- cumsum(res$sdev^2/sum(res$sdev^2))
plot(res)
```




```
print(which(res > 0.5)[1])
# [1] 7
print(which(res > 0.9)[1])
# [1] 52
```