An Introduction to R
R Workshop 4: Principal Component Analysis

Contents

1 Introduction

In this applied homework assignment you will learn how to use principal component analysis with real data. You can find the homework questions at the end of this document.

2 Workshop Procedures

The R code for principal component analysis that will be used in a case study of mtcars is presented below.

This tutorial primarily leverages the mtcars data set that is built into R. The homework assignment uses the US Arrests data. This is a dataset that contains four variables that represent the number of arrests per 100,000 residents for Assault, Murder, and Rape in each of the fifty US states in 1973. The data set also contains the percentage of the population living in urban areas, UrbanPop. In addition to loading the set, we'll also use a few packages that provide added functionality in graphical displays and data manipulation. We use the head command to examine the first few rows of the data set to ensure proper upload.

3 Principal Component Analysis

We will start with a more visual example of PCA using the mtcars dataset. We will also think through the math with mtcars. We will do the homework questions using US Arrests. First, load in the dataset and recall that mtcars is in the mlbench package.

```
library(mlbench)
data("mtcars")
```

You will need certain packages to run a PCA algorithm. If this is the first time using the package, install it; then run the library to bring it into your R session.

```
install.packages("devtools")
install_github("vqv/ggbiplot")
library(devtools)
library(ggbiplot)
```

It is possible to run the PCA algorithm directly. Store it as an object and explore the five pieces of information stored inside: (1) sdev, (2) rotation, (3) center, (4) scale, and (5) x.

- The sdev is used to calculate the proportion of variance explained by each variable.
- The rotation gives the loading vectors that are used to rotate the original data to obtain the principal components.
- The center gives the mean of each of the original variables.

- The scale gives the standard deviation of the original variables.
- x gives the scaled data which happens before the actual PCA analysis is done. Scaling is done behind the scenes in prcomb.

```
mtcars.pca <- prcomp(mtcars[,c(1:7,10,11)], center = TRUE,scale. = TRUE)
summary(mtcars.pca)
str(mtcars.pca)
```

3.1 Proportion of Variance Explained (PVE)

We want to think about how much of the variance is explained by each variable. That is to say, we want to know the proportion of variance explained by each component in the dataset. We can do that in the following way:

```
mtcars.pca$sdev
mtcars.pca$sdev ^ 2 / sum(mtcars.pca$sdev ^ 2)
```

We can write a function to do this:

```
get_PVE = function(pca_out) {
  pca_out$sdev ^ 2 / sum(pca_out$sdev ^ 2)
}
pve = get_PVE(mtcars.pca)
pve
```

Now, plot the proportion of variance explained with each component on the x-axis and the proportion on the y-axis.

```
plot(
  pve,
  xlab = "Principal Component",
  ylab = "Proportion of Variance Explained",
  ylim = c(0, 1),
  type = 'b'
)
```

Another way to visualize this is the cumulative sum of explained variance (this should add up to 1 - think about why that is).

```
cumsum(pve)

plot(
  cumsum(pve),
  xlab = "Principal Component",
  ylab = "Cumulative Proportion of Variance Explained",
  ylim = c(0, 1),
  type = 'b'
)
```

The code above obscures the use of eigenvalues and eigenvectors in the underlying calculations involved. We can be more explicit about this up fron to see what is really happening. In order to do this, we should first center and scale the data.

```
# create new data frame with centered variables
scaled_df <- apply(mtcars, 2, scale)
head(scaled_df)
```

We then want to calculate the the eivenvalues and eigenvectors. These are obtained by using the covariance matrix of the scaled data.

```
mtcars.cov <- cov(scaled_df)
mtcars.eigen <- eigen(mtcars.cov)
str(mtcars.eigen)
```

Check that the eigenvectors are orthogonal to each other. You should get a number very close to zero like theory tells us we should expect.

```
install.packages("pracma")
library(pracma)
vectors <- mtcars.eigen$vectors
vectors <- data.frame(vectors)
dot(vectors$X1, vectors$X2)
```

The PVE for each variable can be calculated as the eigenvalue of the respective variable divided by the sum of the eigenvalues.

```
PVE <- mtcars.eigen$values / sum(mtcars.eigen$values)
round(PVE, 2)
```

We can not plot the PVE by component and plot the cumulative sum of the PVE. In this instance, let's combine the graphs into a single plot.

```
# PVE (aka scree) plot
PVEplot <- qplot(c(1:11), PVE) +
  geom_line() +
  xlab("Principal Component") +
  ylab("PVE") +
  ggtitle("Scree Plot") +
  ylim(0, 1)

# Cumulative PVE plot
cumPVE <- qplot(c(1:11), cumsum(PVE)) +
  geom_line() +
  xlab("Principal Component") +
  ylab(NULL) +
  ggtitle("Cumulative Scree Plot") +
```

```
  ylim(0,1)

grid.arrange(PVEplot, cumPVE, ncol = 2)
```

Having seen two approaches to determine the PVE and thus understand which components are most relevant, we can now plot the center and rotation using the vectors in the PCA analysis. With ggbiplot, the default is to take the two most important principal components. However, it is possible to choose others, an example of which is included in the second line of code (the second and third most important principal components).

```
ggbiplot(mtcars.pca)
ggbiplot(mtcars.pca, choices = 2:3)
ggbiplot(mtcars.pca, labels=rownames(mtcars))
```

We can add elements of clustering to the principal component analysis graphs. For instance, in the cars dataset we know the country to which different car models belong. It is possible to cluster our principal component vectors by country to see if there are patterns in the data.

```
mtcars.country <- c(rep("Japan", 3), rep("US",4), rep("Europe",
 7),rep("US",3), "Europe", rep("Japan", 3), rep("US",4), rep("Europe",
  3), "US", rep("Europe", 3))

ggbiplot(mtcars.pca,ellipse=TRUE,  labels=rownames(mtcars),
 groups=mtcars.country)
```

We can cluster by principal components beyond the first two. An example is below which uses the third and fourth most important PC.

```
ggbiplot(mtcars.pca,ellipse=TRUE,choices=c(3,4),
 labels=rownames(mtcars), groups=mtcars.country)

ggbiplot(mtcars.pca,ellipse=TRUE,obs.scale = 1, var.scale = 1,
 labels=rownames(mtcars), groups=mtcars.country)

ggbiplot(mtcars.pca,ellipse=TRUE,obs.scale = 1, var.scale =
 1,var.axes=FALSE,   labels=rownames(mtcars),
  groups=mtcars.country)

ggbiplot(mtcars.pca,ellipse=TRUE,obs.scale = 1, var.scale = 1,
 labels=rownames(mtcars), groups=mtcars.country) +
  scale_colour_manual(name="Origin", values= c("forest green", "red3",
   "dark blue"))+
  ggtitle("PCA of mtcars dataset")+
  theme_minimal()+
  theme(legend.position = "bottom")
```

Suppose you want to add a new sample to your dataset. This is a very special car, with stats unlike any other. It's super-powerful, has a 60-cylinder engine, amazing fuel economy, no gears and is very light. It's a "spacecar", from Jupiter.

```
spacecar <- c(1000,60,50,500,0,0.5,2.5,0,1,0,0)

mtcarsplus <- rbind(mtcars, spacecar)
mtcars.countryplus <- c(mtcars.country, "Jupiter")

mtcarsplus.pca <- prcomp(mtcarsplus[,c(1:7,10,11)], center = TRUE,scale. = TRUE)

ggbiplot(mtcarsplus.pca, obs.scale = 1, var.scale = 1, ellipse = TRUE,
 circle = FALSE, var.axes=TRUE, labels=c(rownames(mtcars),
  "spacecar"), groups=mtcars.countryplus)+
  scale_colour_manual(name="Origin", values= c("forest green", "red3",
   "violet", "dark blue"))+
  ggtitle("PCA of mtcars dataset, with extra sample added")+
  theme_minimal()+
  theme(legend.position = "bottom")

s.sc <- scale(t(spacecar[c(1:7,10,11)]), center= mtcars.pca$center)
s.pred <- s.sc %*% mtcars.pca$rotation


mtcars.plusproj.pca <- mtcars.pca
mtcars.plusproj.pca$x <- rbind(mtcars.plusproj.pca$x, s.pred)


ggbiplot(mtcars.plusproj.pca, obs.scale = 1, var.scale = 1, ellipse =
 TRUE, circle = FALSE, var.axes=TRUE, labels=c(rownames(mtcars),
  "spacecar"), groups=mtcars.countryplus)+
  scale_colour_manual(name="Origin", values= c("forest green", "red3",
   "violet", "dark blue"))+
  ggtitle("PCA of mtcars dataset, with extra sample projected")+
  theme_minimal()+
  theme(legend.position = "bottom")
```

It is also possible to arrive at some of the vector plots more manually. Explore the following code and determine what similarities exist with the previous process using ggbiplot.

```
# Extract the loadings
(phi <- mtcars.eigen$vectors[,1:2])
phi <- -phi
row.names(phi) <- c("mpg", "cyl", "disp", "hp",
                    "drat", "wt", "qsec", "vs",
```

```
                    "am", "gear", "carb")
colnames(phi) <- c("PC1", "PC2")
phi

# Calculate Principal Components scores
PC1 <- as.matrix(scaled_df) %*% phi[,1]
PC2 <- as.matrix(scaled_df) %*% phi[,2]

# Create data frame with Principal Components scores
PC <- data.frame(State = row.names(mtcars), PC1, PC2)
head(PC)

ggplot(PC, aes(PC1, PC2)) +
  modelr::geom_ref_line(h = 0) +
  modelr::geom_ref_line(v = 0) +
  geom_text(aes(label = State), size = 3) +
  xlab("First Principal Component") +
  ylab("Second Principal Component") +
  ggtitle("First Two Principal Components of USArrests Data")

pca_result <- prcomp(mtcars, scale = TRUE)
names(pca_result)

# means
pca_result$center
# standard deviations
pca_result$scale

pca_result$rotation

pca_result$rotation <- -pca_result$rotation
pca_result$rotation

pca_result$x <- - pca_result$x
head(pca_result$x)

biplot(pca_result, scale = 0)
```

4 Homework Questions

Use the USArrests data to answer each of these questions. You can obtain the USArrests dataset from R:

```
library(datasets)
data(USArrests)
View(USArrests)
```

1. Using the prcomp command, obtain the proportion of variance explained (PVE) by each component and provide them accompanied by a plot of the PVE by component in your submission. Include the code.

2. Using the prcomp command, obtain the cumulative sum of the PVE by each component and provide them accompanied by a plot of the cumulative sum of the PVE in your submission. Include the code.

3. Use the eigenvalues to calculate the PVE and include the combined plot of a graph with PVE by component and a graph with the cumulative sum of the PVE. Include the code in your submission.

4. Obtain a PCA plot with the two most important principal components. Compare this with a PCA plot of the third and fourth most important principal components.

5. Divide Urban Population into a binary outcome (remember you will need to find a splitting rule for urban population into high and low). Create the PCA plot with the two most important principal components grouping by the binary urban population outcome. Include the vector plot in your submission.

6. Repeat question five using the second and third most important principal components. Include the vector plot in your submission.