Big Data and High-Dimensional Data Analysis
Exercise 4

Oleg Seleznjev

December 18, 2017

Last day to hand in the lab: **2018-01-14**.

Each task is graded: Very Good (VG), Good (G) and Not Good (NG). A grading can be given conditionally, e.g., you will get the grade VG if you address the comments and remarks made by the teacher. The grading rules:
Grade 5: All tasks are solved, handed in in time and graded VG. Grade 4: Task 1 and 2 are solved, handed in in time and all are graded VG. Grade 3: Task 1 and 2 are solved, handed in in time and all are graded G or VG.

Background

In this computer exercise, we practice mostly clustering techniques (unsupervised learning) (lectures 15-16) for genetic data. A central problem is to clusterize the data to sub-type of cancer the patient have and to compare the results to given information (cancer types = labels). This problem is an unsupervised problem.

Description of the data

Analysis of cell lines from 9 different cancer tissue of origin types (Breast, Central Nervous System, Colon, Leukemia, Melanoma, Non-Small Cell Lung, Ovarian, Prostate, and Renal) from the NCI-60 microarray data. The National Cancer Institute-60 (NCI-60) cell lines are among the most widely used models of human cancer. Results provide insight into molecular mechanisms underlying the various cancer types.
Data consist of 6, 830 gene expression measurements on 64 cancer cell lines and are available in R, library(ISLR), (*NCI60* data-frame). Each cell line (NCI60$data) is labeled with a cancer type (NCI60$labs). We don't use these label knowledge (unsupervised learning) but after clustering compare the results with these labels cancer types. More information about the data is in library ISLR.

Tasks

Task 1. PCA for data.

Unsupervised techniques are often used in the analysis of genomic data. In particular, PCA and hierarchical clustering are popular tools.

```
library(ISLR)
nci.labs=NCI60$labs
nci.data=NCI60$data
dim(nci.data)
```

We can examine the cancer types for the cell lines

```
table(nci.labs)
```

Next PCA on the data after scaling the variables (genes) to have standard deviation one (variations are possible even without scaling)

```
pr.out = prcomp(nci.data, scale = TRUE)
```

Now plot the first few principal component score vectors, in order to visualize the data. The observations (cell lines) corresponding to a given cancer type will be plotted in the same color, so that you can see to what extent the observations within a cancer type are similar to each other.

```
Cols=function(vec){
cols=rainbow(length(unique(vec)))
return(cols[as.numeric(as.factor(vec))])
}
```

Select $P1, P2$ and $P1, P3$ principal components.

```
par(mfrow = c(1, 2))
plot(pr.out$x[, 1 : 2], col = Cols(nci.labs), pch = 19,
xlab = ”P 1”, ylab = ”P 2”)
plot(pr.out$x[, c(1, 3)], col = Cols(nci.labs), pch = 19,
xlab = ”P 1”, ylab = ”P 3”)
```

Next obtain a summary of the proportion of variance explained (PVE) of the first few principal components using the *summary*() method for a *prcomp* object.

```
summary(pr.out)
```

Now plot the variance explained by the first few principal components

```
plot(pr.out)
```

More informative to plot the PVE of each principal component (i.e. a scree plot) and the cumulative PVE of each principal component (to searching *elbow* in the plot).

```
pve=100*pr.out$sdev^2/sum(pr.out$sdev^2)
par(mfrow=c(1,2))
plot(pve,  type="o", ylab="PVE", xlab="Principal Component", col="blue")
plot(cumsum(pve), type="o", ylab="Cumulative PVE", xlab="Principal Component",
col="brown3")
```

Task 2 Clustering

Task 2a Hierarchial clustering the observations (NCI60 data)

First you hierarchically clusterize the cell lines in the NCI60 data, with the goal of finding out whether or not the observations cluster into distinct types of cancer. To begin, standardize the variables to have mean zero and standard deviation one. As mentioned earlier, this step is optional and should be performed only if we want each gene to be on the same scale.

```
sd.data = scale(nci.data)
```

Now perform hierarchical clustering of the observations using complete, single, and average linkage (as in Lecture R-examples). Euclidean distance is used as the dissimilarity measure.

```
par(mfrow=c(1,3))

data.dist=dist(sd.data)

plot(hclust(data.dist), labels=nci.labs, main="Complete Linkage", xlab="", sub="",ylab="")

plot(hclust(data.dist, method="average"), labels=nci.labs, main="Average Linkage",
xlab="", sub="",ylab="")

plot(hclust(data.dist, method="single"), labels=nci.labs, main="Single Linkage", xlab="",
sub="",ylab="")

hc.out=hclust(dist(sd.data))
```

Cut the dendrogram at the height that will yield a particular number of clusters,(e.g., 4), and compare with known labels:

```
hc.out=hclust(dist(sd.data))
hc.clusters=cutree(hc.out,4)
table(hc.clusters,nci.labs)
```

Plot the cut on the dendrogram that produces these four clusters

```
par(mfrow=c(1,1))
plot(hc.out, labels=nci.labs)
abline(h=139, col="red")
```

Task 2b K**-Means clustering the observations (NCI60 data)**

Perform K-means clustering with $K = 4$ (or more?) and compare with previous hierarchical clustering results

```
set.seed(2)

km.out=kmeans(sd.data, 4, nstart=20)

km.clusters=km.out$cluster

table(km.clusters,hc.clusters)
```

Task 2c Various number of clusters for (NCI60 data)

Apply Silhouette plot in R (function ***silhouette*** in package **cluster**).

You can use also Group Within-Cluster Variation $WCVG$. Run K-means for various number of clusters and plot the $WCVG$ (= within groups sum of squares) (Group within-cluster variation) versus the number of clusters (as in R-examples).

Comment

Discuss the obtained results step by step of your Lab-Exercise. Try to justify and convince me (and yourself) what that you found the optimal clustering of the data. Sure there are no universal techniques for all types of data, e.g., for justification may be *WCVG* plot is sufficient, especially when in your example you know data labels.