

TMA4268 Statistical Learning

Chapter 10: Solution sketches

Thiago G. Martins, Department of Mathematical Sciences, NTNU

Spring 2022

Recommended exercise 1

Get the data

The `pca-examples.rdata` can be downloaded from the Blackboard.

```
# Modify the location based on your filesystem
load("../10_unsupervised_learning/datasets/pca-examples.Rdata")

# We will work with nyt.frame
nyt_data = nyt.frame
```

Compute the PCA

```
nyt_pca = prcomp(nyt_data[, -1])
```

Default biplot

Too much information on the graph, we should select only a few loading vectors to display.

```
biplot(nyt_pca, scale=0, cex = 0.5)
```



```
## [96] 6.420743e-03 6.337557e-03 6.233789e-03 6.191162e-03 6.025576e-03
## [101] 5.283112e-03 3.582418e-32
```

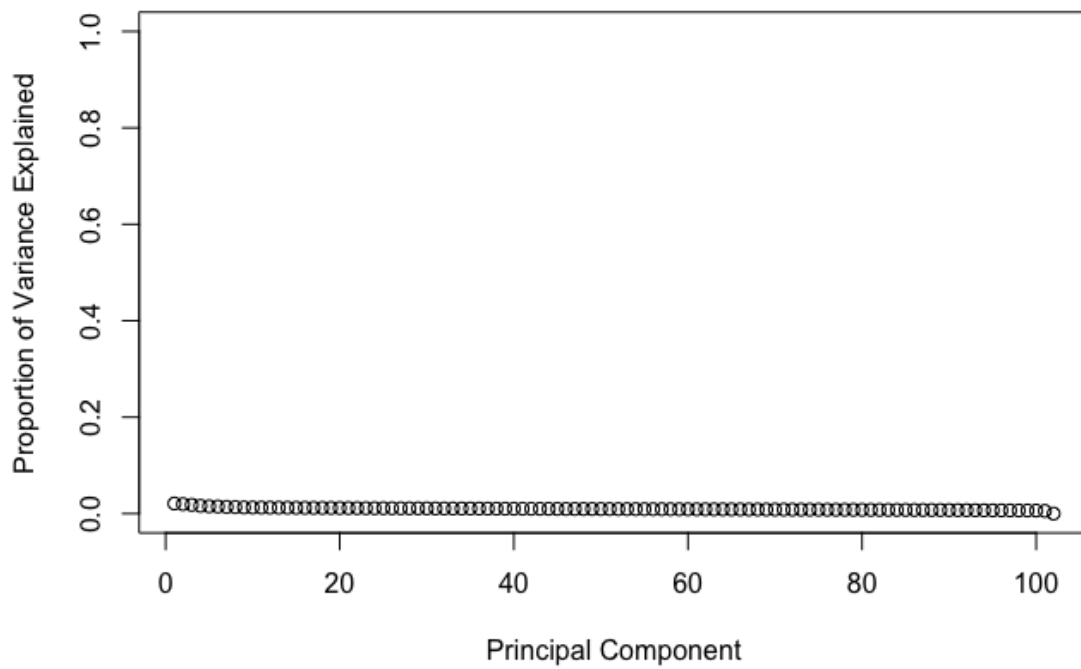
We can then compute the proportion of variance explained by each principal component

```
pve=pr.var/sum(pr.var)
pve
```

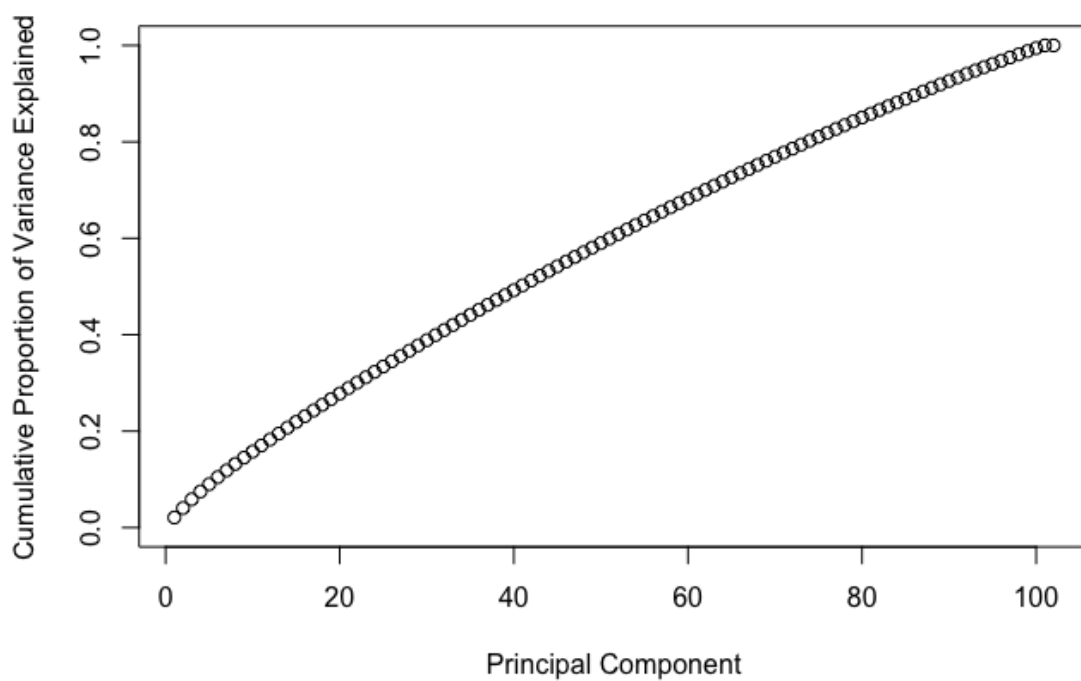
```
## [1] 2.093766e-02 1.961121e-02 1.815344e-02 1.618390e-02 1.516547e-02
## [6] 1.466884e-02 1.382931e-02 1.352942e-02 1.293206e-02 1.271664e-02
## [11] 1.254110e-02 1.241097e-02 1.231328e-02 1.221072e-02 1.200786e-02
## [16] 1.193492e-02 1.183985e-02 1.171789e-02 1.168626e-02 1.152892e-02
## [21] 1.144031e-02 1.129495e-02 1.116105e-02 1.114063e-02 1.112318e-02
## [26] 1.100644e-02 1.090434e-02 1.087014e-02 1.085155e-02 1.076808e-02
## [31] 1.064037e-02 1.057678e-02 1.054310e-02 1.050432e-02 1.047104e-02
## [36] 1.040726e-02 1.024608e-02 1.019026e-02 1.014556e-02 1.008377e-02
## [41] 1.003900e-02 9.966275e-03 9.935469e-03 9.896187e-03 9.798624e-03
## [46] 9.689594e-03 9.664439e-03 9.615711e-03 9.583269e-03 9.519011e-03
## [51] 9.474062e-03 9.397135e-03 9.353120e-03 9.316071e-03 9.264001e-03
## [56] 9.209081e-03 9.177880e-03 9.100248e-03 9.011858e-03 8.979933e-03
## [61] 8.917003e-03 8.914481e-03 8.837220e-03 8.818530e-03 8.687444e-03
## [66] 8.670504e-03 8.580061e-03 8.552182e-03 8.469419e-03 8.439092e-03
## [71] 8.403578e-03 8.304829e-03 8.223201e-03 8.207404e-03 8.184854e-03
## [76] 8.089223e-03 8.034980e-03 8.017438e-03 7.946618e-03 7.878128e-03
## [81] 7.811304e-03 7.755204e-03 7.728131e-03 7.673021e-03 7.653000e-03
## [86] 7.554753e-03 7.538811e-03 7.484412e-03 7.407710e-03 7.352964e-03
## [91] 7.256556e-03 7.164773e-03 7.022798e-03 6.913259e-03 6.808359e-03
## [96] 6.759786e-03 6.672208e-03 6.562961e-03 6.518083e-03 6.343753e-03
## [101] 5.562084e-03 3.771586e-32
```

We can plot the PVE explained by each component, as well as the cumulative PVE, as follows:

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



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



Recommended exercise 2

The answer is on page 388 of the book, with the explanation around Equation (10.12)

Recommended exercise 3

Perform k-means

```
km.out=kmeans(nyt_data[, -1], 2, nstart=20)
```

Cluster assignments

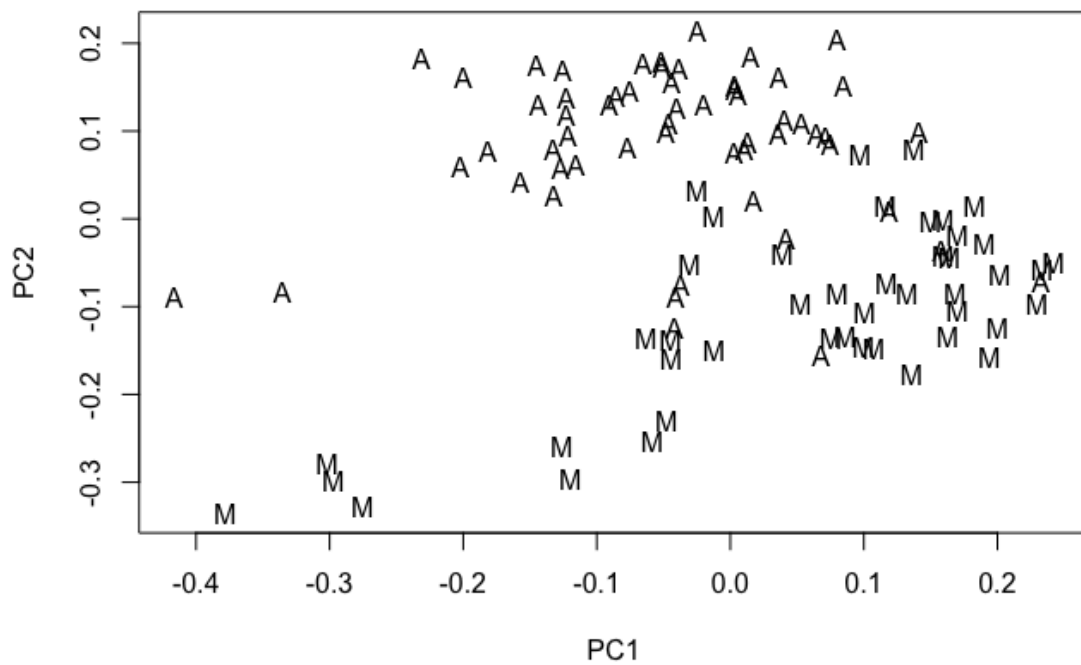
```
km.out$cluster
```

```
##    [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 1 1
##   [36] 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 1 1 1 1 2 1 1 1 1 1 1 1
##   [71] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1
```

Plot the data

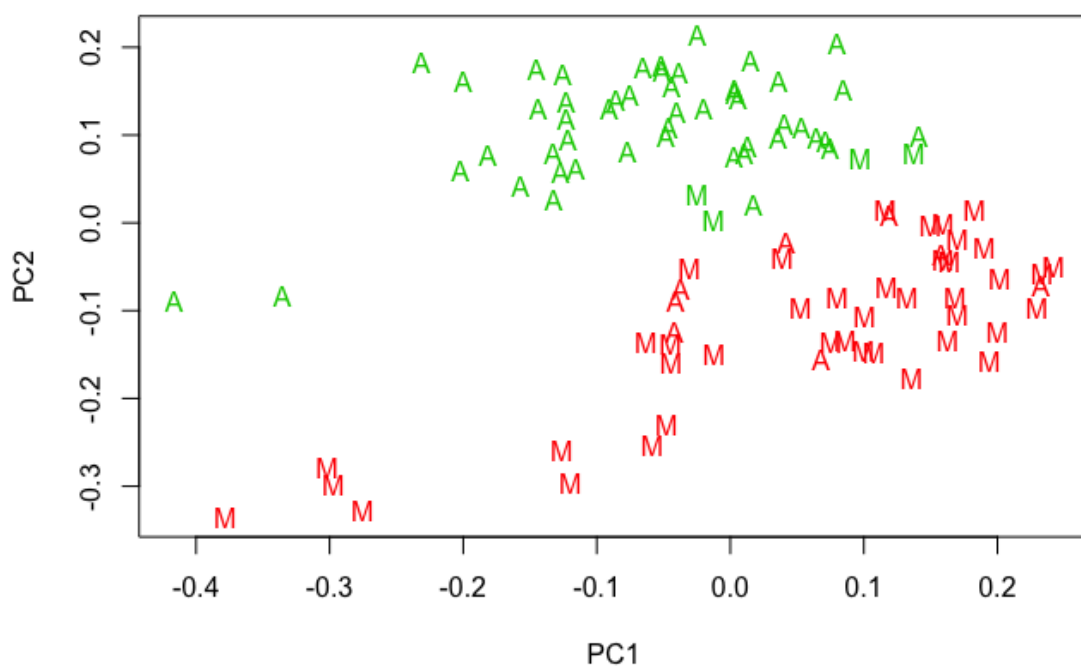
To plot the data we need to use the PCA projections. Below we will use the plot based on PCA with true labels (A for art and M for music) and compare with the plot that color the points according to the k-means clustering.

```
# PCA with true labels
plot(nyt_pca$x[, 1:2], type="n")
points(nyt_pca$x[nyt_data[, "class.labels"]=="art", 1:2], pch="A")
points(nyt_pca$x[nyt_data[, "class.labels"]=="music", 1:2], pch="M")
```



PCA with true labels but colored by k-means clustering

```
plot(nyt_pca$x[,1:2], type="n")
points(nyt_pca$x[nyt_data[, "class.labels"]=="art", 1:2], pch="A", col=
      (km.out$cluster+1)[nyt_data[, "class.labels"]=="art"])
points(nyt_pca$x[nyt_data[, "class.labels"]=="music", 1:2], pch="M", col=
      (km.out$cluster+1)[nyt_data[, "class.labels"]=="music"])
```



Recommended exercise 4

Perform hierarchical clustering

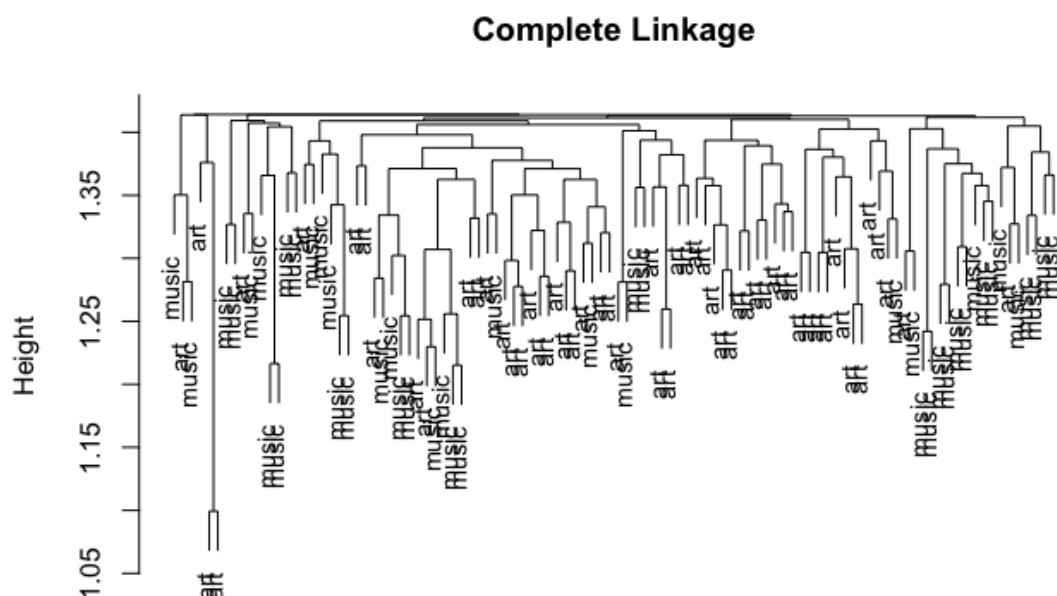
I will use euclidean distance and complete linkage

```
hc.complete=hclust(dist(nyt_data[,-1]), method="complete")
str(hc.complete)
```

```
## List of 7
## $ merge      : int [1:101, 1:2] -46 -84 -68 -18 -65 -23 -71 -64 -95 -37 ...
## $ height     : num [1:101] 1.1 1.22 1.22 1.23 1.24 ...
## $ order      : int [1:102] 94 24 93 16 46 55 59 87 39 90 ...
## $ labels     : NULL
## $ method     : chr "complete"
## $ call       : language hclust(d = dist(nyt_data[, -1]), method = "complete")
## $ dist.method: chr "euclidean"
## - attr(*, "class")= chr "hclust"
```

Plot dendrograms

```
plot(hc.complete,main="Complete Linkage", labels=as.character(nyt_data[,1]),
      xlab="", sub="", cex=.9)
```



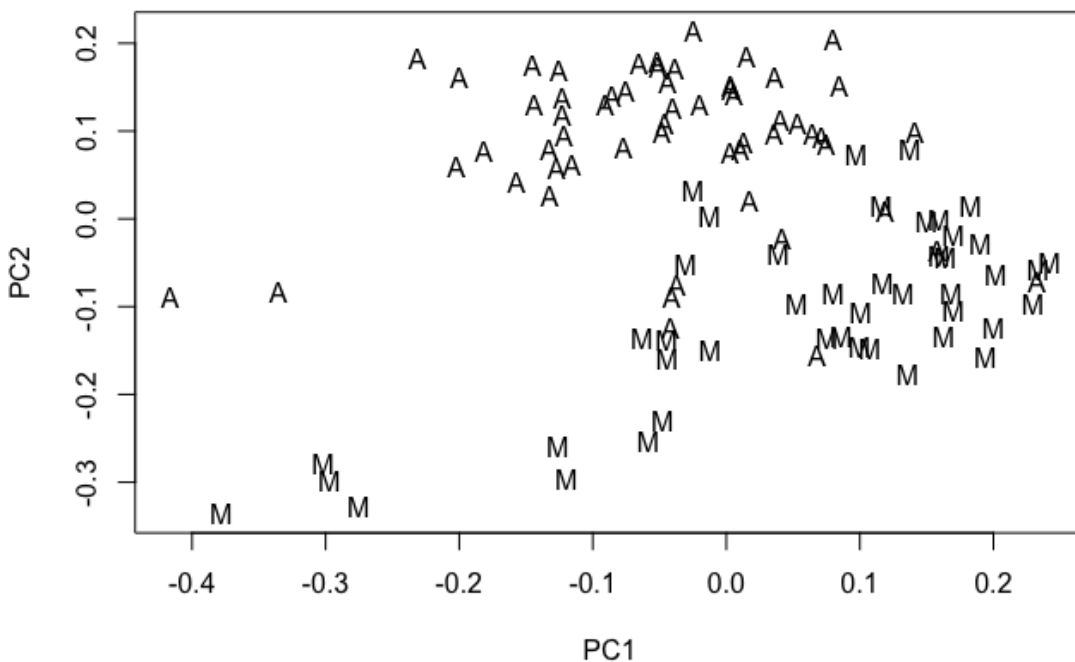
Cluster assignment

```
hc.clusters=cutree(hc.complete, 2)
```

Plot the data

Based on the cluster assignment above and the plots below we see that hierarchical clustering performs worse than k-means for the same number of clusters (K=2)

```
# PCA with true labels
plot(nyt_pca$x[,1:2],type="n")
points(nyt_pca$x[nyt_data[, "class.labels"]=="art",1:2],pch="A")
points(nyt_pca$x[nyt_data[, "class.labels"]=="music",1:2],pch="M")
```



```
# PCA with true labels but colored by k-means clustering
plot(nyt_pca$x[,1:2],type="n")
points(nyt_pca$x[nyt_data[, "class.labels"]=="art",1:2],pch="A",col=(hc.clusters+1)
      [nyt_data[, "class.labels"]=="art"])
points(nyt_pca$x[nyt_data[, "class.labels"]=="music",1:2],pch="M",col=
      (hc.clusters+1)[nyt_data[, "class.labels"]=="music"])
```

