Unsupervised Learning

Fabian Peri

October 10, 2018

```
# packages needed for chapter 7
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
library(ascii)
```

```
##
## Attaching package: 'ascii'

## The following object is masked from 'package:tidyr':
##
##     expand
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```
library(ellipse)
```

```
##
## Attaching package: 'ellipse'

## The following object is masked from 'package:graphics':
##
##     pairs
```

```
library(mclust)
```

```
## Package 'mclust' version 5.4.1
## Type 'citation("mclust")' for citing this R package in publications.
```

```
library(cluster)
```

```
# Import the datasets needed for chapter 7
PSDS_PATH <- file.path('C:/Users/fabia/Desktop', 'psds_data')
```

```
## Import datasets needed for chapter 7
sp500_px <- read.csv(file.path(PSDS_PATH, 'data', 'sp500_px.csv'), row.names = 1)
sp500_sym <- read.csv(file.path(PSDS_PATH, 'data', 'sp500_sym.csv'), stringsAsFactors = FALSE)
loan_data <- read.csv(file.path(PSDS_PATH, 'data', 'loan_data.csv'))
loan_data$outcome <- ordered(loan_data$outcome, levels=c('paid off', 'default'))
```

```
## PCA for oil data
#oil_px = as.data.frame(scale(oil_px, scale=FALSE))
oil_px <- sp500_px[, c('CVX', 'XOM')]
pca <- princomp(oil_px)
pca$loadings
```

```
##
## Loadings:
##     Comp.1 Comp.2
## CVX -0.747  0.665
## XOM -0.665 -0.747
##
##                Comp.1 Comp.2
## SS loadings       1.0    1.0
## Proportion Var    0.5    0.5
## Cumulative Var    0.5    1.0
```

```
## Figure 7-1: principal components for oil stock data
png(filename=file.path(PSDS_PATH, 'figures', 'psds_0701.png'), width = 4, height=4, units='in', res=300
loadings <- pca$loadings
ggplot(data=oil_px, aes(x=CVX, y=XOM)) +
  geom_point(alpha=.3) +
  scale_shape_manual(values=c(46)) +
  stat_ellipse(type='norm', level=.99, color='grey25') +
  geom_abline(intercept = 0, slope = loadings[2,1]/loadings[1,1], color='grey25', linetype=2) +
  geom_abline(intercept = 0, slope = loadings[2,2]/loadings[1,2],  color='grey25', linetype=2) +
  scale_x_continuous(expand=c(0,0), lim=c(-3, 3)) +
  scale_y_continuous(expand=c(0,0), lim=c(-3, 3)) +
  theme_bw()
```

```
## Warning: Removed 33 rows containing non-finite values (stat_ellipse).
```

```
## Warning: Removed 33 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
## Figure 7-2: screeplot
png(filename=file.path(PSDS_PATH, 'figures', 'psds_0702.png'), width = 4, height=4, units='in', res=300

syms <- c( 'AAPL', 'MSFT', 'CSCO', 'INTC', 'CVX', 'XOM', 'SLB', 'COP',
           'JPM', 'WFC', 'USB', 'AXP', 'WMT', 'TGT', 'HD', 'COST')
top_cons <- sp500_px[row.names(sp500_px)>='2011-01-01', syms]
sp_pca <- princomp(top_cons)
par(mar=c(6,3,0,0)+.1, las=2)
screeplot(sp_pca, main='')

dev.off()
```

```
## pdf
##   2
```

```
## Loadings for stock data
loadings = sp_pca$loadings[,1:5]
loadings <- as.data.frame(loadings)
loadings$Symbol <- row.names(loadings)
loadings <- gather(loadings, "Component", "Weight", -Symbol)
head(loadings)
```

```
##   Symbol Component      Weight
## 1   AAPL    Comp.1 -0.30082485
## 2   MSFT    Comp.1 -0.10501241
## 3   CSCO    Comp.1 -0.06405912
## 4   INTC    Comp.1 -0.07695580
## 5    CVX    Comp.1 -0.44449001
## 6    XOM    Comp.1 -0.31795201
```

```
## Figure 7-3: Plot of component loadings
png(filename=file.path(PSDS_PATH, 'figures', 'psds_0703.png'), width = 4, height=4, units='in', res=300

loadings$Color = loadings$Weight > 0
ggplot(loadings, aes(x=Symbol, y=Weight, fill=Color)) +
  geom_bar(stat='identity', position = "identity", width=.75) +
  facet_grid(Component ~ ., scales='free_y') +
  guides(fill=FALSE)  +
  ylab('Component Loading') +
  theme_bw() +
  theme(axis.title.x = element_blank(),
        axis.text.x  = element_text(angle=90, vjust=0.5))

dev.off()
```

```
## pdf
##   2
```

```
## K-means chapter

set.seed(1010103)
df <- sp500_px[row.names(sp500_px)>='2011-01-01', c('XOM', 'CVX')]
km <- kmeans(df, centers=4, nstart=1)

df$cluster <- factor(km$cluster)
head(df)
```

```
##                   XOM        CVX cluster
## 2011-01-03 0.73680496  0.2406809       3
## 2011-01-04 0.16866845 -0.5845157       1
## 2011-01-05 0.02663055  0.4469854       3
## 2011-01-06 0.24855834 -0.9197513       1
## 2011-01-07 0.33732892  0.1805111       3
## 2011-01-10 0.00000000 -0.4641675       1
```

```
centers <- data.frame(cluster=factor(1:4), km$centers)
centers
```

```
##   cluster         XOM        CVX
## 1       1 -0.3287416 -0.5734695
## 2       2  0.9270317  1.3464117
## 3       3  0.2315403  0.3169645
## 4       4 -1.1439800 -1.7502975
```

```
## Figure 7-4: K-means clusters for two stocks
png(filename=file.path(PSDS_PATH, 'figures', 'psds_0704.png'), width = 4, height=3, units='in', res=300

ggplot(data=df, aes(x=XOM, y=CVX, color=cluster, shape=cluster)) +
  geom_point(alpha=.3) +
  scale_shape_manual(values = 1:4,
                     guide = guide_legend(override.aes=aes(size=1))) +
  geom_point(data=centers,  aes(x=XOM, y=CVX), size=2, stroke=2)  +
  theme_bw() +
  scale_x_continuous(expand=c(0,0), lim=c(-2, 2)) +
  scale_y_continuous(expand=c(0,0), lim=c(-2.5, 2.5))
```

```
## Warning: Removed 37 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
## cluster means algorithm
syms <- c( 'AAPL', 'MSFT', 'CSCO', 'INTC', 'CVX', 'XOM', 'SLB', 'COP',
           'JPM', 'WFC', 'USB', 'AXP', 'WMT', 'TGT', 'HD', 'COST')
df <- sp500_px[row.names(sp500_px)>='2011-01-01', syms]

set.seed(10010)
km <- kmeans(df, centers=5, nstart=10)
km$size
```

```
## [1] 106 186 285 288 266
```

```
centers <- km$centers

#centers <- scale(scale(centers, center=FALSE, scale=1/attr(df, 'scaled:scale')),
#                  center=-attr(df, 'scaled:center'), scale=FALSE)

## Figure 7-5 interpreting the clusters
centers <- as.data.frame(t(centers))
names(centers) <- paste("Cluster", 1:5)
centers$Symbol <- row.names(centers)
centers <- gather(centers, "Cluster", "Mean", -Symbol)

png(filename=file.path(PSDS_PATH, 'figures', 'psds_0705.png'), width = 4, height=5, units='in', res=300

centers$Color = centers$Mean > 0
ggplot(centers, aes(x=Symbol, y=Mean, fill=Color)) +
  geom_bar(stat='identity', position = "identity", width=.75) +
  facet_grid(Cluster ~ ., scales='free_y') +
  guides(fill=FALSE)  +
  ylab('Component Loading') +
  theme_bw() +
  theme(axis.title.x = element_blank(),
        axis.text.x  = element_text(angle=90, vjust=0.5))
```

```r
dev.off()
```

```
## pdf
##   2
```

```r
## Figure 7-6: selecting the number of clusters (elbow plot)
pct_var <- data.frame(pct_var = 0,
                      num_clusters=2:14)
totalss <- kmeans(df, centers=14, nstart=50, iter.max = 100)$totss
for(i in 2:14){
  pct_var[i-1, 'pct_var'] <- kmeans(df, centers=i, nstart=50, iter.max = 100)$betweenss/totalss
}

png(filename=file.path(PSDS_PATH, 'figures', 'psds_0706.png'), width = 4, height=3, units='in', res=300

ggplot(pct_var, aes(x=num_clusters, y=pct_var)) +
  geom_line() +
  geom_point() +
  labs(y='% Variance Explained', x='Number of Clusters') +
  scale_x_continuous(breaks=seq(2, 14, by=2))   +
  theme_bw()
dev.off()
```

```
## pdf
##   2
```

```r
## hclust chapter

syms1 <- c('GOOGL', 'AMZN', 'AAPL', 'MSFT', 'CSCO', 'INTC', 'CVX',
           'XOM', 'SLB', 'COP', 'JPM', 'WFC', 'USB', 'AXP',
           'WMT', 'TGT', 'HD', 'COST')

df <- sp500_px[row.names(sp500_px)>='2011-01-01', syms1]
d <- dist(t(df))
hcl <- hclust(d)

## Figure 7-7: dendograme of stock data
png(filename=file.path(PSDS_PATH, 'figures', 'psds_0707.png'), width = 4, height=4, units='in', res=300

par(cex=.75, mar=c(0, 5, 0, 0)+.1)
plot(hcl, ylab='distance', xlab='', sub='', main='')

dev.off()
```

```
## pdf
##   2
```

```r
## Figure 7-8: comparison of the different measuresof dissimilarity
cluster_fun <- function(df, method)
{
  d <- dist(df)
  hcl <- hclust(d, method=method)
  tree <- cutree(hcl, k=4)
  df$cluster <- factor(tree)
  df$method <- method
  return(df)
```

```
}

df0 <- sp500_px[row.names(sp500_px)>='2011-01-01', c('XOM', 'CVX')]
df <- rbind(cluster_fun(df0, method='single'),
            cluster_fun(df0, method='average'),
            cluster_fun(df0, method='complete'),
            cluster_fun(df0, method='ward.D'))
df$method <- ordered(df$method, c('single', 'average', 'complete', 'ward.D'))

png(filename=file.path(PSDS_PATH, 'figures', 'psds_0708.png'), width = 5.5, height=4, units='in', res=30

ggplot(data=df, aes(x=XOM, y=CVX, color=cluster, shape=cluster)) +
  geom_point(alpha=.3) +
  scale_shape_manual(values = c(46, 3, 1,  4),
                     guide = guide_legend(override.aes=aes(size=2))) +
  facet_wrap( ~ method) +
  theme_bw()

dev.off()
```

```
## pdf
##   2
```

```
# Model-based clusting
# Multivariate normal

mu <- c(.5, -.5)
sigma <- matrix(c(1, 1, 1, 2), nrow=2)
prob <- c(.5, .75, .95, .99) ## or whatever you want
names(prob) <- prob ## to get id column in result
x <- NULL
for (p in prob){
  x <- rbind(x,  ellipse(x=sigma, centre=mu, level=p))
}
df <- data.frame(x, prob=factor(rep(prob, rep(100, length(prob)))))
names(df) <- c("X", "Y", "Prob")

## Figure 7-9: Multivariate normal ellipses
dfmu <- data.frame(X=mu[1], Y=mu[2])
png(filename=file.path(PSDS_PATH, 'figures', 'psds_0709.png'), width = 4, height=4, units='in', res=300)

ggplot(df, aes(X, Y)) +
  geom_path(aes(linetype=Prob)) +
  geom_point(data=dfmu, aes(X, Y), size=3) +
  theme_bw()

dev.off()
```

```
## pdf
##   2
```

```
## Figure 7-10 mclust applied XOM and CVX

df <- sp500_px[row.names(sp500_px)>='2011-01-01', c('XOM', 'CVX')]
mcl <- Mclust(df)
```

```
summary(mcl)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEE (ellipsoidal, equal shape and orientation) model with 2
## components:
##
##  log.likelihood    n df       BIC       ICL
##       -2255.125 1131  9 -4573.528 -5075.657
##
## Clustering table:
##   1   2
## 168 963
```

```
cluster <- factor(predict(mcl)$classification)
png(filename=file.path(PSDS_PATH, 'figures', 'psds_0710.png'), width = 5, height=4, units='in', res=300

ggplot(data=df, aes(x=XOM, y=CVX, color=cluster, shape=cluster)) +
  geom_point(alpha=.8) +
  theme_bw() +
  scale_shape_manual(values = c(46, 3),
                     guide = guide_legend(override.aes=aes(size=2)))

dev.off()
```

```
## pdf
##   2
```

```
summary(mcl, parameters=TRUE)$mean
```

```
##            [,1]       [,2]
## XOM -0.04362218 0.05792282
## CVX -0.21109525 0.07375447
```

```
summary(mcl, parameters=TRUE)$variance
```

```
## , , 1
##
##          XOM      CVX
## XOM 1.044671 1.065190
## CVX 1.065190 1.912748
##
## , , 2
##
##           XOM       CVX
## XOM 0.2998935 0.3057838
## CVX 0.3057838 0.5490920
```

```
## Figure 7-11: BIC scores for the different models fit by mclust

png(filename=file.path(PSDS_PATH, 'figures', 'psds_0711.png'), width = 4, height=4, units='in', res=300

par(mar=c(4, 5, 0, 0)+.1)
plot(mcl, what='BIC', ask=FALSE, cex=.75)
```

```
dev.off()
```

```
## pdf
##   2
```

```
#
```

```
# Scaling chapter

defaults <- loan_data[loan_data$outcome=='default',]
df <- defaults[, c('loan_amnt', 'annual_inc', 'revol_bal', 'open_acc', 'dti', 'revol_util')]
km <- kmeans(df, centers=4, nstart=10)
centers <- data.frame(size=km$size, km$centers)
round(centers, digits=2)
```

```
##    size loan_amnt annual_inc revol_bal open_acc   dti revol_util
## 1    52  22570.19  489783.40  85161.35    13.33  6.91      59.65
## 2  7579  18247.71   83069.61  19587.30    11.66 16.79      62.26
## 3  1221  21797.26  164503.32  38652.54    12.61 13.53      63.65
## 4 13819  10577.04   42380.98  10245.27     9.58 17.71      58.09
```

```
df0 <- scale(df)
km0 <- kmeans(df0, centers=4, nstart=10)
centers0 <- scale(km0$centers, center=FALSE, scale=1/attr(df0, 'scaled:scale'))
centers0 <- scale(centers0, center=-attr(df0, 'scaled:center'), scale=FALSE)
centers0 <- data.frame(size=km0$size, centers0)
round(centers0, digits=2)
```

```
##   size loan_amnt annual_inc revol_bal open_acc   dti revol_util
## 1 5309  10363.43   53523.09   6038.26     8.68 11.32      30.70
## 2 6294  13361.61   55596.65  16375.27    14.25 24.23      59.61
## 3 3713  25894.07  116185.91  32797.67    12.41 16.22      66.14
## 4 7355  10467.65   51134.87  11523.31     7.48 15.78      77.73
```

```
km <- kmeans(df, centers=4, nstart=10)
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 1133550)
```

```
centers <- data.frame(size=km$size, km$centers)
round(centers, digits=2)
```

```
##    size loan_amnt annual_inc revol_bal open_acc   dti revol_util
## 1 13902  10606.48   42500.30  10280.52     9.59 17.71      58.11
## 2    52  22570.19  489783.40  85161.35    13.33  6.91      59.65
## 3  7525  18282.25   83458.11  19653.82    11.66 16.77      62.27
## 4  1192  21856.38  165473.54  38935.88    12.61 13.48      63.67
```

```
## Figure 7-12: screeplot for data with dominant variables

syms <- c('GOOGL', 'AMZN', 'AAPL', 'MSFT', 'CSCO', 'INTC', 'CVX', 'XOM',
          'SLB', 'COP', 'JPM', 'WFC', 'USB', 'AXP', 'WMT', 'TGT', 'HD', 'COST')
top_15 <- sp500_px[row.names(sp500_px)>='2011-01-01', syms]
sp_pca1 <- princomp(top_15)

png(filename=file.path(PSDS_PATH, 'figures', 'psds_0712.png'), width = 4, height=4, units='in', res=300

par(mar=c(6,3,0,0)+.1, las=2)
```

```r
screeplot(sp_pca1, main='')

dev.off()
```

```
## pdf
##   2
```

```r
round(sp_pca1$loadings[,1:2], 3)
```

```
##       Comp.1 Comp.2
## GOOGL  0.781  0.609
## AMZN   0.593 -0.792
## AAPL   0.078  0.004
## MSFT   0.029  0.002
## CSCO   0.017 -0.001
## INTC   0.020 -0.001
## CVX    0.068 -0.021
## XOM    0.053 -0.005
## SLB    0.079 -0.013
## COP    0.044 -0.016
## JPM    0.043  0.001
## WFC    0.034 -0.001
## USB    0.026  0.003
## AXP    0.063 -0.006
## WMT    0.026 -0.001
## TGT    0.036 -0.010
## HD     0.051 -0.019
## COST   0.061 -0.019
```

```r
## Figure 7-13: Categorical data and Gower's distance

x <- loan_data[1:5, c('dti', 'payment_inc_ratio', 'home_', 'purpose_')]
x
```

```
##     dti payment_inc_ratio home_           purpose_
## 1  1.00           2.39320  RENT     major_purchase
## 2  5.55           4.57170   OWN     small_business
## 3 18.08           9.71600  RENT              other
## 4 10.08          12.21520  RENT debt_consolidation
## 5  7.06           3.90888  RENT              other
```

```r
daisy(x, metric='gower')
```

```
## Dissimilarities :
##           1         2         3         4
## 2 0.6220479
## 3 0.6863877 0.8143398
## 4 0.6329040 0.7608561 0.4307083
## 5 0.3772789 0.5389727 0.3091088 0.5056250
##
## Metric :  mixed ;  Types = I, I, N, N
## Number of objects : 5
```

```r
set.seed(301)
df <- loan_data[sample(nrow(loan_data), 250),
                c('dti', 'payment_inc_ratio', 'home_', 'purpose_')]
d = daisy(df, metric='gower')
```

```
hcl <- hclust(d)
dnd <- as.dendrogram(hcl)

png(filename=file.path(PSDS_PATH, 'figures', 'psds_0713.png'), width = 4, height=4, units='in', res=300)
par(mar=c(0,5,0,0)+.1)
plot(dnd, leaflab='none', ylab='distance')
dev.off()
```

```
## pdf
##   2
```

```
dnd_cut <- cut(dnd, h=.5)
df[labels(dnd_cut$lower[[1]]),]
```

```
##         dti payment_inc_ratio home_           purpose_
## 7565  26.72          10.29240   OWN              other
## 36140 20.16          11.73840   OWN              other
## 20974 21.63          16.12230   OWN              other
## 44532 21.22           8.37694   OWN debt_consolidation
## 39826 22.59           6.22827   OWN debt_consolidation
## 13282 31.00           9.64200   OWN debt_consolidation
## 31510 26.21          11.94380   OWN debt_consolidation
## 6693  26.96           9.45600   OWN debt_consolidation
## 7356  25.81           9.39257   OWN debt_consolidation
## 9278  21.00          14.71850   OWN debt_consolidation
## 13520 29.00          18.86670   OWN debt_consolidation
## 14668 25.75          17.53440   OWN debt_consolidation
## 19975 22.70          17.12170   OWN debt_consolidation
## 23492 22.68          18.50250   OWN debt_consolidation
```

```
## Problems in clustering with mixed data types
df <- model.matrix(~ -1 + dti + payment_inc_ratio + home_ + pub_rec_zero, data=defaults)
df0 <- scale(df)
km0 <- kmeans(df0, centers=4, nstart=10)
centers0 <- scale(km0$centers, center=FALSE, scale=1/attr(df0, 'scaled:scale'))
round(scale(centers0, center=-attr(df0, 'scaled:center'), scale=FALSE), 2)
```

```
##     dti payment_inc_ratio home_MORTGAGE home_OWN home_RENT pub_rec_zero
## 1 17.20              9.27          0.00        1      0.00         0.92
## 2 16.99              9.11          0.00        0      1.00         1.00
## 3 16.50              8.06          0.52        0      0.48         0.00
## 4 17.46              8.42          1.00        0      0.00         1.00
## attr(,"scaled:scale")
##               dti payment_inc_ratio     home_MORTGAGE          home_OWN
##         0.1305561         0.2286345         2.0190809         3.6191450
##         home_RENT      pub_rec_zero
##         2.0008117         3.5722842
## attr(,"scaled:center")
##               dti payment_inc_ratio     home_MORTGAGE          home_OWN
##       -17.1521684        -8.7700843        -0.4313440        -0.0832782
##        home_RENT      pub_rec_zero
##        -0.4853778        -0.9142958
```