Stochastic Control and Optimization Homework 4

- Akankshi Mody MSBA (am92786)

Problem 1

Akankshi Mody, am92786

HOMEWORK 4 - STOCHASTIC CONTROL & OPTIMIZATION.

Problem 1 :

Choose L, K

To max $0.05 L^{2/3} K^{1/3}$

s.t. $12L + 15K \leq 100{,}000$

$L, K \geq 0$.

let's rewrite this to fix the optimi function :

let $12L + 15K = 100{,}000$

$$L = \left(\frac{100{,}000 - 15K}{12}\right)$$

$\therefore$ Choose K

To max $0.05 \left(\frac{100{,}000 - 15K}{12}\right)^{2/3} K^{1/3}$

$L, K \geq 0$.

From optimi, we get : $K = 127.5942$

max machines $= 102.21$

$\therefore$ max machines $= 102$

```r
func1 <- function(k){
  machines= 0.05 * (((100000-(15*k))/12)^(2/3)) * (k^(1/3))
  return(-machines)
}
S=optim(100,func1,method="CG")
S$par
```

```
## [1] 127.5942
```

```r
S$value
```

```
## [1] -102.2061
```

Thus, the maximum number of machines that can be produced = 102

Problem 2

Problem 2

Choose $w_1, \dots w_{27}$

To min $\sum \sum w_i w_j \sigma_{ij}$

Subject to $\sum w_i = 1$

$w_i \geq 0$

$\sum w_i \mu_i = 0.01$

no short selling

Solve.QP structure $\Rightarrow$

$$\min \; d^T x - \frac{1}{2} x^T D x$$

$$\text{s.t} \quad A^T x \geq b$$

$d = [0, 0 \dots]$ $\quad D_{ij} = 2\sigma_{ij}$

$$A = \begin{bmatrix} 1 & -1 & \mu_1 & -\mu_1 & 1 & 0 & 0 & 0 & \dots \\ 1 & -1 & \mu_2 & -\mu_2 & 0 & 1 & 0 & 0 & 0 \; \dots \\ \vdots & \vdots & \vdots & \vdots & & & \vdots & & \\ 1 & -1 & \mu_{27} & -\mu_{27} & \underbrace{0 & 0 & 0 & 0 \dots & 1}_{diag(27)} \end{bmatrix} \qquad b = \begin{bmatrix} 1 \\ -1 \\ 0.01 \\ -0.01 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \Big\} len = 2$$

```r
library(matrixStats)

## Warning: package 'matrixStats' was built under R version 3.6.3

library(quadprog)
stocks = read.csv("homework4stocks.csv", header = TRUE)
tempdf = data.matrix(stocks[c(2:ncol(stocks))])
#colMeans(tempdf)
#colVars(tempdf)
#cor(tempdf, method = 'pearson')

m=colMeans(tempdf)
s=colSds(tempdf)

rho=cor(data.matrix(stocks[c(2:ncol(stocks))]), method = 'pearson')

covMat=diag(s) %*% rho %*% diag(s)

RVal=0.01
Dmat=2*covMat
dvec=rep(0,ncol(stocks)-1)
Amat=matrix(c(rep(1,27),rep(-1,27),m,-m),27)
bvec=c(1,-1,0.01,-0.01)

#no shorting
Amat=cbind(Amat,diag(27))
bvec=c(bvec,rep(0,27))

S=solve.QP(Dmat,dvec,Amat,bvec)
cat("The fraction to be invested in each company: ",S$solution,"\n")

## The fraction to be invested in each company:  9.282749e-17 0.05169119 -
3.699079e-17 0 -7.268764e-17 5.055161e-18 -7.882407e-17 8.390146e-17
2.995731e-18 -3.452348e-18 -2.530938e-17 4.116472e-17 0.02485645 -3.136847e-
18 4.60379e-17 0.015273 2.010815e-17 -7.220215e-18 0.139413 -3.898155e-18
0.270172 6.079016e-18 6.107254e-18 0.1255622 0.05833401 -4.010437e-17
0.3146983

new_portfolio = S$solution
cat("Estimated mean of portfolio: ",mean(new_portfolio),"\n")

## Estimated mean of portfolio:  0.03703704

cat("Estimated variance of portfolio: ",var(new_portfolio),"\n")

## Estimated variance of portfolio:  0.006812255

cat("Estimated standard deviation of portfolio: ",sd(new_portfolio),"\n")

## Estimated standard deviation of portfolio:  0.08253638
```

Problem 3

```r
var_selection = read.csv("variable_selection.csv", header = TRUE, sep =",")
model1 = lm(y~x1, data = var_selection)
summary(model1)
```

```
##
## Call:
## lm(formula = y ~ x1, data = var_selection)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.6770  -5.0361  -0.2131   5.5414  20.8261
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   21.395      2.934   7.292 7.95e-11 ***
## x1             2.756      0.922   2.989  0.00354 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.979 on 98 degrees of freedom
## Multiple R-squared:  0.08355,    Adjusted R-squared:  0.0742
## F-statistic: 8.934 on 1 and 98 DF,  p-value: 0.003538
```

```r
model2 = lm(y~x2, data = var_selection)
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ x2, data = var_selection)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.0853 -2.1931 -0.4309  2.3346  7.5392
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4198     0.7537   12.50   <2e-16 ***
## x2            3.9343     0.1339   29.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.995 on 98 degrees of freedom
## Multiple R-squared:  0.8981, Adjusted R-squared:  0.897
## F-statistic: 863.4 on 1 and 98 DF,  p-value: < 2.2e-16
```

```r
model3 = lm(y~x3, data = var_selection)
summary(model3)
```

```
##
## Call:
## lm(formula = y ~ x3, data = var_selection)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -27.9249  -5.9699  -0.2864   6.4416  20.9370 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  32.1473     3.4450   9.332 3.42e-15 ***
## x3           -0.2365     0.3262  -0.725     0.47    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 9.354 on 98 degrees of freedom
## Multiple R-squared:  0.005333,   Adjusted R-squared:  -0.004817 
## F-statistic: 0.5255 on 1 and 98 DF,  p-value: 0.4703
```

```
model4 = lm(y~x1+x2, data = var_selection)
summary(model4)
```

```
## 
## Call:
## lm(formula = y ~ x1 + x2, data = var_selection)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -1.4861 -0.2674  0.0260  0.3658  1.1301 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.15258    0.21048   0.725     0.47    
## x1           2.99924    0.05337  56.195   <2e-16 ***
## x2           3.96917    0.02324 170.781   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.5196 on 97 degrees of freedom
## Multiple R-squared:  0.997,  Adjusted R-squared:  0.9969 
## F-statistic: 1.592e+04 on 2 and 97 DF,  p-value: < 2.2e-16
```

```
model5 = lm(y~x1+x3, data = var_selection)
summary(model5)
```

```
## 
## Call:
## lm(formula = y ~ x1 + x3, data = var_selection)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -24.2130  -5.4023   0.2336   5.6330  21.2464 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  23.6973     4.3676   5.426 4.24e-07 ***
## x1            2.7468     0.9244   2.972  0.00374 **
## x3           -0.2239     0.3139  -0.713  0.47747
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.002 on 97 degrees of freedom
## Multiple R-squared:  0.08833,    Adjusted R-squared:  0.06953
## F-statistic: 4.699 on 2 and 97 DF,  p-value: 0.01128
```

```
model6 = lm(y~x2+x3, data = var_selection)
summary(model6)
```

```
##
## Call:
## lm(formula = y ~ x2 + x3, data = var_selection)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.072 -2.111 -0.457  2.337  7.599
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.71994    1.34887   7.206 1.25e-10 ***
## x2           3.93181    0.13484  29.159  < 2e-16 ***
## x3          -0.02828    0.10516  -0.269    0.789
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.009 on 97 degrees of freedom
## Multiple R-squared:  0.8981, Adjusted R-squared:  0.896
## F-statistic: 427.7 on 2 and 97 DF,  p-value: < 2.2e-16
```

From the above models we see model4 (y~x1+x2) performs best with highest Adjusted R-squared and least error.

Problem 4

Problem 4:

let x_{ij} be current flow through path between node i and node j.

$\therefore$ choose, $x_{12}, x_{13}, x_{23}, x_{24}, x_{34}$

To min $x_{12}^2 \times 1 + x_{13}^2 \times 4 + x_{23}^2 \times 6 + x_{24}^2 \times 12 + x_{34}^2 \times 3$

s.t. $x_{12} + x_{13} = 710$

$x_{24} + x_{34} = 710$

$x_{12} - x_{23} - x_{24} = 0$

$x_{13} + x_{23} - x_{34} = 0$

Solve QP structure:

$$\min \ d^T x - \frac{1}{2} x^T D x$$

$$\text{s.t } A^T x \geq b$$

$d = [0 \ 0 \ 0 \ 0 \ 0]$

$$D = 2 \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 12 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 \\ 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 \end{bmatrix}$$

$b = [710, -710, 710, -710, 0, 0, 0, 0]$

```r
library(quadprog)

Dmat=2*matrix(c(1,0,0,0,0,
                0,4,0,0,0,
                0,0,6,0,0,
                0,0,0,12,0,
                0,0,0,0,3),5,5)
dvec=rep(0,5)
Amat=matrix(c(1,1,0,0,0,
              -1,-1,0,0,0,
              0,0,0,1,1,
              0,0,0,-1,-1,
              1,0,-1,-1,0,
              -1,0,1,1,0,
              0,1,1,0,-1,
              0,-1,-1,0,1),5,8)
bvec=c(710,-710,710,-710,0,0,0,0)


S=solve.QP(Dmat,dvec,Amat,bvec)
cat("Current flowing through each resistor: ",S$solution)

## Current flowing through each resistor:  371.3846 338.6154 163.8462
207.5385 502.4615
```

Problem 5

```r
nfl = read.csv("nflratings.csv", header = FALSE, sep = ',')
nfl$ActualPointSpread = nfl$V4-nfl$V5
func5 <- function(ratings){
  nfl$predicted_spread = ratings[nfl$V2]-ratings[nfl$V3] + ratings[33]
  prediction_error = sum((nfl$ActualPointSpread-nfl$predicted_spread)^2)
  return (prediction_error)
}

random_start = c(rep(85,32),3)
S=optim(random_start,func5,method="CG")

cat("Ratings are: ", S$par[1:32],"\n")

## Ratings are:  84.52235 89.84145 92.74569 83.08899 88.75996 79.81205
87.54406 76.88701 92.12112 85.63577 70.50407 92.25558 86.98432 90.86235
78.43978 76.8882 86.61526 92.06484 96.12267 95.62867 85.09888 93.14842
75.03286 90.95815 86.64234 67.71995 92.60581 85.24194 74.73183 79.17108
82.18828 80.13629

cat("Home advantage: ",S$par[33],"\n")

## Home advantage:  2.172733
```