HW4

Jack Cheng

3/8/2018

Problem 1

```
library(quadprog)

produce_machine = function(L){
  K = (100000 - 12*L)/15
  num_machines = 0.05*L^(2/3)*K^(1/3)
  return(-num_machines)
}
S = optim(8000, produce_machine, method="BFGS")
S
```

```
## $par
## [1] 5556.746
##
## $value
## [1] -204.6684
##
## $counts
## function gradient
##       11       10
##
## $convergence
## [1] 0
##
## $message
## NULL
```

Only with this BFGS solve method can this solution be converged. The maximum amount is 204.

Problem 2

```
stocks = read.csv("~/Desktop/UT Austin/MIS 381N.2 Stochastic control and optimization/homework4stocks.cs
mean_return = colMeans(stocks[,2:ncol(stocks)])
variance = apply(stocks[,2:ncol(stocks)], 2, var)
s = apply(stocks[,2:ncol(stocks)], 2, sd)
corr = cor(stocks[,2:ncol(stocks)], use = "pairwise.complete.obs")
covMat = cov(stocks[,2:ncol(stocks)], use = "pairwise.complete.obs")
# get the variance-covariance matrix
D = 2*covMat
d = rep(0,27)
A = matrix(c(rep(1,27),rep(-1,27),mean_return),27,3)
A=cbind(A, diag(27))
b = c(1,-1,0.01,rep(0,27))
```

```
S=solve.QP(D,d,A,b)

S
```

```
## $solution
##  [1]  2.031752e-17  5.169119e-02 -1.287909e-17  0.000000e+00 -2.836481e-17
##  [6]  1.443921e-17 -5.000485e-17  1.024284e-16 -5.362116e-18 -4.004812e-17
## [11]  6.304565e-17  6.094312e-17  2.485645e-02  1.924800e-18  1.241520e-17
## [16]  1.527300e-02 -7.033633e-18 -2.894296e-18  1.394130e-01  1.583292e-18
## [21]  2.701720e-01  3.340617e-18 -4.571458e-17  1.255622e-01  5.833401e-02
## [26]  5.682633e-18  3.146983e-01
##
## $value
## [1] 0.0008840998
##
## $unconstrained.solution
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##
## $iterations
## [1] 22  0
##
## $Lagrangian
##  [1] 1.187626e-03 0.000000e+00 5.805738e-02 2.620540e-03 0.000000e+00
##  [6] 2.023216e-03 1.113172e-04 1.221323e-03 5.650979e-04 1.499278e-03
## [11] 8.287955e-04 2.604176e-03 7.477740e-04 1.898970e-03 2.990408e-03
## [16] 0.000000e+00 1.609423e-04 2.429636e-03 0.000000e+00 8.800233e-04
## [21] 4.814235e-05 0.000000e+00 2.002272e-04 0.000000e+00 4.438667e-04
## [26] 7.588530e-04 0.000000e+00 0.000000e+00 2.450118e-04 0.000000e+00
##
## $iact
##  [1]  1 14 11 10 15  4  8  3 18  6 13 20 29 26  9 12 25 17 23 21  7
```

```
print(paste('total mean return rate is', sum(S$solution * mean_return)))
```

```
## [1] "total mean return rate is 0.01"
```

```
print(paste('total variance of the  return rate is', S$value))
```

```
## [1] "total variance of the  return rate is 0.000884099846223494"
```

```
print(paste('total standard deviation of the  return rate is', sqrt(S$value)))
```

```
## [1] "total standard deviation of the  return rate is 0.0297338165431802"
```

The solution is as above.

Problem 3

```
VS = read.csv("~/Desktop/UT Austin/MIS 381N.2 Stochastic control and optimization/variable_selection.csv")

lm1 = lm(VS$y ~ VS$x1)
lm2 = lm(VS$y ~ VS$x2)
lm3 = lm(VS$y ~ VS$x3)
lm4 = lm(VS$y ~ VS$x1 + VS$x2)
```

```
lm5 = lm(VS$y ~ VS$x1 + VS$x3)
lm6 = lm(VS$y ~ VS$x2 + VS$x3)

sum(resid(lm1)^2)
```

```
## [1] 7901.299
```

```
sum(resid(lm2)^2)
```

```
## [1] 878.8358
```

```
sum(resid(lm3)^2)
```

```
## [1] 8575.636
```

```
sum(resid(lm4)^2)
```

```
## [1] 26.19087
```

```
sum(resid(lm5)^2)
```

```
## [1] 7860.089
```

```
sum(resid(lm6)^2)
```

```
## [1] 878.1811
```

As we can see, the model that includes x1 and x2 gets the lowest sum of squared residuals and therefore fits the data best.

Problem 4

1)

Choose $x1$, $x3$, $x4$, $x6$, $x12$, that denotes the flows through each resistor.

Min sum$(x_R^2 * R)$

S.t.

$x_1 + x_4 = 710$

$-x_1 + x_6 + x_{12} = 0$

$x_3 - x_4 - x_6 = 0$

```
D = matrix(0,5,5)
D[1,1] = 1
D[2,2] = 3
D[3,3] = 4
D[4,4] = 6
D[5,5] = 12
D=2*D
d = rep(0,5)
A = matrix(c(1,0,1,0,0,-1,0,0,1,1,0,1,-1,-1,0),5,3)
b = c(710,0,0)
S=solve.QP(D,d,A,b)
S
```

```
## $solution
## [1] 371.3846 502.4615 338.6154 163.8462 207.5385
##
## $value
## [1] 2031911
##
## $unconstrained.solution
## [1] 0 0 0 0 0
##
## $iterations
## [1] 4 0
##
## $Lagrangian
## [1] 5723.692 4980.923 3014.769
##
## $iact
## [1] 1 2 3
```

Problem 5

```
NFL = read.csv('~/Desktop/UT Austin/MIS 381N.2 Stochastic control and optimization/nflratings.csv',
               header=FALSE)
NFL$point_spread = NFL$V4 - NFL$V5
func = function(R){
  pred_error = 0
  for (i in 1:nrow(NFL)) {
    pred_error = pred_error + (NFL$point_spread[i] - (R[NFL[i,2]] - R[NFL[i,3]] + R[33]))^2
  }
  return(pred_error)
}
S = optim(rep(0, 33), func, method="BFGS")
solution = S$par[1:32] + (85 - mean(S$par[1:32]))
solution = c(solution, S$par[33])
solution
```

```
##  [1] 84.522358 89.841612 92.745183 83.089265 88.760069 79.812553 87.543497
##  [8] 76.886322 92.120642 85.636517 70.503351 92.255064 86.984740 90.862198
## [15] 78.439387 76.888502 86.615294 92.064543 96.123163 95.629714 85.099246
## [22] 93.148915 75.033602 90.958388 86.641868 67.720162 92.605720 85.241670
## [29] 74.731408 79.170973 82.187631 80.136444  2.172819
```

The ratings are as above. The last scalar is the home team advantage.