CompBio HW3

Fei Xia

Lasso

Loading data:

```r
library(glmnet)
```

```
## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-2
```

```r
library(lars)
```

```
## Loaded lars 1.2
```

```r
dataset<-read.table("prostate.txt")
x <- as.matrix(dataset[,1:8])
y <- as.matrix(dataset[,9])
n=length(dataset[,1])
```

Results of glment:

```r
reg_res <- cv.glmnet(x, y, nfolds=5)
parameters <- coef(reg_res$glmnet.fit, s=reg_res$lambda.1se)
print(parameters)
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                     1
## (Intercept) 0.5173355
## lcavol      0.4616486
## lweight     0.3451527
## age         .
## lbph        .
## svi         0.3939218
## lcp         .
## gleason     .
## pgg45       .
```

My Implementation of Lasso:

```r
# normalization
X <- scale(x)
x_coef <- attr(X,"scaled:scale")
Y <- y-mean(y)

# lasso optimization
```

```r
lambda <- reg_res$lambda.1se
eps = 1e-8
A <- t(X)%*%X
b <- t(X)%*%Y
for (i in 1:8)
  {
    A[i,i]=A[i,i]+lambda
  }
w=solve(A)%*%b

for (step in 1:100)
  {
    w0 <- w
    for (i in 1:8)
      {
        c <- t(X[,i])%*%(Y-X%*%w+w[i]*X[,i])/n
        w[i] <- sign(c)*max((abs(c)-lambda),0)
      }
    if(max(abs(w-w0))<eps)
      {
        break
      }
  }
w=w/x_coef
print(w)
```

```
##                [,1]
## lcavol  0.4557090
## lweight 0.3435818
## age     0.0000000
## lbph    0.0000000
## svi     0.3969634
## lcp     0.0000000
## gleason 0.0000000
## pgg45   0.0000000
```

CompBio HW3

Fei Xia

January 18, 2016

1 Problem 1

1.1 a

For multivariate linear regression model $y = x^T\beta + \epsilon,\ \ \epsilon \sim N(0, \sigma^2)$, prove MLE and LSE is equivalent. **LSE** from $y = x^T\beta + \epsilon,\ \ \epsilon\ N(0, \sigma^2)$ and $\hat{y} = x^T\beta$

MSE is equal to:

$$\sum_i (y_i - \hat{y}_i)^2 = \sum_i (y_i - x_i^T\beta)^2 = (Y - X^T\beta)^T(Y - X^T\beta) \tag{1}$$

with $Y = (y_1, y_2, ..., y_n)^T, X = (x_1, x_2, ..., x_n)^T$

according to LSE formulation:

$$\min_\beta (Y - X^T\beta)^T(Y - X^T\beta) \tag{2}$$

derivative of β is 0,

$$\begin{aligned} \frac{\partial(Y - X^T\beta)^T(Y - X^T\beta)}{\partial\beta} &= -2X(Y - X^T\beta) \\ &= -2XY + 2XX^T\beta = 0 \end{aligned} \tag{3}$$

we get

$$\hat{\beta} = (XX^T)^{-1}XY \tag{4}$$

MLE

because $\epsilon \sim N(0, \sigma^2)$

The likelihood function is:

$$\begin{aligned} L(\beta) &= \prod_i P(y_i|x_i, \beta) \\ &= C_1 e^{C_2 \sum_i (y_i - x_i^T\beta)^2} \\ &= C_1 e^{-C_2(Y - X^T\beta)^T(Y - X^T\beta)} \end{aligned} \tag{5}$$

with $C_1 > 0, C_2 > 0$ being constants.

LL is equal to:

$$\ln L(\beta) = -C(Y - X^T\beta)^T(Y - X^T\beta) \tag{6}$$

with $C > 0$ being a constant

gives the optimization formulation of the problem:

$$\max_\beta -C(Y - X^T\beta)^T(Y - X^T\beta) \tag{7}$$

which is exactly the same as LSE formulation.

1.2 b

In univariate regression model, for one dimension x_j in x, we have $y_j = x_j\beta_j + \epsilon$。

Solve this problem using LSE:

$$\min_{\beta_j} \sum_i (y_{ij} - x_{ij}\beta_j)^2 \tag{8}$$

we get

$$\beta_j^* = \frac{\sum_i x_{ij} y_{ij}}{\sum_i x_{ij}^2} \tag{9}$$

We compare this with the results we got above: $\hat{\beta} = (XX^T)^{-1}XY$.

For the coefficient in multivariate regression $\hat{\beta}$, one dimension $\hat{\beta}_j$ is related to all dimension of data, but for univariate regression, each dimension β_j^* is only related to that corresponding dimension of data.