Assignment-8

Abheek Ghosh

140123047

November 6, 2017

Question 1

Code for R

```r
rm(list = ls())
d = read.table(”FRWRD.txt”, header=FALSE);

cvd_error1 <- function(X, Y, n, r) {
    s = 0;

    for (i in 1:n) {
        X1 = X[-i,];
        Y1 = as.matrix(Y[-i,]);
        # print(X1);
        # print(Y1);
        fn <- function(beta) {
            return (sum((Y-X %*% beta)^2));
        }
        # beta = solve(t(X1) %*% X1) %*% (t(X1) %*% Y1);
        beta = optim(runif(r), fn)$par;
        # print(beta);
        s = s + sum((Y-X %*% beta)^2);
    }
    return (s/n);
}

cvd_error2 <- function(X, Y, n, r) {
    s = 0;
    k = 3;
    m = 2*n;

    for (i in 1:m) {
        leave = -sample.int(n, k);
        X1 = X[leave,];
        Y1 = as.matrix(Y[leave,]);
        # print(X1);
        # print(Y1);
        fn <- function(beta) {
            return (sum((Y-X %*% beta)^2));
        }
        # beta = solve(t(X1) %*% X1) %*% (t(X1) %*% Y1);
        beta = optim(runif(r), fn)$par;
        # print(beta);
        s = s + sum((Y-X %*% beta)^2);
    }
    return (s/m);
}

Y = d[,1];
n = length(Y);
Y = matrix(Y, nrow = n, ncol = 1);
# print(Y);

R = c(3, 6, 8);
mincv1 = numeric();
mincv2 = numeric();

for (r in R) {
    a = numeric();
    for (i in 0:r) {
        a = c(a, (1:n)^i);
    }
    X = matrix(a, nrow = n, ncol = r+1);
    mincv1 = c(mincv1, cvd_error1(X, Y, n, r+1));
    mincv2 = c(mincv2, cvd_error2(X, Y, n, r+1));
    # print(a);
}

print('Error LOOCV then k-fold');
print(mincv1); print(mincv2);
print('degree with minimum error by LOOCV')
print(R[which.min(mincv1)]);

print('degree with minimum error by k-fold')
print(R[which.min(mincv2)]);
```

```
”Error LOOCV then k-fold”
1.248019e+03 2.919813e+13 1.949899e+19
1.606876e+04 1.951479e+13 1.587205e+19
```

”degree with minimum error by LOOCV”

3

”degree with minimum error by k-fold”

3

Question 2

Code for R

```
rm(list = ls())

r = 7;
n = 16;

Y = as.matrix(longley[1:n, r]);
X = as.matrix(longley[1:n, 1:(r-1)]);

X = cbind(rep(1, n), X);

beta = solve(t(X) %*% X) %*% (t(X) %*% Y);
print('Best fit');
print(beta[,1]);

# Outliers
xm = colSums(X)/n;

print('Most Outlier');
print(X[which.max(colSums((t(X) - xm)^2) + (Y - X %*% beta)^2), ])

# Leverage point
print('Most Influential Point');
print(X[which.max((Y - X %*% beta)^2), ])


print('Most Leverage point');
print(X[which.max(colSums((t(X) - xm)^2)), ])
```

”Most Outlier”

GNP.deflator GNP Unemployed Armed.Forces Population Year

1.000 115.700 518.173 480.600 257.200 127.852 1961.000

”Most Influential Point”

GNP.deflator GNP Unemployed Armed.Forces Population Year

1.000 104.600 419.180 282.200 285.700 118.734 1956.000

”Most Leverage point”

GNP.deflator GNP Unemployed Armed.Forces Population Year

1.000 115.700 518.173 480.600 257.200 127.852 1961.000