BLin_Assign5

Bin Lin

2017-3-9

1. Problem Set 1

In this problem set we'll work out some properties of the least squares solution that we reviewed in the weekly readings. Consider the unsolvable system Ax = b as given below:

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 3 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 8 \\ 8 \\ 20 \end{pmatrix}$$

Your code should be able to print all of the above requested quantities. Please include enough comments to make it easy to follow your R markdown document.

Write R markdown script to compute ATA and ATb.

```
A <- matrix(c(1, 0, 1, 1, 1, 3, 1, 4), 4, 2, byrow = TRUE)
A
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    1    1
## [3,]    1    3
## [4,]    1    4
```

```
b <- matrix(c(0, 8, 8, 20), 4, 1, byrow = TRUE)
b
```

```
##      [,1]
## [1,]    0
## [2,]    8
## [3,]    8
## [4,]   20
```

```
ATA <- t(A) %*% A
ATA
```

```
##      [,1] [,2]
## [1,]    4    8
## [2,]    8   26
```

```
ATb <- t(A) %*% b
ATb
```

```
##      [,1]
## [1,]   36
## [2,]  112
```

Solve for ^x in R using the above two computed matrices. According equation $A^T A\hat{x} = A^T b$

```
x_hat <- solve(ATA, ATb)
x_hat
```

```
##      [,1]
## [1,]    1
## [2,]    4
```

What is the squared error of this solution? According to the formula: $\|Ax - b\|^2 \quad = \quad \|Ax - p\|^2 + \|e\|^2$

```
p <- A %*% x_hat
squared_error <- (A %*% x_hat - b)^2 - (A %*% x_hat - p)^2
sum(squared_error)
```

```
## [1] 44
```

Instead of b = [0; 8; 8; 20], start with p = [1; 5; 13; 17] and find the exact solution (i.e. show that this system is solvable as all equations are consistent with each other. This should result in an error vector e = 0).

```
# Replace b with p, then  repeat the same process. The result is vector x which can make the pro
jection (Ax) closest to b.

p <- matrix(c(1, 5, 13, 17), 4, 1, byrow = TRUE)
ATb <- t(A) %*% p
x <- solve(ATA, ATb)
x
```

```
##      [,1]
## [1,]    1
## [2,]    4
```

```
#Based on the formular: e = b - Ax. Same idea here, replace b with p to solve for e.
e <- p - A %*% x
e
```

```
##      [,1]
## [1,]    0
## [2,]    0
## [3,]    0
## [4,]    0
```

Show that the error e = b - p = [-1, 3, -5, 3]

```
e <- b - p
e
```

```
##      [,1]
## [1,]   -1
## [2,]    3
## [3,]   -5
## [4,]    3
```

Show that the error e is orthogonal to p and to each of the columns of A.

```
t(e) %*% p
```

```
##      [,1]
## [1,]    0
```

```
for (i in 1:ncol(A))
{
  print (t(e) %*% A[,i])
  }
```

```
##      [,1]
## [1,]    0
##      [,1]
## [1,]    0
```

2. Problem Set 2

Consider the modified auto-mpg data (obtained from the UC Irvine Machine Learning dataset). This dataset contains 5 columns: displacement, horsepower, weight, acceleration, mpg. We are going to model mpg as a function of the other four variables.

Write an R markdown script that takes in the auto-mpg data, extracts an A matrix from the first 4 columns and b vector from the fifth (mpg) column. Using the least squares approach, your code should compute the best fitting solution. That is, find the best fitting equation that expresses mpg in terms of the other 4 variables. Finally, calculate the fitting error between the predicted mpg of your model and the actual mpg. Your script should be able to load in the 5 column data set, extract A and b, and perform the rest of the calculations. Please have adequate comments in your code to make it easy to follow your work.

Please complete both problem set 1 & problem set 2 in one R markdown document and upload it to the site. You don't have to attach the auto-mpg data. Just write your markdown document in such a way that it expects and loads the auto-mpg data file from the current working directory. As always, your code is expected to compile and run successfully. Adding test cases to demonstrate that your code is working will be very helpful.

First step is loading the data, change the column names, then extracts A matrix and b vector.

```
raw_data <- read.table("C:/Users/blin261/Desktop/DATA605/assign5/auto-mpg.data")
colnames(raw_data) <- c("displacement", "horsepower", "weight", "acceleration", "mpg")
head(raw_data)
```

```
##   displacement horsepower weight acceleration mpg
## 1          307        130   3504         12.0  18
## 2          350        165   3693         11.5  15
## 3          318        150   3436         11.0  18
## 4          304        150   3433         12.0  16
## 5          302        140   3449         10.5  17
## 6          429        198   4341         10.0  15
```

```
A <- as.matrix(raw_data[,1:4])
b <- as.matrix(raw_data[,5])
```

Secondly, I am using the same step as presented in problem set 1 to calculate for vector x_hat. When x = x_hat, the squared length of Ax - b is minimized.

```
ATA <- t(A) %*% A
ATb <- t(A) %*% b
x_hat <- solve(ATA, ATb)
x_hat
```

```
##                      [,1]
## displacement -0.030037938
## horsepower    0.157115685
## weight       -0.006217883
## acceleration  1.997320955
```

Therefore, the best fitting equation that expresses mpg in terms of the other 4 variables is:

mpg = -0.030(displacement) + 0.157(horsepower) - 0.006(weight) + 1.997(acceleration) + e

Fitting error between the predicted mpg of your model and the actual mpg.

```
p <- A %*% x_hat
e <- b - p
head(e)
```

```
##           [,1]
## [1,]  4.616218
## [2,] -0.417359
## [3,]  3.378827
## [4,] -1.057679
## [5,]  4.548870
## [6,]  3.795990
```

```
elength <- sqrt(t(e) %*% e)
elength
```

```
##          [,1]
## [1,] 114.4615
```

```
#Second way of getting fitting error.
#squared_error <- (A %*% x_hat - b)^2 - (A %*% x_hat - p)^2
#sqrt(sum(squared_error))
```