Statistics 5014: Homework 6

Due In Class October 11, 9am

2017-10-08

For each assignment, turn in by the due date/time. Late assignments must be arranged prior to submission. In every case, assignments are to be typed neatly using proper English in Markdown.

This week, we spoke about R's dual handling of vectors and matrices. In this homework, we will use this duality to our advantage to both simplify our code and perhaps speed up computation.

Problem 1

Not a swirl problem. :)

New this week, please create a new R *Notebook* file within the project folder within the "06_vector_matrix_dual_math_speed" subfolder (file–>new–>R Notebook–>save as. This time we will knit to html.

The filename should be: HW6_lastname_firstname, i.e. for me it would be HW6_Settlage_Bob

You will use this new R Notebook file to solve the following problems:

Problem 2: Sums of Squares

One basic and recurring theme you will hear in statistics is "sums of squares". Sums of squares error, regression, total, . . .

In this problem, we will calculate sums of squares total using:

a. a for loop to iterate through all data points calculating the summed squared difference between the data points and mean of the data.
b. repeat part a, but use vector operations to effect the same computation

In both cases, wrap the code in "system.time({})". You should report the final answer and timings for both a and b.

To generate the data, use:

```
set.seed(12345)
y <- seq(from = 1, to = 100, length.out = 1e+08) + rnorm(1e+08)
```

Problem 3: Using the dual nature to our advantage

As above, sometimes using a mixture of true matrix math plus component operations cleans up our code giving better readibility. Suppose we wanted to form the following computation:

- $while(abs(\Theta_0^i - \Theta_0^{i-1}) \text{ AND } abs(\Theta_1^i - \Theta_1^{i-1}) > tolerance)$ {

$$\begin{aligned}
\Theta_0^i &= \Theta_0^{i-1} - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_0(x_i) - y_i) \\
\Theta_1^i &= \Theta_1^{i-1} - \alpha\frac{1}{m}\sum_{i=1}^{m}((h_0(x_i) - y_i)x_i)
\end{aligned}$$

}

Where $h_0(x) = \Theta_0 + \Theta_1 x$.

Given $\mathbf{X}$ and $\vec{h}$ below, implement the above algorithm and compare the results with lm(h~0+**X**). State the tolerance used and the step size, α.

```
set.seed(1256)
theta <- as.matrix(c(1, 2), nrow = 2)
X <- cbind(1, rep(1:10, 10))
h <- X %*% theta + rnorm(100, 0, 0.2)
```

Problem 4: Inverting matrices

Ok, so John Cook makes some good points, but if you want to do:

$$\hat{\beta} = (X'X)^{-1}X'\underline{y}$$

what are you to do?? Can you explain what is going on?

Problem 5: Need for speed

In this problem, we are looking to compute the following:

$$y = p + AB^{-1}(q - r) \tag{1}$$

Where A, B, p, q and r are formed by:

```
set.seed(12456)

G <- matrix(sample(c(0, 0.5, 1), size = 16000, replace = T),
    ncol = 10)
R <- cor(G)  # R: 10 * 10 correlation matrix of G
C <- kronecker(R, diag(1600))  # C is a 16000 * 16000 block diagonal matrix
id <- sample(1:16000, size = 932, replace = F)
q <- sample(c(0, 0.5, 1), size = 15068, replace = T)  # vector of length 15068
A <- C[id, -id]  # matrix of dimension 932 * 15068
B <- C[-id, -id]  # matrix of dimension 15068 * 15068
p <- runif(932, 0, 1)
r <- runif(15068, 0, 1)
C <- NULL  #save some memory space
```

Part a.

How large (bytes) are A and B? Without any optimization tricks, how long does the it take to calculate y?

Part b.

How would you break apart this compute, i.e., what order of operations would make sense? Are there any mathmatical simplifications you can make? Is there anything about the vectors or matrices we might take advantage of?

Part c.

Use ANY means (ANY package, ANY trick, etc) necessary to compute the above, fast. This compute will be what we will use in our fastest compute challenge next week. Wrap your code in "system.time({})", everything you do past assignment "C <- NULL".

Problem 6

Push your homework and submit a pull request.

When it is time to submit, –ONLY– submit the .Rmd and .nb.html solution files. Names should be formatted HW#_lastname_firstname.Rmd

Optional preperation for next class:

Next week we will talk about parallel computing in R. We will focus on MPI and parfor.