SDS 385 – Big Data
Oct 13, 2016
Peer Review 2

Reviewer: Jennifer Starling
Reviewing: Matteo Vestrucci

Topic: Exercise 4, Adagrad for large sparse matrices, Rcpp implementation.

General Overview of Code:

It was very interesting and helpful to see your implementation of the Adagrad algorithm for sparse matrices in R. First, your 42-second run time was extremely impressive, given that most of us could not get Adagrad to work for the large sparse matrix in R without running out of memory.

I thought your solution of saving the matrices in the sparse row format, with three vectors (i), (j) and (k), was well done. This was a unique solution; I did not see anybody else try this, and it was a good way to tie in some of our very first exercises where we discussed sparse matrix storage.

Rcpp Implementation:
We spent a good amount of time working through translating your algorithm to Rcpp, and your code seemed to be quite far along when we wrapped up. (I know there were one or two errors in execution you were still working through, but overall you seemed to be in very good shape.) I think your solution translated well to Rcpp, and it was very good to see a different way to think about the problem. (I implemented the same type of sparse iterator solution as most of the class did, where the Rcpp function intakes the whole matrix, and not just the three representative columns.)

Suggestions:
I would suggest that once you have ironed out any last Rcpp implementation issues, that you add some commenting in your code to explain your solution, since it is a very good one. I think if I were not familiar with the problem after discussing with you in person, that it might be difficult to see the intuition of your solution from reading the code alone.

You could take advantage of some of the Armadillo or Eigen linear matrix operations. I do not suspect they are really any faster than the for loops we implemented in pure Rcpp, but in terms of code readability, they do make it more immediately obvious which linear algebra operation is occurring. This would be very easy to do with Armadillo, since we did include the Armadillo headers in the Rcpp file.

Overall, you have done excellent work on this. Thanks for sharing and walking me through your solution!