Peer Review 1

Reviewer: Yanxin Li
Reviewing: Natalia Zuniga-Garcia
Topic: EX01: Weighted Linear Regression & Generalized Linear Models

In this review I will provide my overall impressions of comments on both your written report and the R code you provided on your GitHub.

1. Comments on written report (solution01-SDS385.pdf)

In each part, you included the questions and then provided the answers. This is readable and complete for a beginner who would like to learn Big Data from your GitHub. Also, your GitHub is well-organized with "README" describing every folder and file generally. The following will be mainly on your solutions.

1.1 Topic "Weighted Linear Regressions"

For Topic "Linear Regression" part (A), I would recommend including some basic properties for matrix derivatives as Prof. James Scott suggested in the class. Much more details make it more efficiently understandable.

For pseudo-code of part (B), it is very clear but it would be much better to explain why you choose Cholesky methods instead of QR decomposition and SVD (Singular Value Decomposition) methods.

For part (C), you included part of your R code in the solution. A table frame is recommended to have the explanatory code in, which makes the report look nicely. I like the methods you used for comparing efficiency of inverse method and "my methods"—two figures. This data visualization helps explicitly conclude which methods is better.

In Part (D) you searched *Matrix* and *Slam Packages* and used *Matrix* for your implementation. This is helpful for all of us to search different methods to achieve one goal. We do learn a lot during this process. The figures showed a full comparison of computation time with different sparsity, N and P. In my solution for both part (C) and (D), there is no figure for comparison with a simple data output. It is not fairly self-explanatory. I would definitely improve my code by referring to your coding.

1.2 Topic "Generalized Linear Models"

For Topic "Generalized Linear Models" part (A) and (C), you did a great job. I actually referred to your methods in part (C) for completing the square from a quadratic form. Here is the part I learned from you and make it not messy at all:

$$f(x) = c + bx + \frac{1}{2}x^T ax = \frac{1}{2}(x^T ax + 2bx + c) = \frac{1}{2}(x-u)^T a(x-u)$$

$$= \frac{1}{2}(x^T ax - 2u^T ax + u^T au)$$

$$\Rightarrow b = -u^T a, c = \frac{1}{2}u^T au \Rightarrow u = -(ba^{-1})^T, c = \frac{1}{2}b(a^{-1})^T b^T$$

For part (B) and part (D), it clearly described the gradient descent algorithm, Newton's methods and the convergence criteria, which is critical for explaining R code. The results showed in the figures and tables are explicitly self-explanatory. The summary in part (E) analyzed both advantages of disadvantages of Newton's methods, which is fully considered.

1.3 What I learn from your written report

It is well-organized and you solutions for every problem is perfect explained. You searched many literatures for every topic and shared the in your GitHub. It is helpful for me to improve my knowledge system and understand deeply for every topic.

2. Comments on R code (file Ex01R)

Overall, your R code looks very clean and organized. The spacing is done well so that it is easy to read. You include plenty of comments so that a reader can get through your code more efficiently.

You sometimes put a lot of commands in one line. For example, in P1C2Simulation.R line 14 and line 15,

```
m1 = mean(unlist(microbenchmark(inv.method(X, y, W), times=5L), use.names = FALSE)
[6:10])/10^6
m2 = mean(unlist(microbenchmark(my.method(X, y, W), times=5L), use.names = FALSE)
[6:10])/10^6
```

I'm not sure, but it may be more computationally efficient to break this off into two lines. Your plotting has the same problems. Of course, sometimes it happens to me and I am currently changing my coding habit gradually.

There are some general tips I need learn from you on the best ways to perform matrix multiplication in R. For example, crossprod(X,y) is faster than t(X) %*% y, and A %*% (B %*% y) is faster than A %*% B %*% y. It is better to separate functions and simulations, and leave more space for each section (functions, loops, results, and plots), making it readable and understandable. "=" and "<-" should be uniform to keep consistency.

I like the cleanness in how you define your functions. For example, you define a

sigmoid function to make calculations of wi easier later on (This is exercise02. It looks very nice because it makes your subsequent functions less cluttered. It's a good application of function-oriented programming. When I ran your code and went through the iterations of a for loop and updating a list for each iteration, it is more efficient to define a list before loop of NA values with the same length as the number of iterations, and then changing the ith entry in the list during the ith iteration of the loop. You did great for this.

3. Conclusion

I hope this review has been helpful. Since there are some incompleteness and inefficient commands in my written report and R code, it is like a tutorial when reading your solutions and going through your R code. As for algorithm and complementation, basically, they looks perfect. I hope to discuss with you about the following exercises and share what is good for us to improve. As for exercise02, I am working on it now and still need some more perfection, so I would be happy to review that later as well when I fully understand the Stochastic Gradient Descent Algorithm.