IE 510 project milestone

Team number: Muxia Yi, Tianqi Wu

Overview of the Proposal

In this project, we aim to study an application of Kaggle competition. Dealing with real world problem is pretty challenging since the dataset from Kaggle competition is taken from real-world including number of samples with various features. Also, we have to focus more on the efficiency because it is important for practical problems. The project will focus on finding a most suitable algorithm to provide predictions by applying several methods learned from the class (i.e, gradient descent, heavy ball, Nesterov, coordinate descent and stochastic descent).

Background and Motivation

This project will implement and compare different algorithms to solve real world problems provided by Kaggle. The topic is of interest to the technical community since it would validate the theoretical results derived from the algorithms. We are particularly interested in since it would give us a perfect chance to apply what we have learned from lectures and enhance our understanding. Potential problems worth further investigating includes trying different variants of the algorithms with various stepsize tuning.

Proposed Project

After carefully examining several datasets, the dataset decided to be study is "Breast Cancer Wisconsin Data Set"[1]. Logistic model would be used to fit the data and optimization of the algorithms is then carries out. Finally, this project will compare different behaviors of the optimization algorithms and conclude the best one. We believe the project can be completed by the end of the semester since some groups have already used the logistic regression model to predict the dataset and we can focus more on the optimization part. The algorithms to be studied were thoroughly discussed in class and homework.

Timeline

4/7: Clean the data
4/14: Fit the data using logistic regression
4/22: Implement the GD, HB
4/29: Implement CD method
5/6: Implement SGD method.
5/13: Compare the results and finish report

---

[1] https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/data

Expected Results
We aim to understand the behavior of the algorithms better given different stepsize and variants. Our goal it to Compare the algorithms considering the complexity, convergence speed and computation time and conclude the best one.

Initial Progress
After making the proposal of the tentative topic, our group has examined several datasets from Kaggle and we have decided the suitable dataset to be studied.

Milestone
After carefully examining nearly 20 datasets on Kaggle, we successfully found the right dataset. So far, we have cleaned the data, applied the logistic regression and implemented GD, HB methods.

To clean the data, we examined the dataset and did some preliminary analysis of the features. First, we found that there is no missing value in the dataset. Then, we standardized the data by removing the mean and scaling to the unit variance.

Furthermore, we removed some of the features having correlation below than 0.02 to the response. Then, we start implementing the logistic regression with different methods.

Consider the following logistic regression problem:

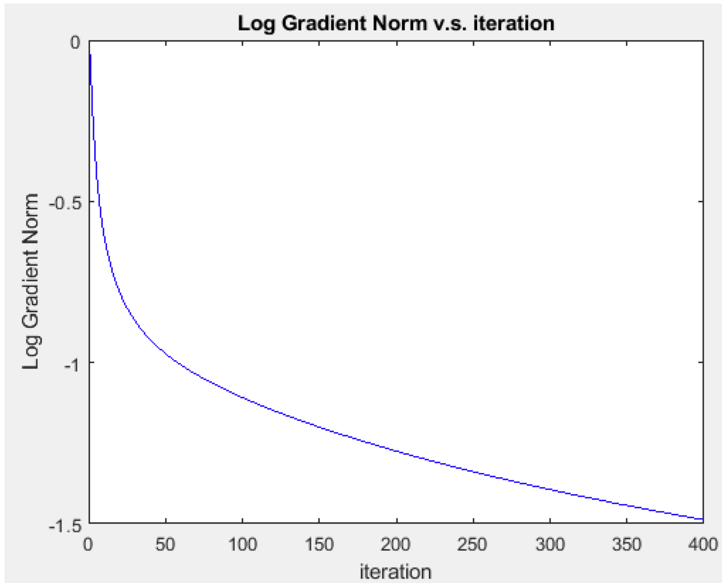
$$\min_{w \in \mathbb{R}^d} f(w) = \frac{1}{n} \Sigma_{i=1}^{n} log(1 + exp(-y_i w^T x_i))$$

After removing four attributes with low correlation, we have 569 samples with 26 features (n = 569, d = 26). First, we started with small sample of 12 samples and 5 features and implemented the gradient descent.

For parameter tuning, we tried Bayesian optimization with baseline of 1/L and we found it was very slowly. Then, we decided to manually operate parameter with constant stepsize of 1/L, 2/L, 3/L and the below graphs show the result of 1/L.

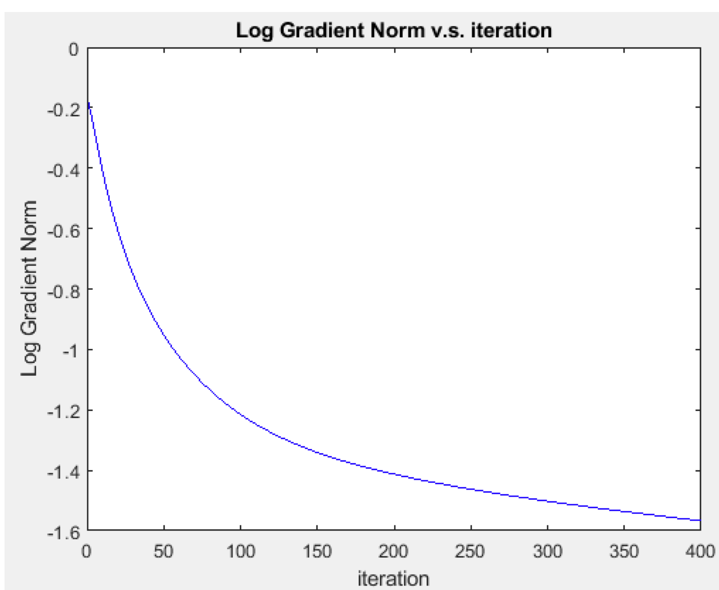
GD:

n = 12, d = 5



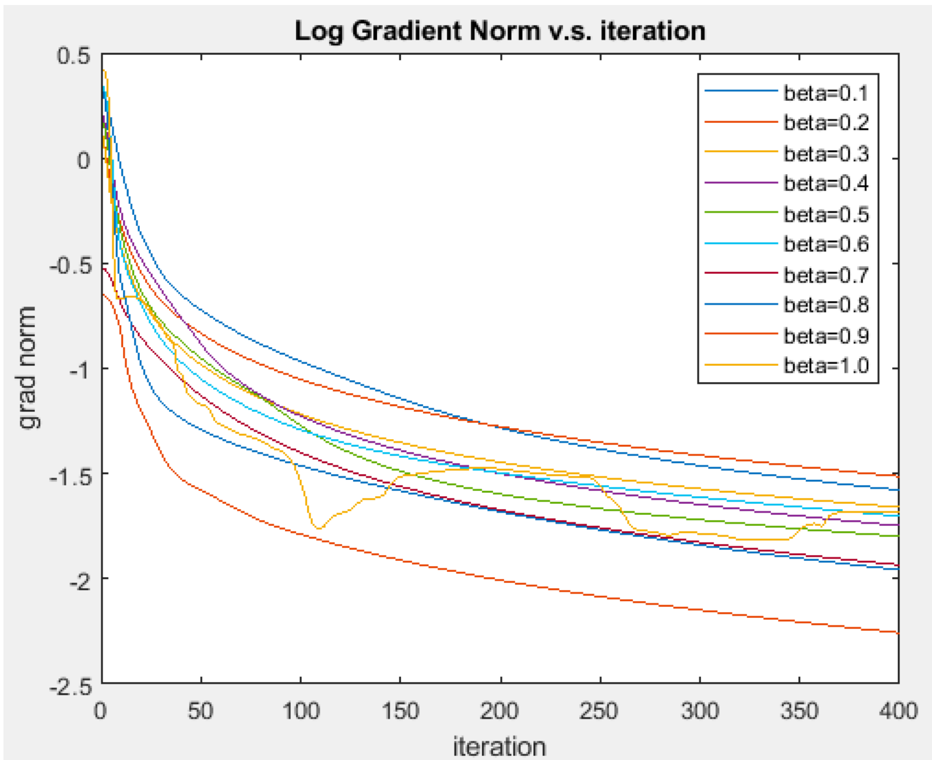
Then, we used the whole dataset to run the gradient descent. Similar behaviors can be seen for small and large samples that the data setting is separable since the gradient norm arrives at flat region.

n = 569, d = 26



HB:

Then, we used the whole dataset to run the heavy ball method with parameter tuning. It can be seen the best beta here is 0.9 and the convergence speed is faster than the gradient descent.



Next step, we will implement coordinate descent and stochastic gradient descent. Hopefully, we can try different variants of the methods and get a better understanding of the behaviors of the algorithms.