IE 510 Course Project
Final report

University of Illinois at Urbana-Champaign

Introduction

1.1 Overview of the project

In order to understand the behavior of machine learning optimized algorithms better, this project studied an application of Kaggle competition by several algorithms with different step-size and variants. Dealing with real world problem was pretty challenging since the dataset from Kaggle competition was taken from real-world including a large number of samples with various features. Also, the efficiency was important for practical problems. So the algorithms applied in machine learning were the key factor in deciding success.

In order to have a comprehensive perspective about the algorithms we learned from lecture, to find whether a new algorithm would lead to a better convergence speed while conventional machine learning packages such as support vector machine work well, this project implemented and compared different algorithms to solve real world problems provided by Kaggle. The project found the most suitable algorithm to provide predictions by applying several methods learned from the class, namely, gradient descent method, gradient descent with momentum(including heavy ball method and Nesterov's method), decomposition-type methods(stochastic gradient descent) and quasi-Newton algorithm(BFGS method) and BB (Barzilai-Borwein) Method (including LBB method and SBB method). In additional, Our project not only contains experiments with real world dataset, but also contains experiments with artificial data with gaussian distribution as comparative tests.

After comparing the algorithms considering the complexity, convergence speed and computation time, we found that for medical prediction problems with logistic regression model, Barzilai-Borwein method was fastest, then heavy ball method with parameter beta 0.9 was the second.

1.2 literature background and Motivation

In practice, all engineering design should include optimization phases. Nowadays, with the development of artificial intelligence, machine learning is becoming one of the hottest topics since machine learning is a method to prediction [1]. Because algorithms in machine learning are the key factor which decides the efficiency and results, we are particularly interested in how different algorithms will contribute to a real world problem for prediction.

Traditional machine learning algorithms in classification such as support vector machines (SVM) [2] had been well studied. Hirji, Karim, Anastasios, Tsiatis and Cyrus (1989) had studied how to deal with binary data in 1989, which provided a meaning theory for prediction. At present, people are more likely to take advantage of gradient or momentum to logistic regression problems in practice, though Allison (2008) pointed out that traditional optimization methods may result in failure in logistic regression problem. In addition, Heinze and Schemper (2002) have shown that penalized maximum likelihood estimation always yields finite estimates of parameters under complete or quasi-complete separation.

This topic was very interesting because it would validate the theoretical results derived from the algorithms. What was the difference between theory and practice? Can we find an algorithm which had a better performance in logistic regression predictions? In order to find out whether a new algorithm would lead to a better convergence speed while conventional machine learning packages seem to work well, this project implemented and compared different algorithms with a series of experiments.

1.3 Problem definition

Predictions were frequently happened in medical field, and the accuracy of prediction was particularly significant. So our project would take into prediction problems. Logistic model would be used to fit the data and optimization of the algorithms was then carries out. Mathematical model of the problem is as follow:

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

1.4 Timeline

4/7:	Clean the data
4/14:	Fit the data using logistic regression
4/22:	Implement the GD, HB, Nesterov’s
4/29:	Decomposition-type methods (stochastic gradient descent)
5/3:	Implement BB (LBB and SBB) method
5/9:	Implement BFGS method.
5/13:	Compare the results and finish report

Methodology

Methodology of project was mainly divided into three parts: data description, implementation of algorithms, experimentation and results.

2.1 Data description

After carefully examining 20 datasets, the dataset decided to be study was “Breast Cancer Wisconsin Data Set” [3], which was a classic problem in this area. Our data was download from Kaggle[3].

To clean the data, we examined the dataset and did some preliminary analysis of the features. First, we found that there was 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, the dataset was ready for implementing the logistic regression with different methods.

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 as a preprocessing.

2.2 Implementation of algorithms

Parameter tuning was an important part of exploring the different convergence behavior. For parameter tuning, we tried Bayesian optimization with baseline of 1/L and 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, and L is the max eigenvalue of A, and A=X’*X.

Analysis of convergence behavior in terms of convergence speed, efficiency is the main part of this project. Algorithms for providing predictions are including, gradient descent method, gradient descent with momentum (including heavy ball method and Nesterov’s method), decomposition-type methods(stochastic gradient descent) and quasi-Newton algorithm(BFGS method) and BB (Barzilai-Borwein) Method (including LBB method and SBB method).

Gradient descent (GD) uses following recursion:

$$w_{i+1} = w_i - \alpha \nabla f(w_i)$$

Where α= 1/L, L is the max eigenvalue of A, and A=X’*X.

Heavy ball method and Nesterov’s method:

HB:

$W^{k+1} = w^k - \alpha\nabla f(w^{k)} + \text{beta}*(w^k - w^{k-1})$;

Nesterov:

$$y^r = x^r + \beta_r(x^r - x^{r-1}), \quad \text{slip due to momentum}$$
$$x^{r+1} = y^r - \alpha\nabla f(y^r). \quad \text{move along gradient}$$

Stochastic gradient descent method (SGD):

After applying Gradient descent method and momentum method (HB and Nesterov's), we continues to study decomposition method. Since in our database, n=569, d=26, n is much larger than d, so we choose SGD method in the project. The algorithm is:

$w^{t+1}=w^t-\alpha^t\nabla f(w^t)$

Quasi-Newton method(BFGS):

The algorithm is:

For k=0,1,2….. n,

initial w_0 and B_0

Obtain a direction pk by $B^k*p^k= -\nabla f(w^k)$; $w^{k+1}=w^k +\alpha^k p^k$; update B^k to B^{k+1}

end.

Barzilai-Borwein(BB) method:

Smilar to BFGS, BB method contains LBB and SBB. The algorithm of BB is:

$w^{k+1}=w^k-H^k*\nabla f(w^k)$, where $H^k=I*\alpha k$ I denotes the identity matrix.

2.3 Experimentation and results

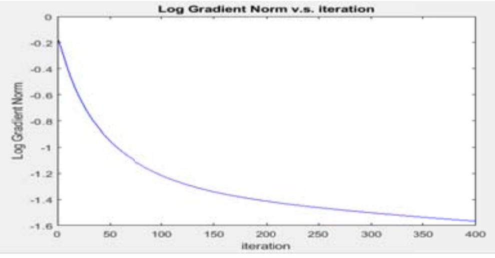
Our project not only contains experiments with real world dataset, but also contains experiments with artificial data with Gaussian distribution as comparative tests.

All code for experiments see attach. They were written in Maltab. We also had some code in python.

2.3.1 Experiments with real world data

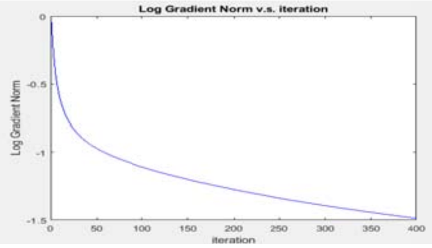
GD method

First, we started with small sample of 12 samples and 5 features and implemented the gradient descent. 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. Results see figures below.



n = 12, d = 5

Figure 1



n = 569, d = 26

Figure 2

HB method

In the second experiment 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. The results showed in figure 3.

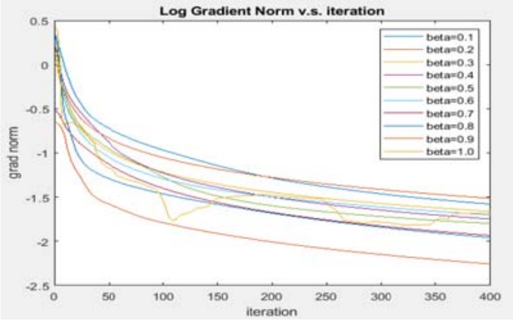


Figure 3

Nesterov's method

Then we did the nesterov's experiment. The result showed as follow. Since we had found when beta was 0.9, heavy ball had the best performance. So In this experiment, we first tune beta to 0.9, then, 1.0, 1.1, 0.8,0.7 manually. Similar to HB, when beta was 0.9 had the best performance. Below was the picture of beta =0.9.

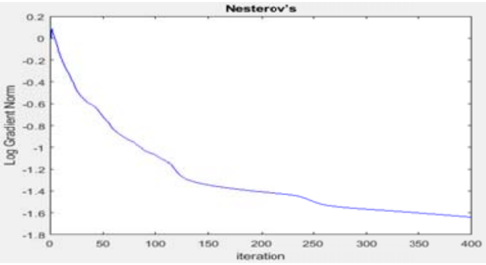


Figure 4

SGD

Our database, n=569, d=26, n is much larger than d, so we chose SGD method in the project, the result of SGD was as below.

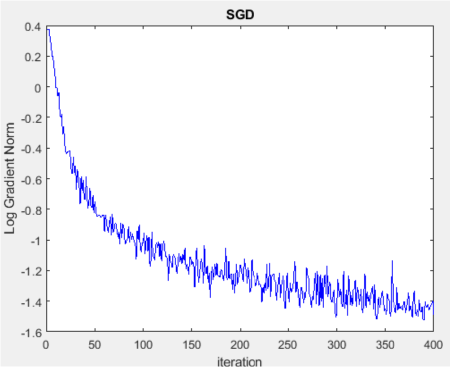


Figure 5

LBB, SBB and BFGS:

Next step, we implemented and compared LBB, SBB and BFGS. We plot the gradient norm (log scale) vs. iterates. LBB and SBB were showed as figures 6 and 7. Figures 8 showed the results of BFGS.

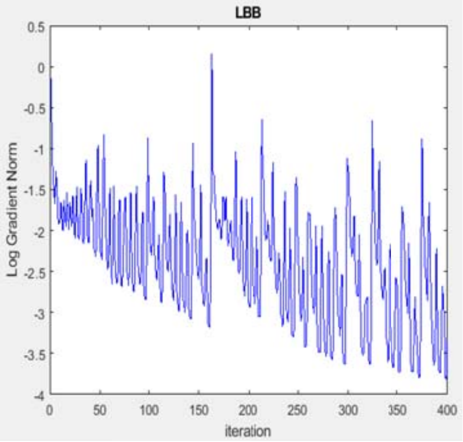


Figure 6

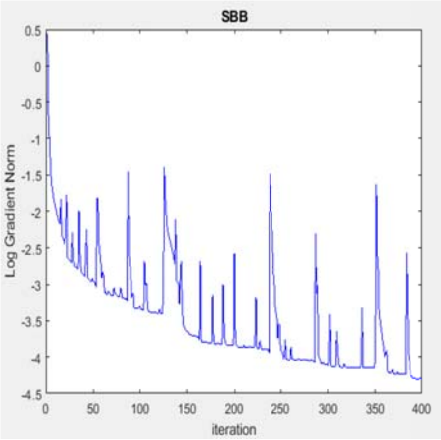


Figure 7

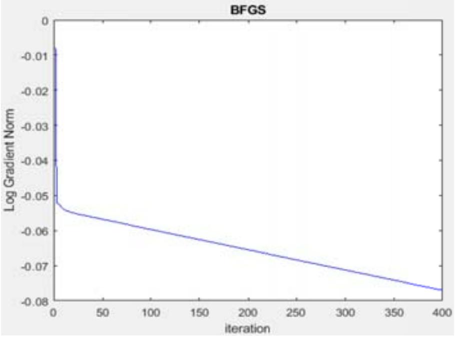


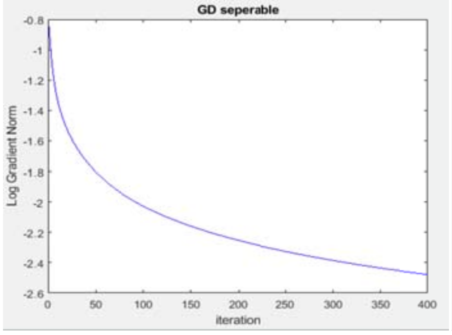
Figure 8

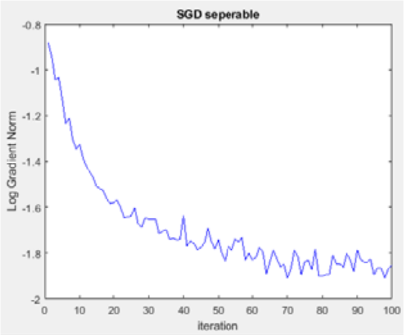
2.3.2 Experiments with Artificial Data

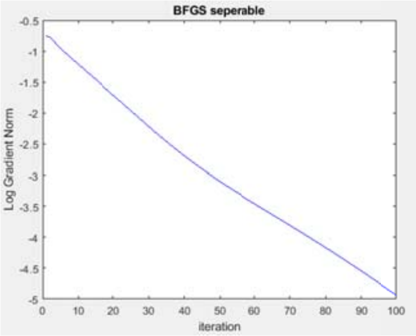
Then, we also compared different behaviors of the algorithms given separable artificial data with gaussian distribution of following setting:

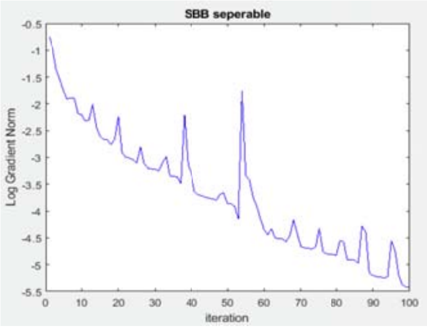
Define $w^* = [1;1;\ldots;1] \in \mathbb{R}^{dX1}$, and generate label $y_i = sign(x_i^T w^*), \forall i$. Here, $sign(z) = 1$ if $z \geq 0$ and $sign(z) = -1$ if $z < 0$.

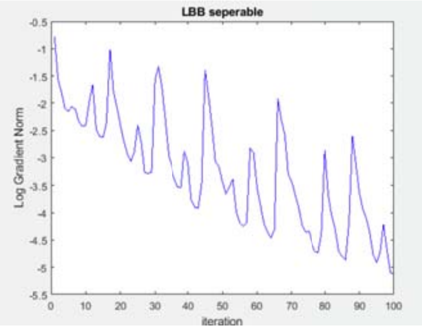
All settings are consistent with previous experiments. Results with artificial data as follow.











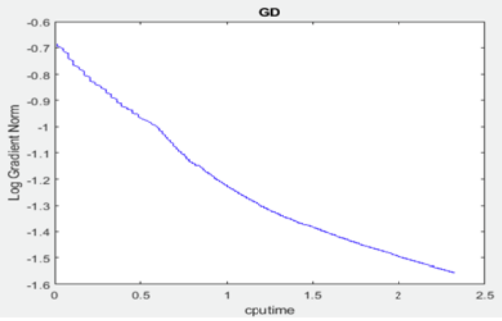
Analysis & Results

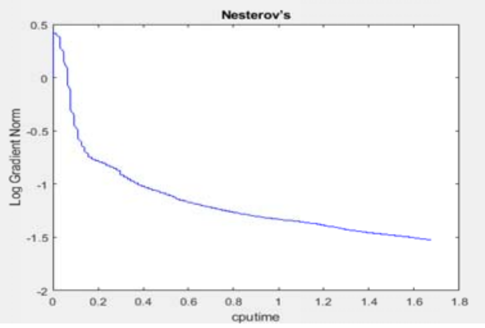
The problem was taken from real-world dataset and it has relatively high dimensions. We first tried solving it using logistic regression from sklearn machine learning package from python and it reaches 82% accuracy after data preprocess. Then, we tried small dataset and compared the result with large dataset. It shows that there is no large difference between the two.

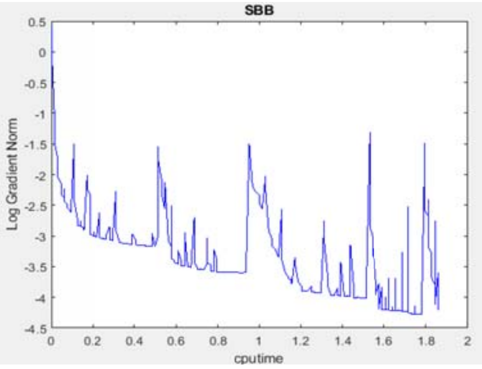
Hence, we are confident solving the problem with logistic regression. According to the figures above, it is a separable case and w diverges which arrives at flat region. Since the iterates diverge and it does affect the convergence rate a lot. After 400 iterates, the log gradient norm is only about -1.5. BFGS, however, happens to be the slowest among all the algorithms since the question is not strongly convex and we have little theory for this case. LBB with gradient norm in log level and SBB with -3.5 are fastest since they use the optimal stepsize for GD while they both have severe oscillation along the curve. HB(-2) with beta=0.9 is faster than GD and Nesterov's (-1.6).

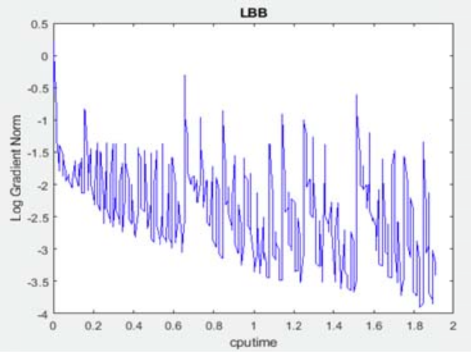
Afterwards, we also tried artificial data with Gaussian distribution and it turns out that BFGS is fastest with no vibration compared with SBB and LBB. The situation for this case differs from the real data that the convergence rate of BFGS is reversed.

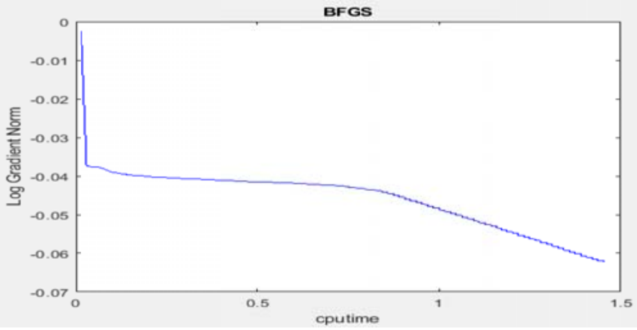
CPU time which indicated the storage and speed of algorithm also show the same performance with the iteration.











Conclusion

This project presents the study of convergence behavior discovery and evaluation of several different kinds of algorithms for medical prediction problems.

Unlike previous works which either only focus on gradient descent method or only focus on newly method, this project applies GD, heavy ball method and Nesterov’s method, stochastic descent and quasi-Newton algorithm and Barzilai-Borwein Method. As a result, this project has a comprehensive analysis of algorithms for logistic regression prediction.

The results of our project shows that constant step size, and classic routines for momentum seem to be robust, this result can explain why nearly all machine learning packages prefer to applying GD or momentum algorithm when handle with logistic regression problem. However, the project found that Barzilai-Borwein(BB) method have a better performance than momentum when considering the convergence speed.

The project also found an interesting phenomenon that in BFGS experiments, results differs when using the real data and artificial data. For better result of the solving of this problem, we may introduce regularizer to resolve the issues in separable case.

As regards to future directions, further investigating of this area by also considering to turn more parameters and by trying more advanced method will be an interesting topic.

Reference

[1] Machine learning. Retrieved from: https://en.wikipedia.org/wiki/Machine_learning

[2] Vapnik, V. (1995). The natural of statistical Learning Theory. Springer, New York.

[3]dataset, available at：https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/data

[4] Allison 2008, Convergence Failures in Logistic Regression. *SAS Global Forum* .

[5]Heinze, George and Michael Schemper (2002) “A Solution to the Problem of Separation in Logistic Regression.” *Statistics in Medicine 21*: 2409-2419.

[6]Hirji, Karim F., Anastasios A. Tsiatis and Cyrus R. Mehta (1989) “Median Unbiased Estimation for Binary Data.” *The American Statistician* 43: 7-11