IE 510 Course Project

Final report

Muxia Yi,   Tianqi Wu

University of Illinois at Urbana-Champaign

Introduction

1.1 Overview of the project

In this project, we aim to study a dataset from Kaggle competition and better understand the behavior of several optimization algorithms given different step-size and variants. Dealing with real world problems is pretty challenging since the dataset from Kaggle competition is taken from real world including number of samples with various features. Also, the efficiency is important for practical problems. Hence, the algorithms applied in machine learning are the key factor in success.

This project implements and compares different algorithms to have a comprehensive perspective about the algorithms we learned from lecture and check whether a new algorithm leads to a better convergence rate compared to conventional machine learning packages such as support vector machine. The project finds 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), quasi-Newton algorithm(BFGS method) and BB (Barzilai-Borwein) Method (including LBB method and SBB method). In addition, our project not only contains analysis of real world dataset, but also includes comparison with artificial data simulated by gaussian distribution.

After comparing the algorithms regarding complexity, convergence speed and computation time, we find that for medical prediction problems with logistic regression model, Barzilai-Borwein method is the fastest and heavy ball method with parameter beta = 0.9 is the second best.

1.2 literature background and Motivation

In practice, all engineering designs should include optimization phases. Nowadays, with the development of artificial intelligence, machine learning becomes one of the hottest topics since it is a method to make predictions [1]. Since machine learning algorithms are key factors to efficiency and results, we are particularly interested in how different algorithms contribute to real world problems.

Traditional machine learning algorithms of classification, such as support vector machines (SVM) [2], has been well studied. Hirji, Karim, Anastasios, Tsiatis and Cyrus (1989) had studied how to deal with binary data in 1989, which provided a meaningful theory for prediction. At present, people are more likely to take advantage of gradient or momentum to logistic regression problems in practice, although 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 is full of interest to the technical community because it validates the theoretical results derived from the algorithms. How was the difference between theory and practice? Can we find an algorithm which has a better performance in logistic regression predictions? In order to find out whether a new algorithm would lead to a better convergence rate while conventional machine learning packages seem to work well, this project implements and compares different algorithms with a series of analysis.

1.3 Problem definition

Predictions happen frequently in medical field and the accuracy of prediction is particularly significant. Thus, our project focuses on prediction problems solved by logistic regression. Mathematical formula 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 data

4/14: Fit data using logistic regression

4/22: Implement 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 results and finish report

Methodology

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

2.1 Data description

After carefully examining 20 datasets, we chose to study “Breast Cancer Wisconsin Data Set” [3], a classic problem in this area. Our data was downloaded from Kaggle[3].

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 mean and scaling to unit variance. Furthermore, we removed some of the features with correlation below 0.02 to the response.

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.

2.2 Implementation of algorithms

Parameter tuning is an important part of exploring different convergence behavior. For parameter tuning, we tried Bayesian optimization with baseline of 1/L and found that it was very slow. Then, we decided to manually operate parameters with constant step-size of 1/L, 2/L, 3/L and the graphs below illustrate the result of 1/L, where L is the max eigenvalue of A and A=X’*X.

Analysis of convergence behavior in terms of convergence rate and efficiency is the main part of this project. Algorithms providing predictions include 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:

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

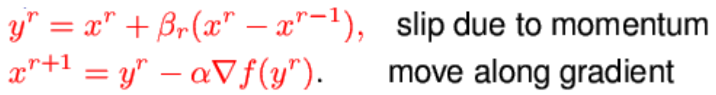
  Where α= 1/L, L is the max eigenvalue of A, and A=X’*X.

- Heavy ball method and Nesterov’s method:

  HB:

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

Nesterov:



$$y^r = x^r + \beta_r(x^r - x^{r-1}),$$ slip due to momentum

$$x^{r+1} = y^r - \alpha \nabla f(y^r).$$ move along gradient

- Stochastic gradient descent method (SGD):

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

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

- Quasi-Newton method(BFGS):

  The algorithm is:

  ```
  For  k=0,1,2….. n,
  initial $w_0$ and $B_0$
  Obtain a direction pk by $B^k*p^k= -\bigtriangledown f(w^k)$ ;   $w^{k+1}=w^k +\alpha^k p^k$ ;  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:

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

2.3 Experimentation and results

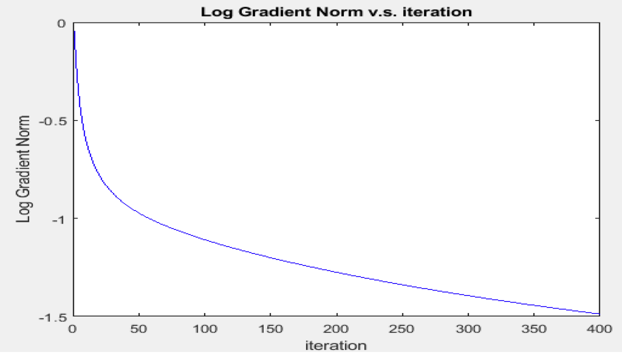
Our project not only contains experiments with real world dataset, but also contains experiments with artificial data with gaussian distribution as comparative tests.

All code for experiments see attach. They were written in Maltab. We also treid to write some code in python and Matlab was still at the top of the list for algorithmic prototyping, so all our code for optimization algorithms were written in python, only code for normalizing data was written in python.

2.3.1 Experiments with real world data

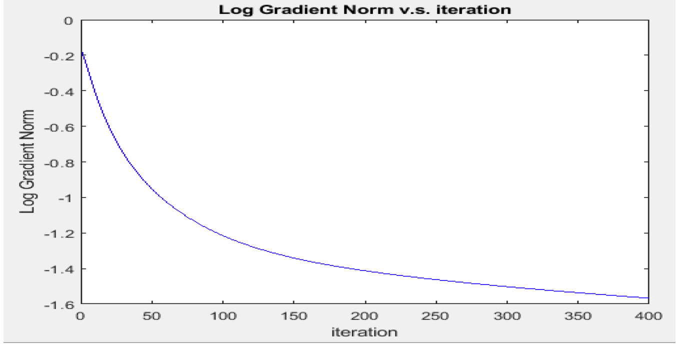
GD method

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.

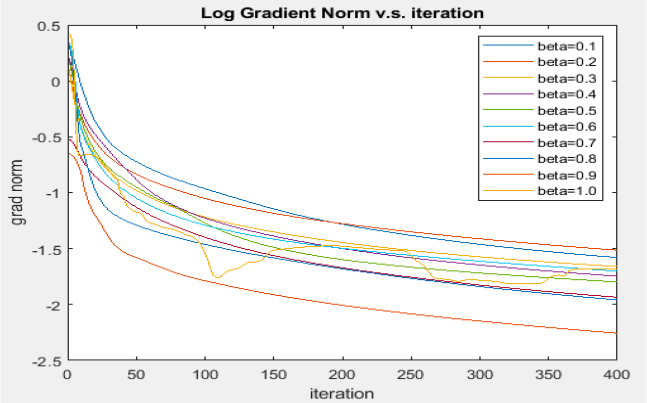


Figure 3

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.

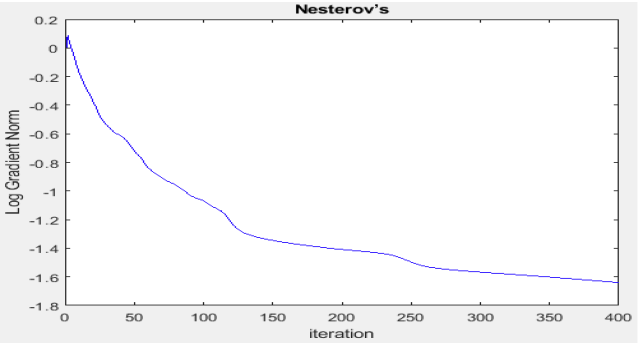


Figure 4

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.

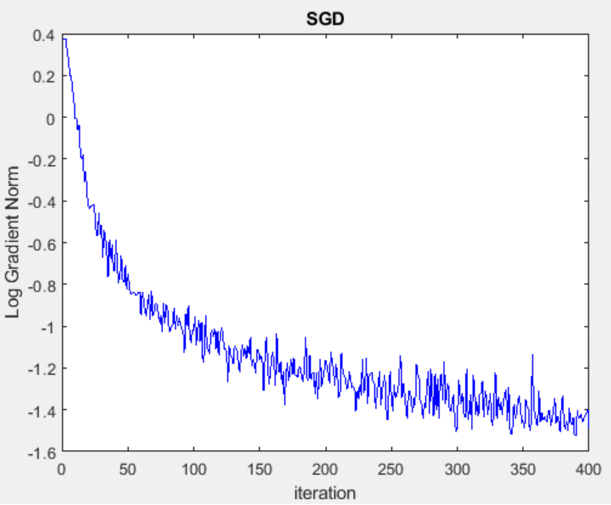
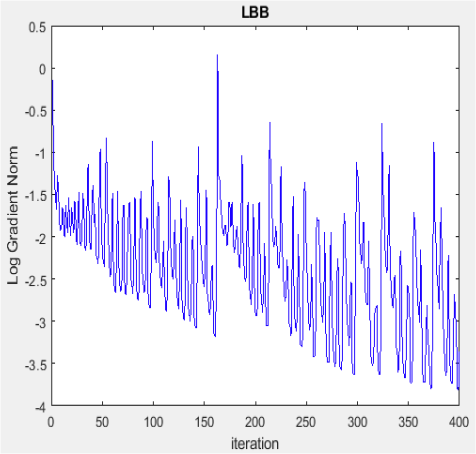


Figure 5

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.



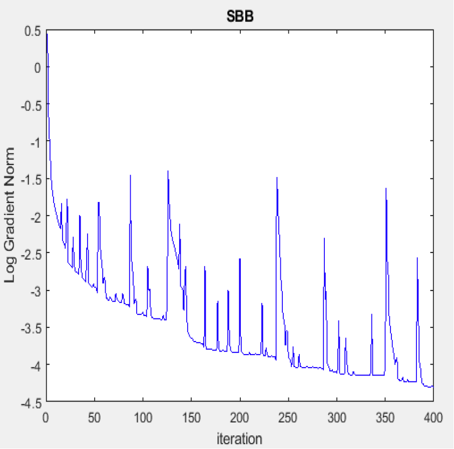


Figure 6

Figure 7

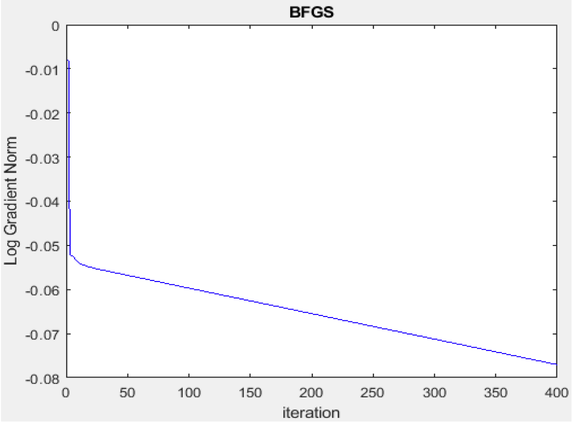
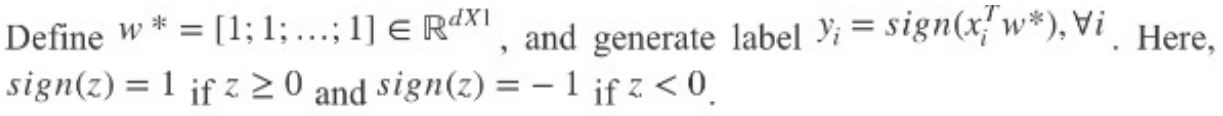


Figure 8

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.

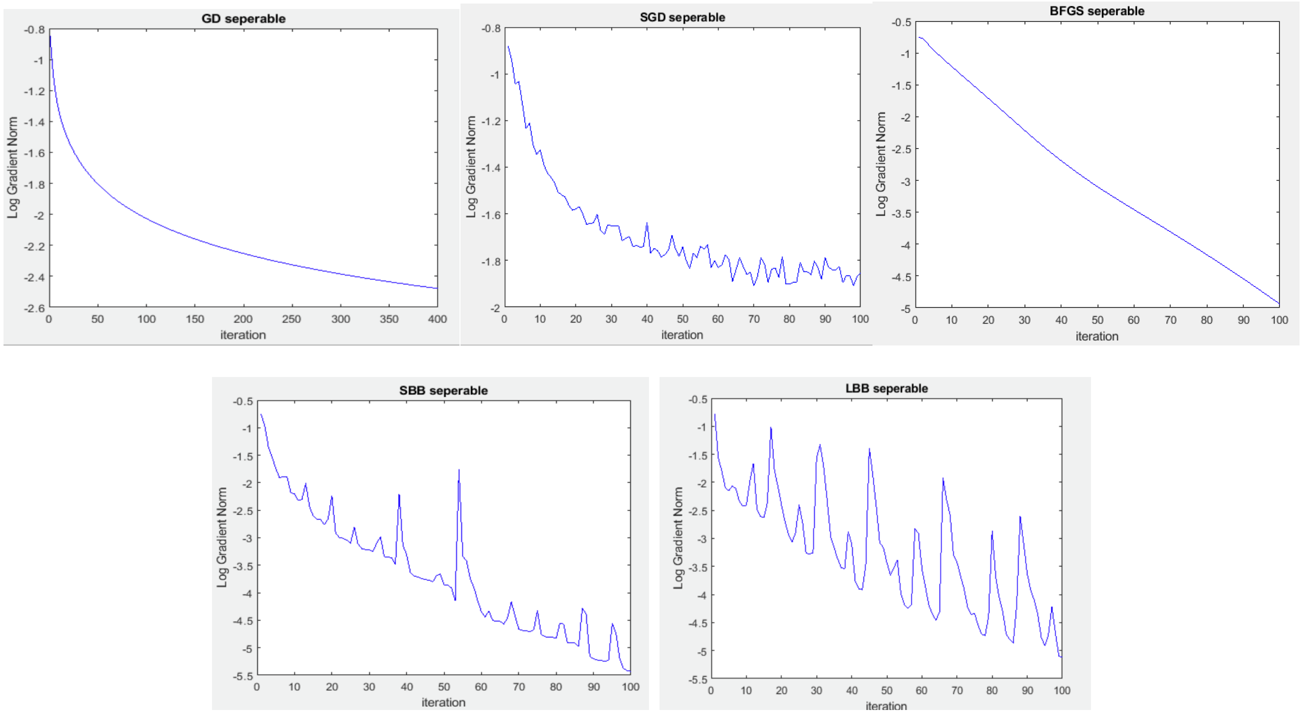


Figure 9

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 diverges 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 of negitive 4(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.

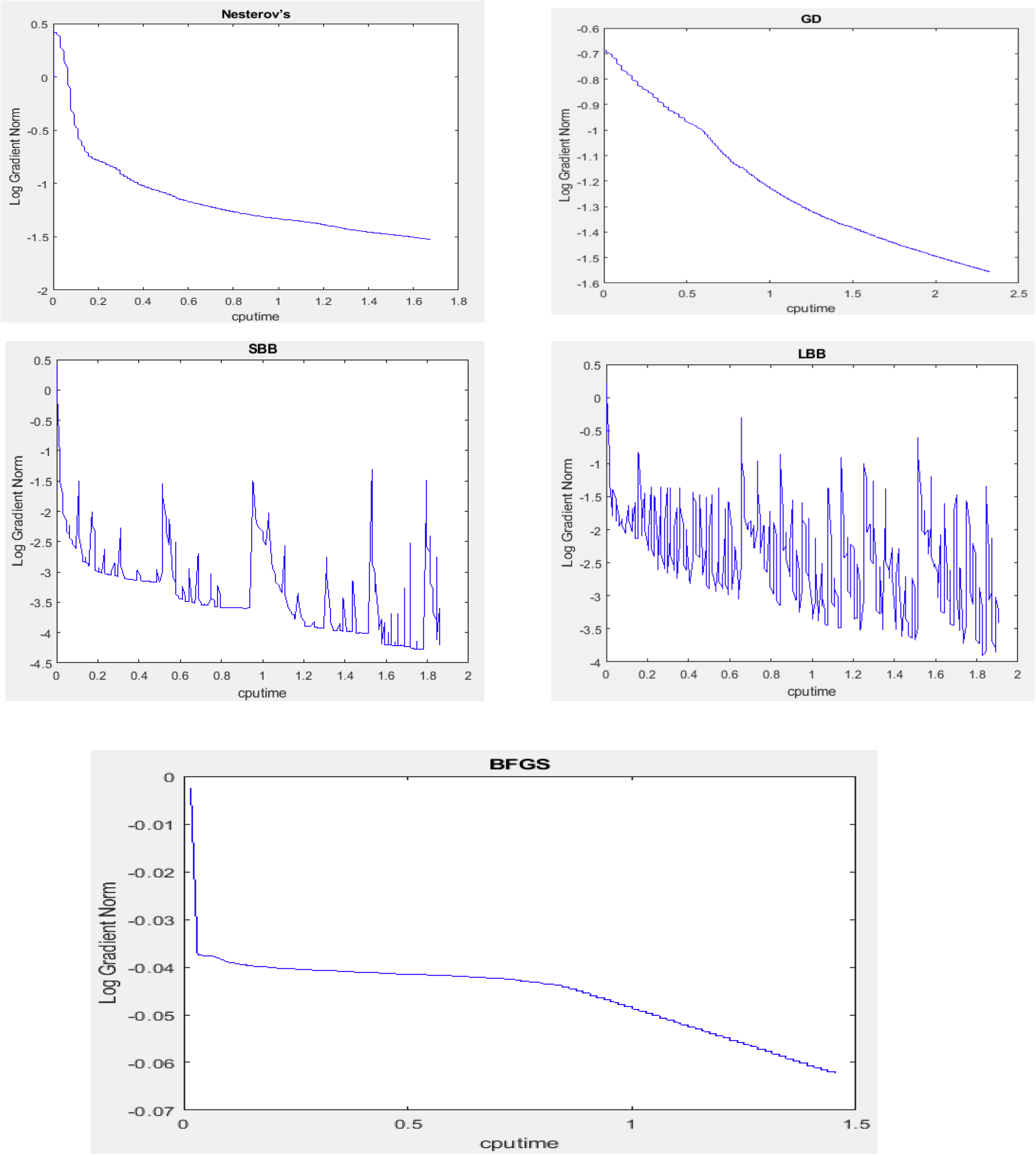


Figure 10

Conclusion

This project presents our study of convergence behavior 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 new methods, this project applies gradient descent, heavy ball method and Nesterov's method, stochastic descent, 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 show that constant step-size and classic routines for momentum are robust. They can explain why nearly all machine learning packages prefer to apply GD or momentum algorithm when handling logistic regression problems. However, we find that Barzilai-Borwein(BB) method has a better performance than momentum when considering the convergence rate.

We also find an interesting phenomenon in BFGS analysis where the results differ when using the real data and artificial data. For better result, we may introduce regularizer to resolve the issues in separable case.

As for future directions, tuning more parameters and 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