# AMS 598 - Project4 Report

Weiwei Tao

November 12, 2021

## 1 Problem Description

In this project, we are given 10 chunks of data, where each file contains 1 million rows and 26 variables (1 response variable and 25 feartures). The goal is to implement the Alternating Direction Method of Multipliers (ADMM) algorithm to run a logistic regression on 10 chunks of data and to obtain one set of consensus estimate of coefficients of the explanatory variables.

## 2 ADMM Algorithm

ADMM is an approach for convex optimization problems by breaking them into small pieces. ADMM ensures that the estimations on local node are coordinated with the global ground truth. We can write ADMM logistic regression problem as:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N} f_i(\beta_i) \\ \text{s.t.} \quad & \beta_i = \beta \end{aligned}$$

where $f_i(\beta_i)$ is the log loss function for $i^{th}$ block of training data, which is:

$$f_i(\beta_i) = -\frac{1}{m}\sum_{j=1}^{m}(y_i^{(j)}\log(\pi_i^{(j)}) + (1-y_i^{(i)})\log(1-\pi_i^{(j)}))$$

where $\pi_i^{(j)} = \text{Sigmoid}(X_i^{(j)}\beta_i)$. $\beta_i$ is the parameter estimated for $i^{th}$ data block and $\beta$ is the ground truth of estimation. We define augmented Lagrangian for a parameter $\rho > 0$:

$$L_\rho(\beta_i, \beta, z) = \sum_{i=1}^{N}(f_i(\beta_i) + z_i^T(\beta_i - \beta) + \rho/2\,\|\beta_i - \beta\|_2^2)$$

To solve for $\beta$, we repeat for $k = 1, 2, 3....n$

$$\beta_i^{k+1} := \arg\min_{\beta_i}(f_i(\beta_i) + z_i^{kT}(\beta_i - \bar{\beta}^k) + (\rho/2)\left\|\beta_i - \bar{\beta}^k\right\|_2^2)$$

$$z_i^{k+1} := z_i^k + \rho(\beta_i^{k+1} - \bar{\beta}^{k+1})$$

where $\bar{\beta}^k = \frac{1}{N}\sum_{i=1}^{N}(\beta_i^{k+1})$. We can further simplify the equation as:

$$\beta_i^{k+1} := \arg\min_{\beta_i}(f_i(\beta_i) + (\rho/2)\left\|\beta_i - \bar{\beta}^k + u_i^k\right\|_2^2)$$

$$u_i^{k+1} := u_i^k + (\beta_i^{k+1} - \bar{\beta}^{k+1})$$

Thus for each iteration of our analysis, we will do

1. Fit logistic regression to obtain $\beta_i^k$ which minimize the augmented Lagrangian.
2. Gather $\beta_i^k$ and average to get $\bar{\beta}^k$.
3. Scatter $\bar{\beta}^k$ to each processor and find out $z_i^{k+1}$ and $\beta_i^{k+1}$.

Repeat the above processes until $\beta_i^k$ converges.

# 3 Implementing in Python

To implement ADMM in python, I performed following analysis:

1. Load all 10 data chunks and combine them together. Check the data to make sure no missing values and muticollinearity between features. Standardize feature columns and build a logistic regression model based upon all data as the baseline model.
2. Initialize $\beta_i$, $\beta$, $u_i$ using zero vectors. Use optimization function in Scipy package to find optimal $\beta_i^k$ and update $\beta^k$ and $u_i^k$ accordingly. Repeat the algorithm until reaches 100 steps (note that if time allows, we can also set convergence criteria based upon $\beta$ values. )
3. Test the algorithm with $\rho = 0.1, 0.5, 1, 5$ and number of iterations of 100.

Table 1 summarizes results for both the baseline model and ADMM models with different $\rho$ values. It seems that the higher the $\rho$ value is, the slower the model converges to reference. However, even after 100 iterations, the estimated coefficient is pretty off from the ground truth (reference column in Table 1). But the estimated values are proportional to the ground truth for each feature.

| Variables | Reference (All data) | $\rho = 0.1$ | $\rho = 0.5$ | $\rho = 1$ | $\rho = 5$ |
|---|---|---|---|---|---|
| const | 9.5987 | 7.44045943 | 5.0129 | 4.1416 | 2.5651 |
| x1 | -0.002 | -0.0053504 | -0.0037 | -0.003 | -0.0013 |
| x2 | 0.0035 | 0.00393685 | 0.0023 | 0.0017 | 0.0005 |
| x3 | 0.0033 | -1.957E-05 | 0.0001 | 0.0001 | 0.0001 |
| x4 | 0.0004 | -0.001375 | -0.0009 | -0.0006 | -0.0001 |
| x5 | 0.9921 | 0.74793281 | 0.4489 | 0.3306 | 0.12 |
| x6 | 3.6942 | 2.70710175 | 1.5749 | 1.1638 | 0.4397 |
| x7 | 0.0013 | -0.0014926 | -0.0008 | -0.0006 | -0.0002 |
| x8 | -0.0042 | 0.00062767 | 0.0004 | 0.0003 | 0.0003 |
| x9 | 0.0044 | 0.00294595 | 0.0024 | 0.0021 | 0.0009 |
| x10 | 0.3055 | 0.23632687 | 0.1533 | 0.1198 | 0.0494 |
| x11 | -0.0017 | -0.0040493 | -0.003 | -0.0025 | -0.0011 |
| x12 | -0.012 | -0.0031862 | -0.0017 | -0.0011 | -0.0004 |
| x13 | 0.0021 | -0.0009433 | -0.0005 | -0.0004 | -0.0001 |
| x14 | 0.9938 | 0.76470085 | 0.4886 | 0.3769 | 0.1502 |
| x15 | -0.0022 | -0.001403 | -0.0012 | -0.001 | -0.0005 |
| x16 | 0.0076 | 0.00679977 | 0.0048 | 0.0039 | 0.0018 |
| x17 | 0.0013 | -0.0015932 | -0.0008 | -0.0006 | -0.0002 |
| x18 | 1.7828 | 1.3730482 | 0.8808 | 0.6821 | 0.2741 |
| x19 | -0.0003 | -0.0006573 | -0.0005 | -0.0004 | 0 |
| x20 | -0.0066 | -0.0026687 | -0.002 | -0.0018 | -0.0009 |
| x21 | -0.0061 | -0.0016666 | -0.001 | -0.0006 | -0.0001 |
| x22 | -0.0063 | -0.0066944 | -0.0041 | -0.0031 | -0.0012 |
| x23 | 2.9852 | 2.29937953 | 1.4753 | 1.1426 | 0.4591 |
| x24 | -0.0002 | 0.00163926 | 0.0013 | 0.0011 | 0.0005 |
| x25 | -0.0096 | -8.619E-05 | 0.0002 | 0.0001 | 0.0001 |

Table 1: Logistic regression coefficients for baseline model and ADMM model with different $\rho$ values. Number of iterations were set to 100 for all $\rho$.