Final project (Students should submit their homework before 10 a.m. on June 24, 2020.)

In this project students are asked to estimate a logistic regression model using different **estimation methods** and different **optimization algorithms**. Before listing the estimation methods and algorithms, we first describe a simulation process for data generation.

Data generation: We consider a situation when the log-odds of the probability $\mathbb{P}(y_i = 1)$ is a linear combination of groups of covariates:

$$\log\left(\frac{\mathbb{P}(y_i=1)}{1-\mathbb{P}(y_i=1)}\right) = \sum_{j=1}^{m} \mathbf{x}_{ij}^T \boldsymbol{\beta}_j^{\text{true}},$$

where $\mathbf{x}_{ij}$ is a p_j-dimensional vector of covariates associated with the ith observation, and $\boldsymbol{\beta}_j^{\text{true}}$ is the regression coefficient vector associated with the jth group of covariates. To generate the data, let the number of groups $m = 200$ and assume each $\boldsymbol{\beta}_j^{\text{true}}$ is a $p_j = 5$-dimensional vector. We use the following model to generate the data:

$$\begin{aligned}
\boldsymbol{\beta}_{100}^{\text{true}} &= (-1, 1, -1, 1, -1), \\
\boldsymbol{\beta}_{200}^{\text{true}} &= (1, -1, 1, -1, 1), \\
\boldsymbol{\beta}_{j}^{\text{true}} &= \mathbf{0}, j \in \{1, 2, \cdots, 200\} \setminus \{100, 200\}, \\
\mathbf{x}_{ij} &= (x_{ij1}, x_{ij2}, \cdots, x_{ij5}), \\
x_{ijk} &\sim \text{Normal}(0, 1) \text{ for } j = 1, 2, \cdots, 200 \text{ and } k = 1, 2, \cdots, 5, \\
y_i &\sim \text{Bernoulli}\left(\frac{\exp(\sum_{j=1}^{m} \mathbf{x}_{ij}^T \boldsymbol{\beta}_j^{\text{true}})}{1 + \exp(\sum_{j=1}^{m} \mathbf{x}_{ij}^T \boldsymbol{\beta}_j^{\text{true}})}\right).
\end{aligned}$$

Here we have assumed **only the 100th and 200th $\boldsymbol{\beta}_j^{\text{true}}$ have nonzero-valued regression coefficients**. The rest of $\boldsymbol{\beta}_j^{\text{true}}$'s are zero-valued vectors. Therefore from the definitions given above, we have

$$\sum_{j=1}^{200} \mathbf{x}_{ij}^T \boldsymbol{\beta}_j^{\text{true}} = \mathbf{x}_{i,100}^T \boldsymbol{\beta}_{100}^{\text{true}} + \mathbf{x}_{i,200}^T \boldsymbol{\beta}_{200}^{\text{true}}.$$

We consider to use the above setting to generate both **training** data and **test** data.

We fixed the size of training data at

$$n^{\text{train}} = 200.$$

Students should check if the size of the training covariate matrix $\mathbf{X}^{\text{train}}$ is an $n^{\text{train}} \times (\sum_{j=1}^{200} p_j) = 200 \times 1000$ matrix.

Below students are asked to use estimation results from the training data for prediction. Here the test data are used to validate such prediction. The size of test data is set to be $n^{\text{test}} = 10 \times n^{\text{train}}$.

Statistical estimation

Below we describe several methods for estimating regression coefficients $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_m)$. We first derive a loss function for $\boldsymbol{\beta}$.

Loss function construction: Assume that we have observed data $\{y_i, \mathbf{x}'_{ij}s\}_{i=1}^n$, where $y_i \in \{0, 1\}$ is the label of observation i and $\mathbf{x}_{ij}$ is a p_j-dimensional vector of the corresponding covariates. Our aim is to build a regression model using $\mathbf{x}_{ij}$'s to predict y_i. A common way is to model the log odds of the probability that $y_i = 1$ as a linear combination of $\mathbf{x}_{ij}$'s:

$$\log\left(\frac{\mathbb{P}(y_i = 1)}{1 - \mathbb{P}(y_i = 1)}\right) = \sum_{j=1}^{m} \mathbf{x}_{ij}^T \boldsymbol{\beta}_j = f_i.$$

The corresponding *minus* log likelihood function of $\boldsymbol{\beta}$ is

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \Big\{ -y_i f_i + \log\Big[1 + \exp(f_i)\Big]\Big\}.$$

Estimation methods: We consider four estimation methods:

i. **The lasso estimation (lasso):** It is defined by

$$\widehat{\boldsymbol{\beta}}^{\text{lasso}} = \arg\min_{\boldsymbol{\beta}} \Big\{ l(\boldsymbol{\beta}) + \lambda ||\boldsymbol{\beta}||_1 \Big\},$$

where $l(\boldsymbol{\beta})$ is defined in (1), and $\lambda \geq 0$ is a scalar-valued tuning parameter.

ii. **The l_0-norm penalized estimation (zero):** It is defined by

$$\widehat{\boldsymbol{\beta}}^{\text{zero}} = \arg\min_{\boldsymbol{\beta}} \left\{ l(\boldsymbol{\beta}) + \lambda ||\boldsymbol{\beta}||_0 \right\},$$

where $l(\boldsymbol{\beta})$ is defined in (1), and $\lambda \geq 0$ is a scalar-valued tuning parameter.

iii. **The l_0-norm constrained estimation (crt0):** It is defined by

$$\widehat{\boldsymbol{\beta}}^{\text{crt0}} = \arg\min_{\boldsymbol{\beta}} \left\{ l(\boldsymbol{\beta}) + \iota\{||\boldsymbol{\beta}||_0 \leq \lambda\} \right\},$$

where $l(\boldsymbol{\beta})$ is defined in (1), and $\iota\{\mathcal{A}\} = 0$ if $\mathcal{A}$ is true, and $\iota\{\mathcal{A}\} = 0$ otherwise, $\lambda \in \{0, 1, 2, \cdots\}$ is a nonnegative integer.

iv. **The group lasso estimation (grp-lasso):** It is defined by

$$\widehat{\boldsymbol{\beta}}^{\text{grp-lasso}} = \arg\min_{\boldsymbol{\beta}} \left\{ l(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{m} \sqrt{p_j} ||\boldsymbol{\beta}_j||_2 \right\},$$

where $l(\boldsymbol{\beta})$ is defined in (1), $\lambda \geq 0$ is a tuning parameter, and p_j is the number of elements in $\boldsymbol{\beta}_j$.

Optimization algorithms

We consider the following four optimization algorithms:

a. The alternating direction method of multipliers (**ADMM**, Week 13);

b. The proximal gradient algorithm (**PG**, Week 14).

c. The fast proximal gradient algorithm (**FPG**, Week 14);

d. The stochastic subgradient algorithm (**SG**, Week 15);

Programming work

Students should choose two estimation methods (i to iv) and two optimization algorithms (a to d) for implementation and report their results for the 4 (2 methods $\times$ 2 algorithms) cases. For reporting details, please see below. Students may follow

steps introduced in previous homework to specify the **stepsize** and **stopping criterion** for implementing those algorithms. For each algorithm, the **tolerance** for the iteration error and the **maximum number of iterations** should be

$$\begin{aligned} \text{tol} &= 5 \times 10^{-6}, \\ \text{max\_iter} &= 10,000. \end{aligned}$$

Tasks: Students should report the following tasks for their results:

- **1. Line plots for iteration errors (2%):** For the first estimation method you choose for implementation, fix the tuning parameter λ at 2 different values you like and run the iterative schemes **under the 2 different values of λ, separately**. Produce 2 plots for the 2 different values of λ with the following format: The x-axis is the number of iterations r and the y-axis is the **iteration error** at rth iteration. You should use different colors to indicate iteration errors produced by the 2 chosen optimization algorithms.

- **2. Line plots for iteration errors (2%):** For the second estimation method you choose for implementation, follow exact the same procedure as in the Task 1 to produce the line plots of iteration errors.

- **3. Table for performance measures (6%):** For prediction performance, report the following values using your estimation results:

  - The value of tuning parameter λ^* used in each estimation method that leads to the best prediction performance.
  - The mean squared error of $\widehat{\boldsymbol{\beta}}^{\cdot}$, which is defined by

    $$\text{MSE} = ||\widehat{\boldsymbol{\beta}}^{\cdot} - \boldsymbol{\beta}^{\text{true}}||_2^2,$$

  - The training error, which is defined by

    $$\text{Err}^{\text{train}} = \frac{\sum_{i=1}^{n} \mathbb{I}\{\widehat{y}_i \neq y_i\}}{n^{\text{train}}},$$

    where $\mathbb{I}\{\mathcal{A}\}$ is an indicator function such that $\mathbb{I}\{\mathcal{A}\} = 1$ if $\mathcal{A}$ is true and $\mathbb{I}\{\mathcal{A}\} = 0$ otherwise, and y_i is the ith response in the training data.

	λ^*	MSE	$\text{Err}^{\text{train}}$	Err^{test}
Estimation method 1 (algorithm 1)				
Estimation method 2 (algorithm 1)				
Estimation method 1 (algorithm 2)				
Estimation method 2 (algorithm 2)				

Table 1: Table format.

– The test error, which is defined by

$$\text{Err}^{\text{test}} = \frac{\sum_{i=1}^{n} \mathbb{I}\{\widehat{y}_i^{\text{test},\cdot} \neq y_i^{\text{test}}\}}{n^{\text{test}}}.$$

You may report the results in the Table 1 format.

Important: Please use abbreviation (stated in the brackets) to indicate which estimation methods and which optimization algorithms you implement to obtain the estimation results.