A-optimal Leveraging for Logistic Regression with Big Data

Rong Zhu, HaiYing Wang* and Ping Ma
*University of University of New Hampshire

Introduction

The Logistic regression model assume that, given a covariate $\mathbf{x} \in \mathbb{R}^d$, the model assumes

$$P(y=1) = p(\mathbf{x};\boldsymbol{\beta}) = \frac{\exp(\mathbf{x}^T\boldsymbol{\beta})}{1+\exp(\mathbf{x}^T\boldsymbol{\beta})}, \qquad (1)$$

where $y \in \{0,1\}$ is the response variable; $\boldsymbol{\beta}$ is a $d \times 1$ vector of unknown regression parameters. For massive data (n is large), it is computationally difficult to find the MLE, $\hat{\boldsymbol{\beta}}$. The aim of this work is to approximate the MLE for logistic regression efficiently to deal with Big Data.

Leveraging methods

- Leveraging methods are designed under a sub-sampling framework, in which we sample a small proportion of the data from the full sample, and then used as a surrogate to perform intended computations for the full sample.
- The key of the success of the leveraging methods relies on effectively constructing nonuniform sampling probabilities so that influential data points are to be sampled with high probabilities.
- All existing literature on leveraging methods are on solving the OLS in linear regression with Big Data.
- The time complexity of solving OLS using the full data is $O(nd^2)$.
- Leveraging methods often has a time complexity of $O(nd)$.
- The time complexity of for solving the MLE using the full data is $O(\zeta nd^2)$, where ζ is the number of iterations required for the optimization procedure to converge.
- Our method has a time complexity of $O(nd)$.

General Sub-sampling Algorithm

- **Sub-sampling.**
  - Assign sampling probability $\{\pi_i\}_{i=1}^n$ for all data points.
  - Draw a random sub-sample of size $r \ll n$ from the full sample according to the probability $\{\pi_i\}_{i=1}^n$, denoted as $(\mathbf{X}^*, \mathbf{y}^*)$.
  - Record the corresponding sampling probabilities for the sub-sample $\{\pi_k^*\}$, $k = 1, \ldots, r$.
- **Estimation.**
  - Maximize a weighted log-likelihood to get an estimate $\tilde{\boldsymbol{\beta}}$, of $\hat{\boldsymbol{\beta}}$, i.e., solve:

$$\arg\max_{\boldsymbol{\beta}\in\mathbb{R}^d} \sum_{i=1}^r \frac{1}{\pi_i^*}\{y_i^* \log p(\mathbf{x}_i^*;\boldsymbol{\beta}) + (1-y_i^*)\log(1-p(\mathbf{x}_i^*;\boldsymbol{\beta}))\}$$

Theorem: asymptotic properties

Let $\mathcal{F}_n$ be the full data and $\hat{\boldsymbol{\beta}}$ be the MLE using $\mathcal{F}_n$. Under Assumptions A1-A3, given $\mathcal{F}_n$, as $n, r \to \infty$,

$$\mathbf{V}^{-1/2}(\tilde{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}) \xrightarrow{L} N(0, I), \qquad (2)$$

where $\mathbf{V} = \mathbf{M}_X^{-1}\mathbf{V}_b\mathbf{M}_X^{-1}$,

$\mathbf{M}_X = \sum_{i=1}^n w_i\mathbf{x}_i\mathbf{x}_i^T$,

$w_i = p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})\{1-p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})\}$ and

$\mathbf{V}_b = r^{-1}\sum_{i=1}^n \left\{y_i - p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})\right\}^2 \mathbf{x}_i\mathbf{x}_i^T/\pi_i$.

Moreover,

$$\mathbf{V} = O_p(r^{-1}), \text{ and } \quad E(\tilde{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}|\mathcal{F}_n) = O_p(r^{-1}).$$

Theorem: A-optimal sub-sampling

The variance dominates the mean-squared error, and $\mathbf{M}_X$ is independent of π_i. So we choose to minimize $tr(\mathbf{V}_b)$ as an optimal criterion.

If the sub-sampling probabilities are chosen such that

$$\pi_{iOPTA} = \frac{|y_i - p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})|\|\mathbf{x}_i\|}{\sum_{j=1}^n |y_j - p(\mathbf{x}_j;\hat{\boldsymbol{\beta}})|\|\mathbf{x}_j\|}, \quad i = 1, 2, \ldots, n, \qquad (3)$$

then $\mathrm{tr}(\mathbf{V}_b)$ attains its minimum.

Remarks

The optimal sub-sampling probabilities are determined by two factors.

- **Covariate information represented by** $\|\mathbf{x}_i\|$:
- **Discrimination difficulty represented by** $|y_i - p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})|$
  - If $y_i = 0$;
    $$\pi_{iOPTA} \propto p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})\|\mathbf{x}_i\|$$
  - If $y_i = 1$
    $$\pi_{iOPTA} \propto \{1 - p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})\}\|\mathbf{x}_i\|$$
  - This echos the result of Silvapulle (JRSSB 1981).

Expected A-optimal sub-sampling

Another way is to minimize $E\{tr(\mathbf{V}_b|\mathbf{X})\}$. If the sub-sampling probabilities are chosen such that

$$\pi_{iOPTB} = \frac{\sqrt{p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})\{1-p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})\}}\|\mathbf{x}_j\|}{\sum_{j=1}^n \sqrt{p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})\{1-p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})\}}\|\mathbf{x}_j\|}, i = 1, 2, \ldots, n,$$

then $E\{\mathrm{tr}(\mathbf{V}_b|\mathbf{X})\}$ attains its minimum.

Remarks

The optimal SSP are determined by:

- **Covariate information represented by** $\|\mathbf{x}_i\|$:
- **Discrimination difficulty:** $p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})\{1-p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})\}$:
  - it reaches its maximum when $p(\mathbf{x}_i;\hat{\boldsymbol{\beta}}) = 0.5$
  - The closer to 0.5 the value of $p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})$ is, the more difficult it's to classify these data points into their true categories, so more sub-sampling probabilities are put into these data points.

Two step algorithms

The optimal weight choices depend on $\hat{\boldsymbol{\beta}}$, we propose the following two-step algorithm.

- Sampling for a sub-sample of size r_1 by SRS, obtain an estimate $\tilde{\boldsymbol{\beta}}_1$ and estimate the optimal SSPs.
- Sampling a sub-sample of size r_2 with SSPs calculated in Step 1, and obtain the estimate $\tilde{\boldsymbol{\beta}}$.

Simulation settings

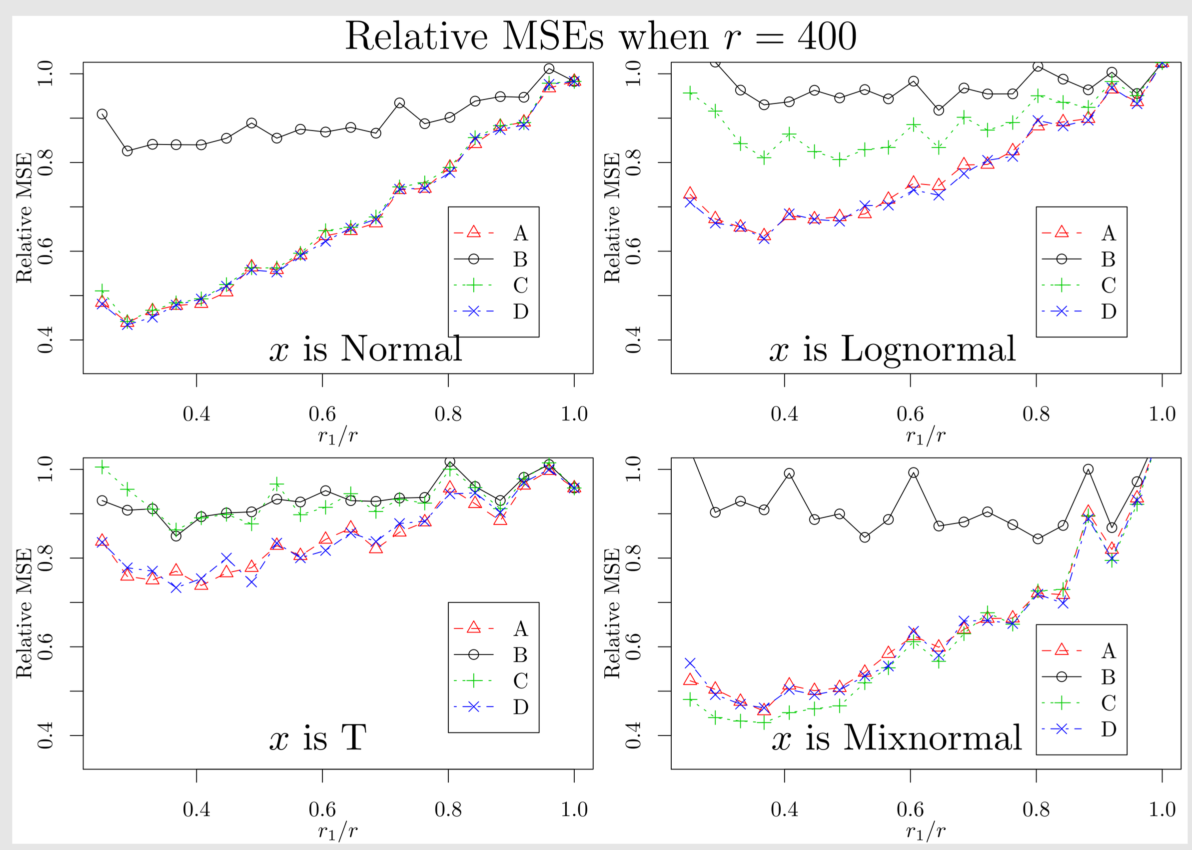
In addition to the weights proposed, we also consider the following sub-sampling probabilities (SSP) for comparison.

$$\text{SSP C: } \pi_i \propto |y_i - p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})|$$
$$\text{SSP D: } \pi_i \propto |y_i - p(\mathbf{x}_i;\hat{\boldsymbol{\beta}})|\|\mathbf{x}_i\|_\infty$$

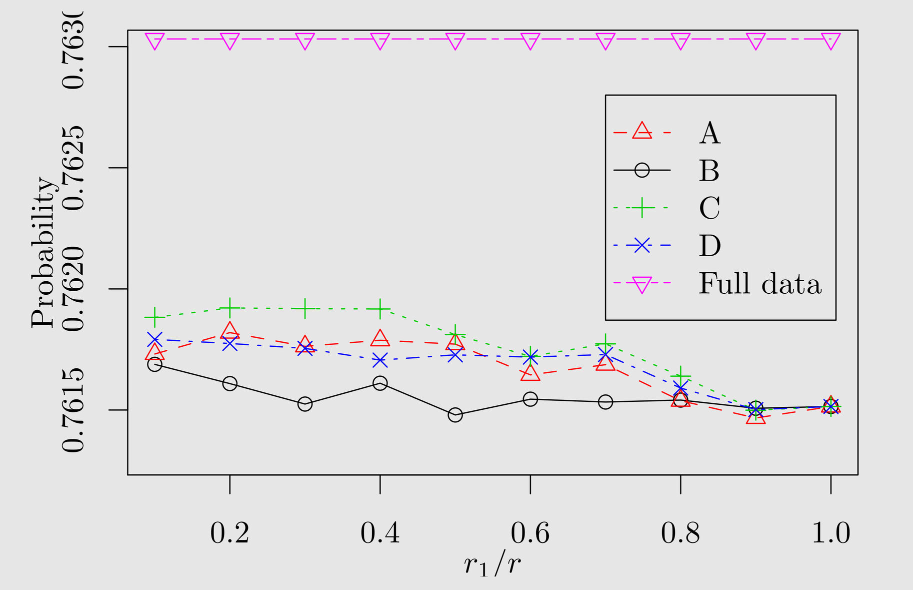
A relative MSE is a MSE scaled by the MSE of the estimator from a uniform sub-sampling with the same sub-sample size r.

$$\frac{\text{MSE of the estimator from a proposed SSP}}{\text{MSE of the estimator from a uniform Sampling}}$$

Results for simulation studies



Results for income data



Results for SUSY data

To distinguish between a process where new supersymmetric particles are produced and a background process. The sample size $n = 5,000,000$ and the data file is 2.4GB. We used a sub-sample of $r = 1000$.

Features used	SSP A	SSP B	SSP C	SSP D
First 8	0.827	0.826	0.831	0.825
Last 10	0.832	0.830	0.830	0.830
All	0.850	0.851	0.853	0.852

Comparisons with Deep Learning:

Our method	DL
AUC=0.853	AUC=0.88
$r = 1000$	$n = 5,000,000$
Logistic model	A five-layer neural nets with 300 hidden units in each layer
R with package BB	Combinations of pre-training methods, network architectures, initial learning rates, and regularization methods
A normal PC with an Intel I7 processor and 8GB memory	Machines with 16 Intel Xeon cores, an NVIDIA Tesla C2070 graphics processor, and 64 GB memory. All neural networks were trained using the GPU-accelerated Theano and Pylearn2 software libraries