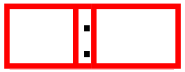
Informative Subdata Selection for Big Data

HaiYing Wang

Department of Mathematics and Statistics
University of New Hampshire, Durham, NH, USA

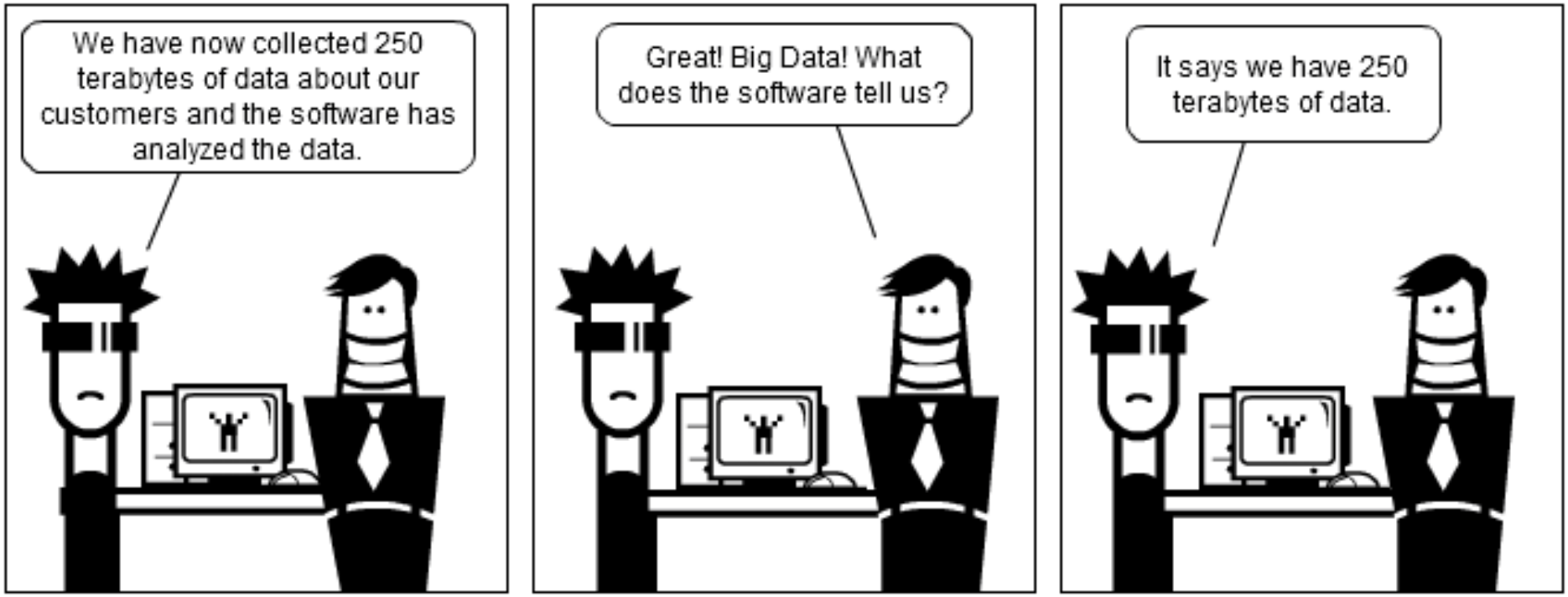
August 16, 2016



Outline

Outline

Big Data challenge and data reduction



Big Data challenge and data reduction

- A common challenge from Big Data is how to extract useful information with available computational facilities.
- Data reduction is crucial.
  - There is not enough computing resources to analyze the full data.
  - It is inconvenient to work with big full data.
  - It is not always possible to store the data in full.
- Subsampling of big data is a cutting-edge problem and already has gotten a lot of traction in the field of computer science (CS).
- However, most CS-style investigations have very real limitations.
  - The applications mostly focus on speeding up algorithms.
  - They do not provide distributional results.

Subsampling-based methods

- The key is to effectively construct nonuniform sampling probabilities so that influential data points are sampled with high probabilities.
- Most of the existing methods use the normalized leverage scores as subsampling probabilities.
- A major limitation of this approach is that information obtained is typically at the scale of the subdata size and not the full data size.
- In other words, for a fixed subdata size, the variance does not go to 0 as $n \to \infty$.

References on subsampling-based method

- Drineas, P., Mahoney, M. W., and Muthukrishnan, S. (2006). Sampling algorithms for L_2 regression and applications. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1127–1136.
- Rokhlin, V. and Tygert, M. (2008) A fast randomized algorithm for overdetermined linear least-squares regression. *Proceedings of the National Academy of Sciences of the United States of America*, 105(36):13212–13217.
- Drineas, P., Mahoney, M., Muthukrishnan, S., and Sarlos, T. (2011). Faster least squares approximation. *Numerische Mathematik* **117**, 219–249.
- Dhillon, P., Lu, Y., Foster, D. P. and Ungar, L. (2013) New subsampling algorithms for fast least squares regression. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 360–368.
- Ma, P., Mahoney, M., and Yu, B. (2015). A statistical perspective on algorithmic leveraging. *Journal of Machine Learning Research* **16**, 861–911.
- Ma, P. and Sun, X. (2015). Leveraging for big data regression. *Wiley Interdisciplinary Reviews: Computational Statistics* **7**, 1, 70–76.
- R. Zhu, P. Ma, M. Mahoney, and B. Yu. Optimal leveraging for linear regression in super-large sample. Technical report, 2016.

Outline

Logistic regression

- It is a statistical model that is widely used for inference and classification in many disciplines.
- For a given covariate $\mathbf{x} \in \mathbb{R}^d$, the model assumes that

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

  where
  $y \in \{0, 1\}$ is the response variable
  $\boldsymbol{\beta}$ is a $d \times 1$ vector of unknown regression parameters.

- In a classification problem, we assume that the training data set consists of n data points $\{\mathbf{x}_i, y_i\}$, for $i = 1, 2, \cdots, n$.

Maximum likelihood

- $\boldsymbol{\beta}$ is often estimated by the method of maximum likelihood (ML), say $\hat{\boldsymbol{\beta}}_{\text{MLE}}$, which is the maximizer of the following log-likelihood.

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \{y_i \log p(\mathbf{x}_i; \boldsymbol{\beta}) + (1 - y_i) \log(1 - p(\mathbf{x}_i; \boldsymbol{\beta}))\}. \qquad (2)$$

- For massive data (n is large), this takes $O(nd^2\zeta)$ time.
- The aim of this work is to approximate the MLE for logistic regression efficiently for Big Data.

The General Sub-sampling Approach

- **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 $(\mathbf{X}^*, \boldsymbol{y}^*)$.
- **Estimation.**
  - Maximize a weighted log-likelihood to get an estimate $\tilde{\boldsymbol{\beta}}$, of $\hat{\boldsymbol{\beta}}_{\text{MLE}}$, i.e., solve:

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

OSMAC

- An easy way to determine π_i is to perform a uniform random sampling, in which $\pi_i = 1/n$.
- However a uniform sampling may not be effective, as all the data points are viewed as equally important.
- The strategy is to derive the asymptotic distribution of $\tilde{\boldsymbol{\beta}}$, and then derive π_i's that minimize the asymptotic MSE of the resultant estimator.
- It is **O**ptimal **S**ubsampling **M**ethod under the **A**-optimality **C**riterion. We call our method the OSMAC.

Asymptotic distribution

Theorem

Under some regularity conditions, as $n \to \infty$ *and* $r \to \infty$*, conditional on* $\mathcal{F}_n$ *in probability,*

$$\mathbf{V}^{-1/2}(\tilde{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{\text{MLE}}) \longrightarrow N(0, \mathbf{I}) \tag{3}$$

in distribution, where

$$\mathbf{V} = \mathbf{M}_X^{-1}\mathbf{V}_c\mathbf{M}_X^{-1} = O_p(r^{-1}), \tag{4}$$

$$\mathbf{V}_c = \frac{1}{rn^2}\sum_{i=1}^{n}\frac{\{y_i - p_i(\hat{\boldsymbol{\beta}}_{\text{MLE}})\}^2\mathbf{x}_i\mathbf{x}_i^T}{\pi_i}, \tag{5}$$

$$\mathbf{M}_X = n^{-1}\sum_{i=1}^{n} w_i(\hat{\boldsymbol{\beta}}_{\text{MLE}})\mathbf{x}_i\mathbf{x}_i^T, \quad \textit{and} \quad w_i(\boldsymbol{\beta}) = p_i(\boldsymbol{\beta})\{1 - p_i(\boldsymbol{\beta})\}. \tag{6}$$

OSMAC: Minimum Asymptotic MSE of $\hat{\beta}$

- From the A-optimality criterion, we choose to minimize $\mathrm{tr}(\mathbf{V})$ as an optimal criterion.
- This is the same as minimizing the Asymptotic MSE of $\tilde{\boldsymbol{\beta}}$

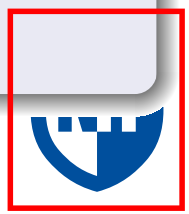
$$\mathrm{AMSE}(\tilde{\boldsymbol{\beta}}) = \mathrm{tr}(\mathbf{V}) \quad \text{since} \quad \tilde{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{\mathrm{MLE}} \big| \mathcal{F}_n \overset{a}{\sim} N(0, \mathbf{V}) \tag{7}$$

Theorem

In Algorithm 1, if the subsampling probability (SSP) is chosen such that

$$\pi_i^{\mathrm{mMSE}} = \frac{|y_i - p_i(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})| \|\mathbf{M}_X^{-1}\mathbf{x}_i\|}{\sum_{j=1}^{n} |y_j - p_j(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})| \|\mathbf{M}_X^{-1}\mathbf{x}_j\|}, \ i = 1, 2, ..., n, \tag{8}$$

then the asymptotic MSE of $\tilde{\boldsymbol{\beta}}$, $\mathrm{tr}(\mathbf{V})$, *attains its minimum.*



OSMAC: Minimum Asymptotic MSE of I

$$\mathbf{V} = \mathbf{M}_X^{-1}\mathbf{V}_c\mathbf{M}_X^{-1}$$

$$\mathbf{M}_X(\tilde{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{\text{MLE}})\big|\mathcal{F}_n \overset{a}{\sim} N(0, \mathbf{V}_c) \tag{9}$$

Another criterion is to minimize $\text{tr}(\mathbf{V}_c)$, which is the asymptotic MSE of $\mathbf{M}_X\tilde{\boldsymbol{\beta}}$.

Theorem

In Algorithm 1, if the sub-sampling probabilities are chosen such that

$$\pi^{\text{mVc}} = \frac{|y_i - p(\mathbf{x}_i; \hat{\boldsymbol{\beta}}_{\text{MLE}})|\|\boldsymbol{x}_i\|}{\sum_{j=1}^{n} |y_j - p(\mathbf{x}_j; \hat{\boldsymbol{\beta}}_{\text{MLE}})|\|\boldsymbol{x}_j\|}, \ i = 1, 2, ..., n, \tag{10}$$

then $\text{tr}(\mathbf{V}_c)$ *attains its minimum.*

Notes

$$\pi^{\mathrm{mVc}} \propto |y_i - p(\mathbf{x}_i; \hat{\boldsymbol{\beta}}_{\mathrm{MLE}})| \|\boldsymbol{x}_i\|,$$

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

1. **Covariate information represented by $\|\boldsymbol{x}_i\|$:**
   - larger values of $\|\boldsymbol{x}_i\|$ indicates larger re-sampling probabilities.

2. **Discrimination difficulty represented by $|y_i - p(\mathbf{x}_i; \hat{\boldsymbol{\beta}}_{\mathrm{MLE}})|$**
   - If $y_i = 0$;
   $$\pi^{\mathrm{mVc}} \propto p(\mathbf{x}_i; \hat{\boldsymbol{\beta}}_{\mathrm{MLE}}) \|\boldsymbol{x}_i\|$$
   - If $y_i = 1$
   $$\pi^{\mathrm{mVc}} \propto \{1 - p(\mathbf{x}_i; \hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\} \|\boldsymbol{x}_i\|$$
   - This echos the result of Silvapulle (JRSSB 1981)[1].

[1] On the existence of maximum likelihood estimators for the binomial response models. J. R. Stat. Soc. Ser. B. Stat. Methodol., 43(3):310-313, 1981.

Two step algorithms

Step 1 Take a random subsample of size r_0 to obtain an pilot estimate $\tilde{\boldsymbol{\beta}}_0$, using either the uniform subsampling or case-control subsampling. Replace $\hat{\boldsymbol{\beta}}_{\text{MLE}}$ with $\tilde{\boldsymbol{\beta}}_0$ to get an approximate optimal SSP.

Step 2 Subsample with replacement for a subsample of size r with the approximate optimal SSP calculated in Step 1. Obtain the estimate $\breve{\boldsymbol{\beta}}$ based on the total subsample of size $r_0 + r$.

Theoretical results

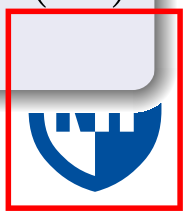
Theorem

Assume the covariate distribution satisfies that $\mathrm{E}(\mathbf{x}\mathbf{x}^T)$ *is positive definite and* $\mathrm{E}(e^{\mathbf{a}^T\mathbf{x}}) < \infty$ *for any* $\mathbf{a} \in \mathbb{R}^d$. *Let* $r_0/\sqrt{r} \to 0$. *As* $r_0 \to \infty$, $r \to \infty$ *and* $n \to \infty$, *conditional on* $\mathcal{F}_n$ *and* $\tilde{\boldsymbol{\beta}}_0$,

$$\mathbf{V}^{-1/2}(\breve{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{\mathrm{MLE}}) \longrightarrow N(0, \mathbf{I})$$

in distribution, in which $\mathbf{V} = \mathbf{M}_X^{-1}\mathbf{V}_c\mathbf{M}_X^{-1}$ *with* $\mathbf{V}_c$ *having the expression of*

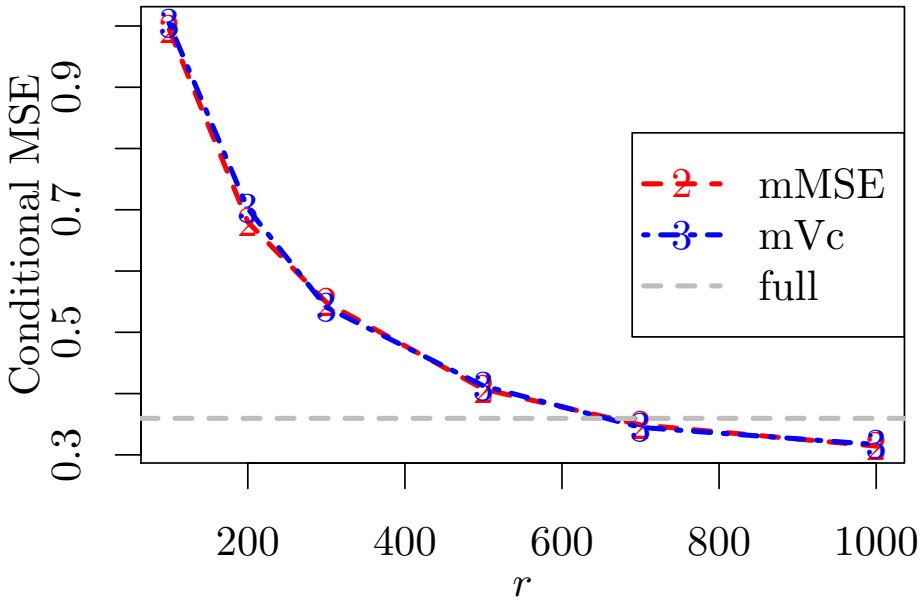
$$\mathbf{V}_c = \frac{1}{rn^2}\sum_{i=1}^{n}|y_i - p_i(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})|\|\mathbf{x}_i\|\sum_{i=1}^{n}\frac{|y_i - p_i(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})|\mathbf{x}_i\mathbf{x}_i^T}{\|\mathbf{x}_i\|}. \tag{11}$$

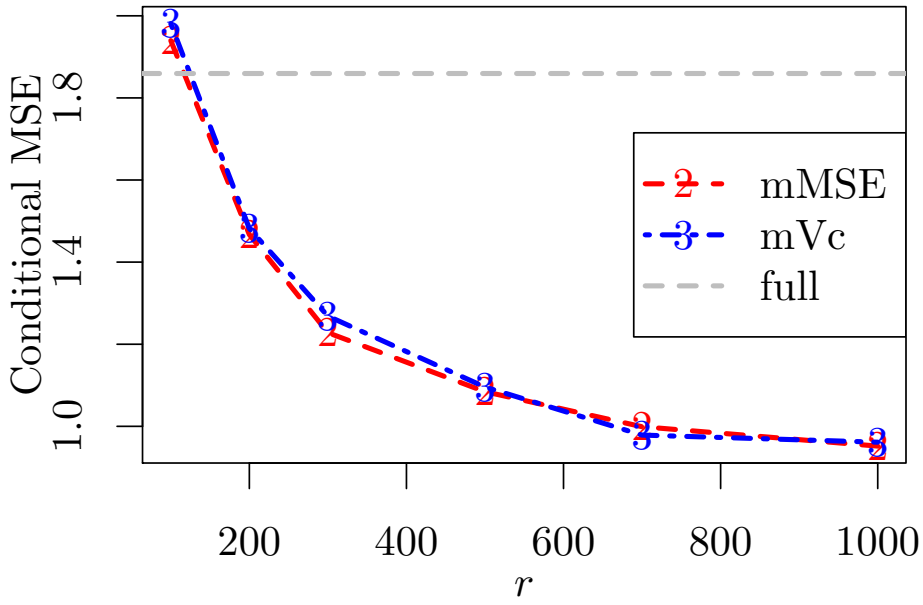


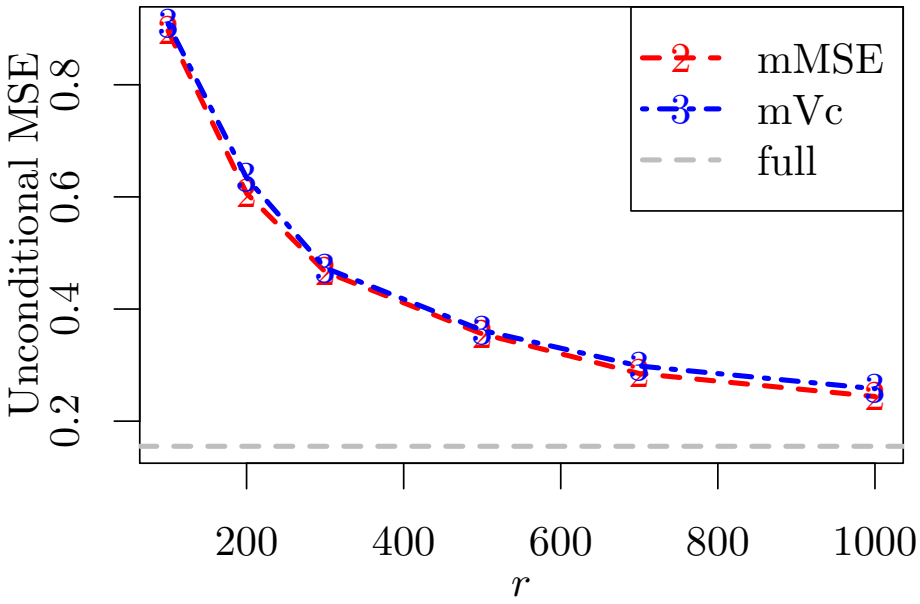
Simulation

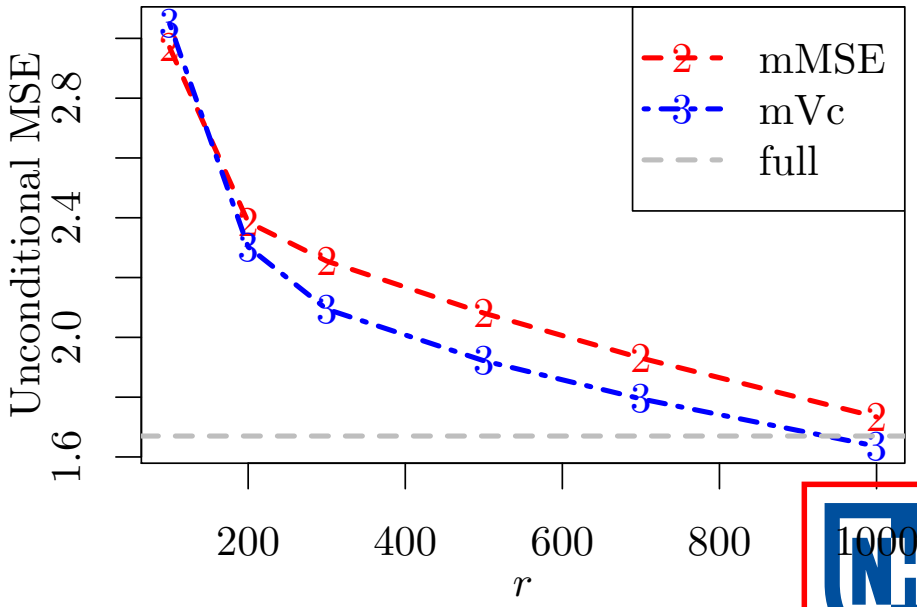
- Data of size $n = 10,000$ are generated from a logistic model (19) with $\boldsymbol{\beta}_0$ being a 7×1 vector of 0.5.
- $\mathbf{x}$ follows a multivariate normal distribution, $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}_{ij} = 0.5^{I(i \neq j)}$ and $I()$ is the indicator function.
- Consider two values of $\boldsymbol{\mu}$:
  1) $\boldsymbol{\mu} = \mathbf{-2.14}$ so that 1.01% of the responses are 1's.
  2) $\boldsymbol{\mu} = \mathbf{-2.9}$ so that 0.14% of the responses are 1's.
- We calculate MSEs for $\breve{\boldsymbol{\beta}}$ with different second stage subsample size r and a fixed first stage subsample size $r_0 = 200$, using

$$\text{MSE} = \frac{1}{S} \sum_{s=1}^{S} \|\breve{\boldsymbol{\beta}}^{(s)} - \boldsymbol{\beta}_0\|^2.$$









(a) 1.01% of y_i's are 1

(b) 0.14% of y_i's are 1

Computing time for large data

Table: CPU seconds with $r_0 = 200$, $r = 1000$ and different full data size n when the covariates are from a $d = 50$ dimensional normal distribution.

Method	n			
	10^4	10^5	10^6	10^7
mMSE	0.050	0.270	3.290	37.730
mVc	0.030	0.070	0.520	6.640
Uniform	0.010	0.030	0.020	0.830
Full	0.150	1.710	16.530	310.450

Particle physics data

Consider a supersymmetric (SUSY) benchmark data set (Baldi *et al.* 2014[2]). The sample size $n = 5,000,000$ and the data file is 2.4GB.

- The goal is to distinguish between a process where new supersymmetric particles are produced and a background process.
- There 18 features in the data set.
  - $x_1 - x_8$ are kinematic properties measured by the particle detectors in the accelerator
  - $x_9 - x_{18}$ are functions of the first 8 features that are high-level features derived by physicists to help discriminate between the two classes.

[2] P. Baldi, P. Sadowski, and D. Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, 5(4308), 2014.

Proportions of correct classification

- We draw sub-samples of size 1000 with $r_0 = 200$ and $r = 800$, from the training set of 4,500,000 observations.
- Use the sub-samples to calculate $\breve{\boldsymbol{\beta}}$ which is then used to classify the validation set of $500,000$.
- We calculate the areas under the ROC curves (AUC), which is a measure of the performance of a classifier.

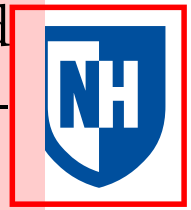
Table: Average AUC for the SUSY data set based on 1000 subsamples. A number in the parentheses is the associated standard error of the 1000 AUCs.

Method	AUC (SE)
uniform	0.8506 (0.0029)
mMSE	0.8508 (0.0030)
mVc	0.8517 (0.0025)
Full	0.8575

Comparisons with deep learning (DL)

- The deep learning (DL) method (Baldi *et al.* 2014) produced an AUC of **0.88**.
- Our methods give AUCs about **0.85**.
- Baldi *et al.*'s (2014) method requires special computing resources and coding skills
- Anyone with basic programming ability are able to implement our method.

Our method	DL
$r_0 + r = 1000$	$n = 4,500,000$
Logistic model	A five-layer neural nets with 300 hidden units in each layer
R with standard Newton's method	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



Questions

1. How to adjust the method if covariates $\mathbf{x}$ are measured with errors?
2. How to adjust the method if some response are misclassified?
3. **The real question: do these problems worth to be investigate? Are there real big data with measurement errors?**

Outline

Model setup

Assume the linear regression model,

$$y_i = \beta_0 + \sum_{j=1}^{p} z_{ij}\beta_j + \varepsilon_i, \ \ i = 1, ..., n, \tag{12}$$

where

- y_i are responses, $\mathbf{z}_i = (z_{i1}, ..., z_{ip})^{\mathrm{T}}$ are covariate vectors;
- β_0 is the scalar intercept parameter;
- $\boldsymbol{\beta}_1 = (\beta_1, \beta_2, ..., \beta_p)^{\mathrm{T}}$ is the slope vector;
- ε_i are uncorrelated error terms with mean 0 and variance σ^2.

Denote

- $\boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_1^{\mathrm{T}})^{\mathrm{T}}$
- $\mathbf{x}_i = (1, \mathbf{z}_i^{\mathrm{T}})^{\mathrm{T}}$
- $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$
- $\mathbf{y} = (y_1, ..., y_n)$

Existing methods based on Subsampling

Subsampling-based methods use nonuniform sampling probabilities so that influential data points are sampled with high probabilities.

- Let $\boldsymbol{\pi} = \{\pi_1, ..., \pi_n\}$ be nonuniform sampling probabilities for the full data such that $\sum_{i=1}^{n} \pi_i = 1$.
- Take a random subsample according to $\boldsymbol{\pi}$, say, $(\mathbf{x}_1^*, y_1^*)$, ..., $(\mathbf{x}_k^*, y_k^*)$.
- A subsampling-based estimator has a general form of

$$\tilde{\boldsymbol{\beta}}_L = \left(\sum_{i=1}^{k} w_i^* \mathbf{x}_i^* \mathbf{x}_i^{*\mathrm{T}} \right)^{-1} \sum_{i=1}^{k} w_i^* \mathbf{x}_i^* y_i^*, \tag{13}$$

  where the weight w_i^* is often taken to be $1/\pi_i^*$.

For the popular leveraging method (**LEV**), π_i's are the statistical leverage scores.

The IBOSS framework

Let δ_i be the indicator that observation $(\mathbf{z}_i, y_i)$ is included in a subdata. Then the information matrix of a subdata of size k is

$$\mathbf{M}(\boldsymbol{\delta}) = \frac{1}{\sigma^2}\sum_{i=1}^{n}\delta_i \mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}. \tag{14}$$

where $\boldsymbol{\delta} = \{\delta_1, \delta_2, ..., \delta_n\}$ such that $\sum_{i=1}^{n}\delta_i = k$.

- It is the inverse of the variance covariance matrix of $\hat{\boldsymbol{\beta}}_{\boldsymbol{\delta}}$, namely,

$$\mathrm{V}(\hat{\boldsymbol{\beta}}_{\boldsymbol{\delta}}|\mathbf{X}) = \mathbf{M}(\boldsymbol{\delta})^{-1} \tag{15}$$

- In order to have an optimal estimator based on a subdata, we choose a $\boldsymbol{\delta}$ that "maximizes" $\mathbf{M}(\boldsymbol{\delta})$.

The IBOSS framework

- $\mathbf{M}(\boldsymbol{\delta})$ is a matrix, so the meaning of "maximization" has to be defined.
- We adopt the idea of optimal experimental designs (Kiefer, 1959), and use a convex function of $\mathbf{M}(\boldsymbol{\delta})$ to induce a complete ordering.
- If ψ is a specific convex function, then we want to find subdata with indicator $\boldsymbol{\delta}^{opt}$ so that

$$\boldsymbol{\delta}^{opt} = \arg\max_{\boldsymbol{\delta}} \psi\{\mathbf{M}(\boldsymbol{\delta})\} \quad \text{subject to} \quad \sum_{i=1}^{n} \delta_i = k. \tag{16}$$

- For an IBOSS subdata, the LS estimator is still the BLUE because the responses remain uncorrelated.
- Inferences will however be more efficient than with subsampling.

A lower bound for the subsampling appro

Theorem

Let $\tilde{\boldsymbol{\beta}}_L$ *be an estimator from the subsampling approach. Denote* $\Delta = \{\boldsymbol{\delta}_L : \sum_{i=1}^{n} \delta_{Li}\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}$ *is non-singular*$\}$. *Then given* $\boldsymbol{\delta}_L \in \Delta$, $\tilde{\boldsymbol{\beta}}_L$ *is unbiased for* $\boldsymbol{\beta}$, *and*

$$\mathrm{V}(\tilde{\boldsymbol{\beta}}_L|\mathbf{X}, \boldsymbol{\delta}_L \in \Delta) \geq P(\boldsymbol{\delta}_L \in \Delta|\mathbf{X}) \left[\mathrm{E}\{\mathbf{M}(\boldsymbol{\delta}_L)|\mathbf{X}\}\right]^{-1}$$

$$= \frac{\sigma^2 P(\boldsymbol{\delta}_L \in \Delta|\mathbf{X})}{k} \left\{\sum_{i=1}^{n} \pi_i \mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\right\}^{-1} \tag{17}$$

in the Loewner ordering.

Moreover, the inequality holds regardless whether the subsampling is with or without replacement.

IBOSS subdata based on the D-optimalit

For given full data and a subdata size k, the D-optimality suggests the selection of subdata so that

$$\boldsymbol{\delta}_D^{opt} = \arg\max_{\boldsymbol{\delta}} |\mathbf{M}(\boldsymbol{\delta})| = \arg\max_{\boldsymbol{\delta}} \left| \sum_{i=1}^{n} \delta_i \mathbf{x}_i \mathbf{x}_i^{\mathrm{T}} \right|. \tag{18}$$

- Maximizing $|\mathbf{M}(\boldsymbol{\delta})|$ is equivalent to minimizing the volume of the joint confidence ellipsoid for all unknown parameters.
- For the D-optimality criterion, it is difficult to get a closed-form solution in general.
- We first derive an upper bound for $|\mathbf{M}(\boldsymbol{\delta})|$, and then propose an algorithm to approximate this upper bound.

An upper bound for the determinant

Theorem (D-optimality)

For any subdata with size k,

$$|\mathbf{M}(\boldsymbol{\delta})| \leq \frac{k^{p+1}}{4^p} \prod_{j=1}^{p} (z_{(n)j} - z_{(1)j})^2, \tag{19}$$

where

- $z_{(n)j} = \max\{z_{1j}, z_{2j}, ..., z_{nj}\}$ *and* $z_{(1)j} = \min\{z_{1j}, z_{2j}, ..., z_{nj}\}$,
- *i.e., they are the nth and first order statistics of* $z_{1j}, z_{2j}, ..., z_{nj}$.

If the subdata consists of the 2^p *points* $(a_1, \ldots, a_p)^{\mathrm{T}}$ *where* $a_j = z_{(n)j}$ *or* $z_{(1)j}$, $j = 1, 2, ..., p$, *each occurring equally often, then the equality holds.*

An approximation algorithm

Algorithm (D-optimality motivated IBOSS algorithm)

Suppose that $r = k/(2p)$ is an integer. Using a partition-based selection algorithm, perform the following steps:

1. *For z_{i1}, $1 \le i \le n$, include* **r** *data points with the* **r smallest** z_{i1} **values** *and* **r** *data points with the* **r largest** z_{i1} **values** *in the subdata;*
2. *For $j = 2, ..., p$, exclude data points that were previously selected, and from the remainder select* **r** *data points with the* **r smallest** z_{ij} **values** *and* **r** *sample points with the* **r largest** z_{ij} **values** *for the subdata;*
3. *For the selected subdata $(\mathbf{X}_D^*, \mathbf{y}_D^*)$, calculate*

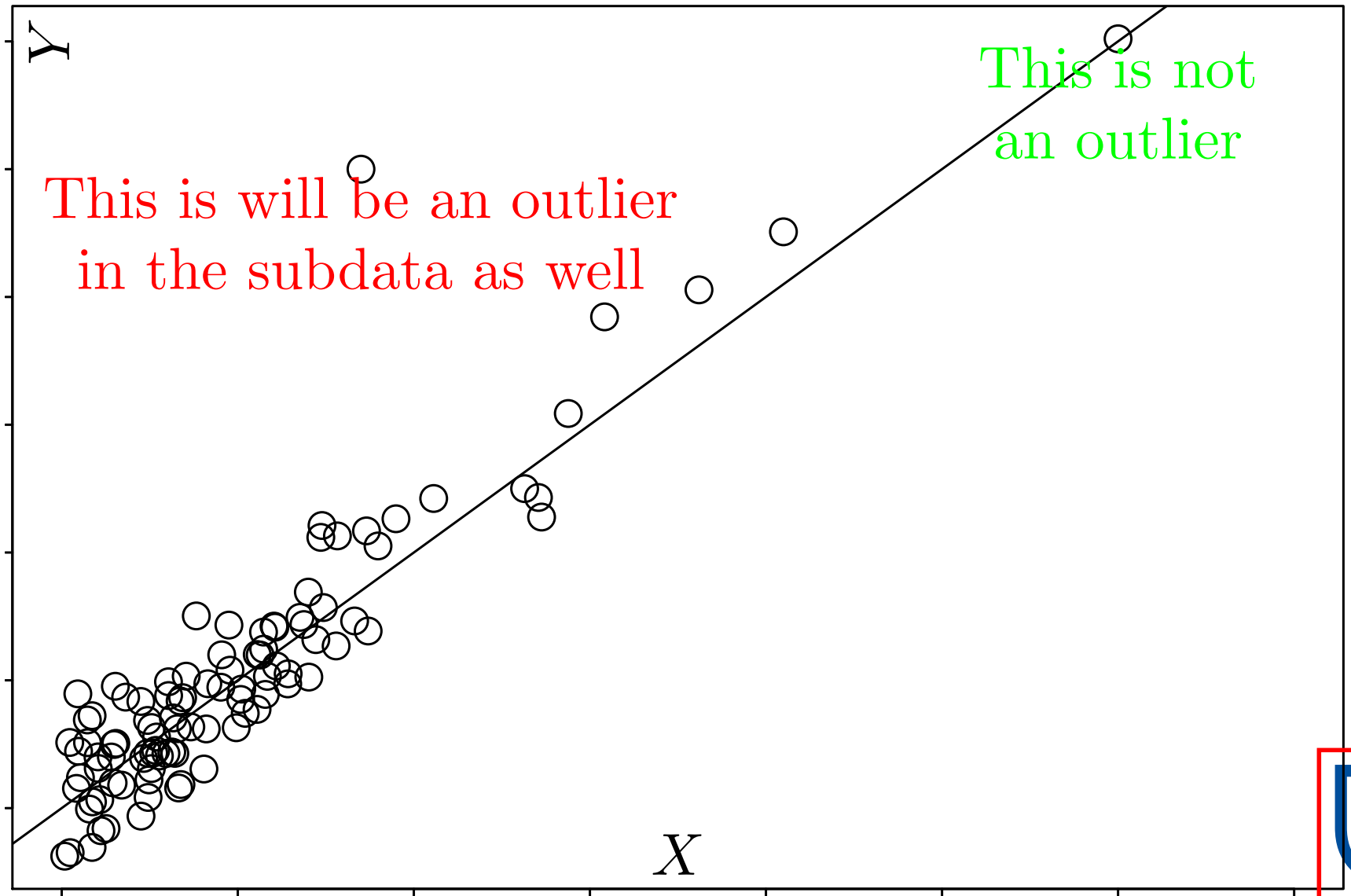
$$\hat{\boldsymbol{\beta}}^D = \{(\mathbf{X}_D^*)^{\mathrm{T}}\mathbf{X}_D^*\}^{-1}(\mathbf{X}_D^*)^{\mathrm{T}}\mathbf{y}_D^*, \qquad \widehat{\mathrm{V}}(\hat{\boldsymbol{\beta}}^D) = \hat{\sigma}^2\{(\mathbf{X}_D^*)^{\mathrm{T}}\mathbf{X}_D^*\}^{-1},$$

and, if needed, statistics for assessing model fit.

Remarks

- A partition-based selection algorithm has an average time complexity of $O(n)$ to find the r largest (or smallest) values.
- The time complicity of this algorithm is $O(np + kp^2)$.
- For the scenario that $n > kp$, time complicity is $O(np)$.
- This algorithm is very suitable for parallel computing.
- This algorithm gives the variance covariance matrix of the resultant estimator as well, which is very crucial for statistical inference.

About outliers



Asymptotic result for $\hat{\boldsymbol{\beta}}^D$

Assume that $n \to \infty$, k and p are fixed, and $\mathbf{z}_i$'s are i.i.d.

Theorem

If that covariate distributions are in the domain of attraction of the generalized extreme value distribution, and $\liminf_{n\to\infty} \lambda_{\min}(\mathbf{R}^D) > 0$, *then,*

$$\mathrm{V}(\hat{\beta}_0^D|\mathbf{X}) \asymp_P 1 \tag{20}$$

$$\mathrm{V}(\hat{\beta}_j^D|\mathbf{X}) \asymp_P \frac{\mathbf{1}}{(\mathbf{z_{(n)j}} - \mathbf{z_{(1)j}})^{\mathbf{2}}}, \quad j = 1, ..., p, \tag{21}$$

where $\mathbf{R}^D$ *is the correlation matrix of* $\mathbf{X}_D^*$
and $A \asymp_P B$ *means* $A = O_P(B)$ *and* $B = O_P(A)$.

If $\mathbf{z}_i \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} = \boldsymbol{\Phi}\boldsymbol{\rho}\boldsymbol{\Phi}$ and $\boldsymbol{\Phi} = \text{diag}(\sigma_1, ..., \sigma_p)$.

$$\text{V}\left(\hat{\boldsymbol{\beta}}^D|\mathbf{X}\right) = \begin{bmatrix} \frac{\sigma^2}{k} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\log n}\frac{p\sigma^2}{2k}(\boldsymbol{\Phi}\boldsymbol{\rho}^2\boldsymbol{\Phi})^{-1} \end{bmatrix} + O_P \begin{bmatrix} \frac{1}{\sqrt{\log n}} & \frac{1}{\log n} \\ \frac{1}{\log n} & \frac{1}{(\log n)^{3/2}} \end{bmatrix}.$$

If $\mathbf{z}_i \sim \text{Lognormal}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$,

$$\text{V}(\hat{\boldsymbol{\beta}}^D|\mathbf{X}) = \mathbf{A}_n \begin{bmatrix} \frac{2\sigma^2}{k} & \frac{2\sigma^2}{k}\mathbf{u}^{\text{T}} \\ \frac{2\sigma^2}{k}\mathbf{u} & \frac{2\sigma^2}{pk}\boldsymbol{\Lambda} + \frac{2\sigma^2}{k}\mathbf{u}\mathbf{u}^{\text{T}} \end{bmatrix} \mathbf{A}_n\{1 + o_P(1)\}$$

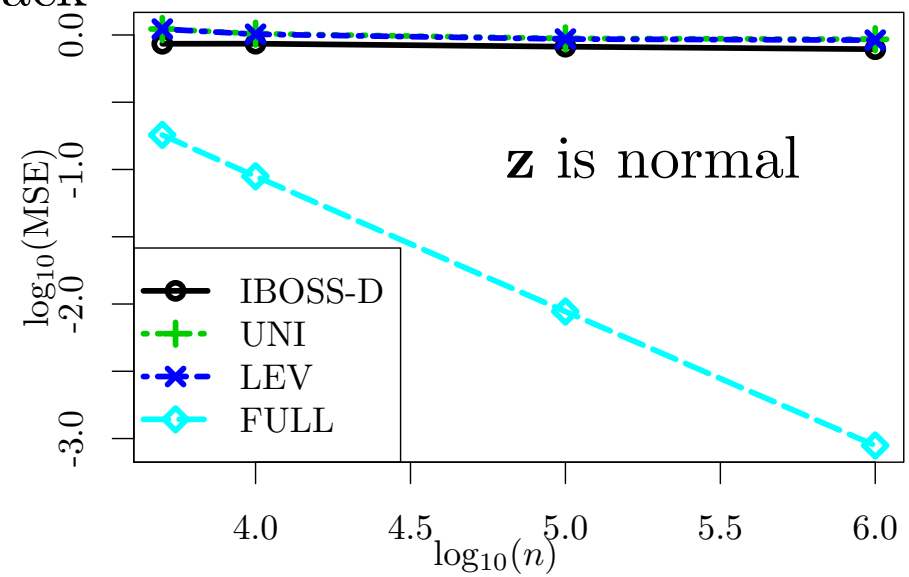
$\mathbf{A}_n = \text{diag}(1, e^{-\sqrt{2\log n}\sigma_1}, \ldots, e^{-\sqrt{2\log n}\sigma_p})$, $\mathbf{u} = \text{diag}(e^{-\mu_1}, ..., e^{-\mu_p})^{\text{T}}$, and $\boldsymbol{\Lambda} = \text{diag}(e^{-2\mu_1}, ..., e^{-2\mu_p})$.

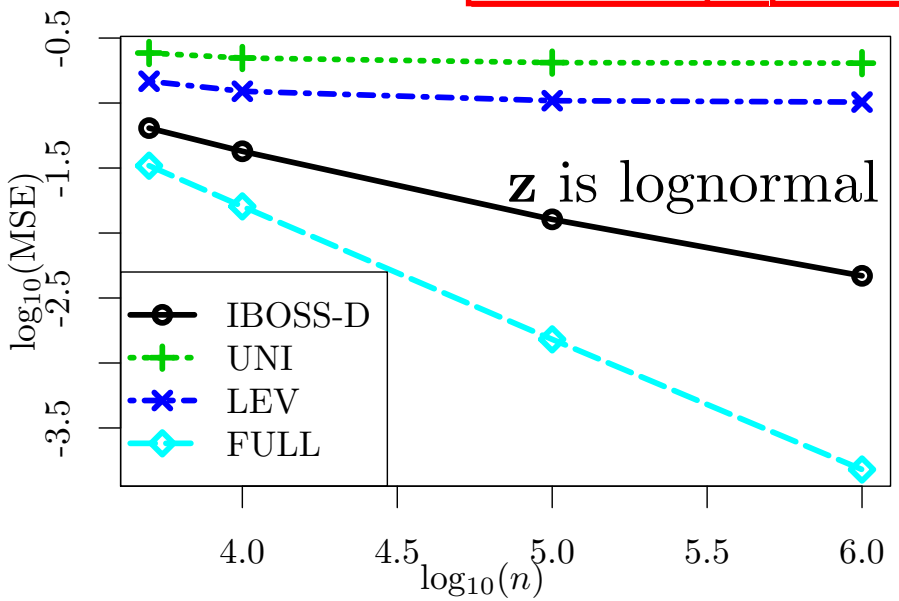
Simulation setup

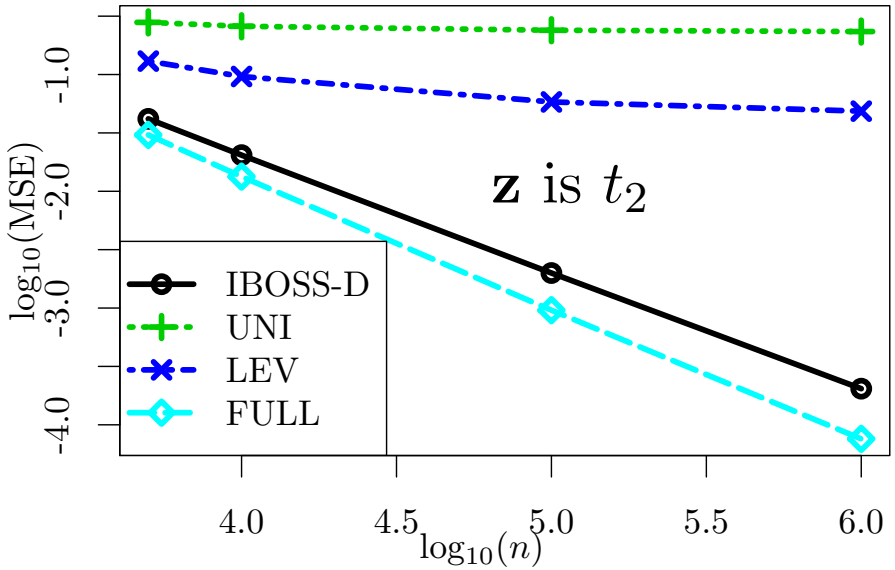
- $p = 50$, $\boldsymbol{\beta} = \mathbf{1}_{51\times 1}$, $\varepsilon_i \sim N(0, \sigma^2)$ with $\sigma^2 = 9$.
- $\mathbf{z}_i$'s are generated from the following distributions.
  1. **Normal**, $N(\mathbf{0}, \boldsymbol{\Sigma})$;
  2. **Lognormal**, $\exp\{N(\mathbf{0}, \boldsymbol{\Sigma})\}$;
  3. $\mathbf{t_2}$, $t_2(\mathbf{0}, \boldsymbol{\Sigma})$;
  4. **Mixture** of $N(\mathbf{1}, \boldsymbol{\Sigma})$, $t_2(\mathbf{1}, \boldsymbol{\Sigma})$, $t_3(\mathbf{1}, \boldsymbol{\Sigma})$, Unif$[\mathbf{0}, \mathbf{2}]$ and $\exp\{N(\mathbf{0}, \boldsymbol{\Sigma})\}$ with equal proportions.
- The simulation was repeated $S = 1000$ times.
- Empirical mean squared errors (MSE) are compared.

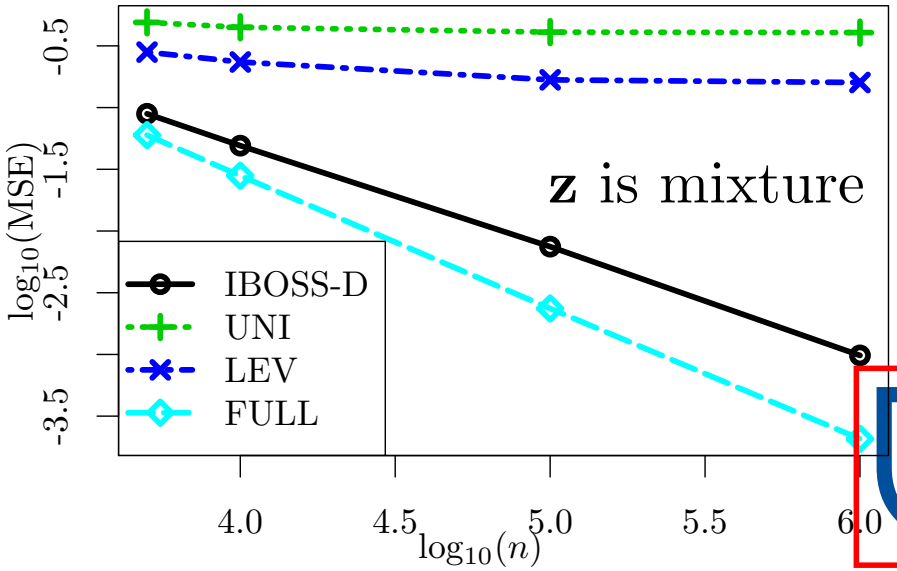
MSE of the slope estimator with k = 10

Back



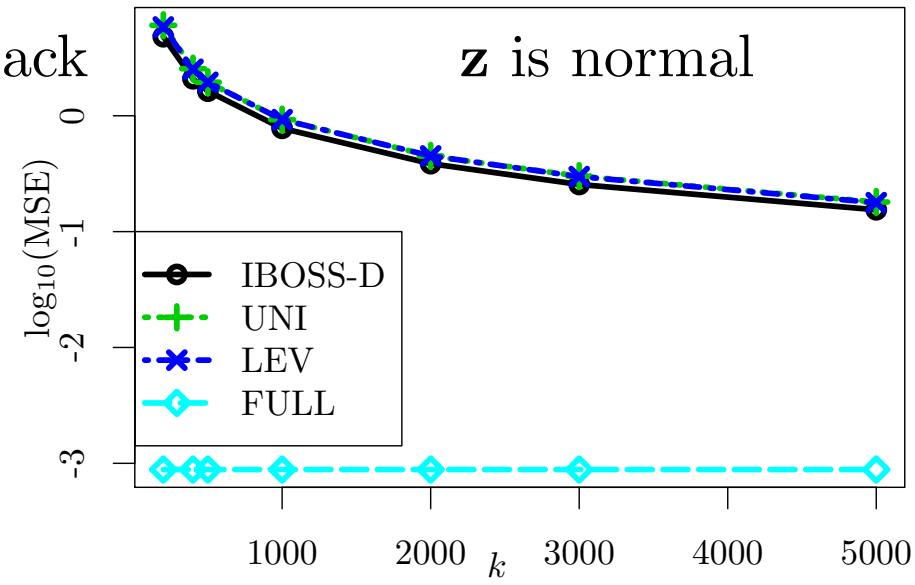


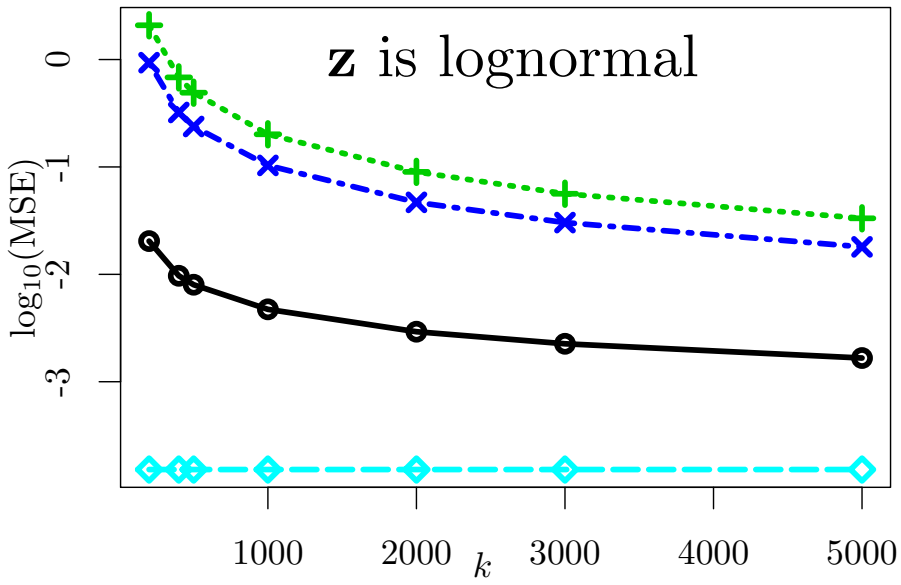


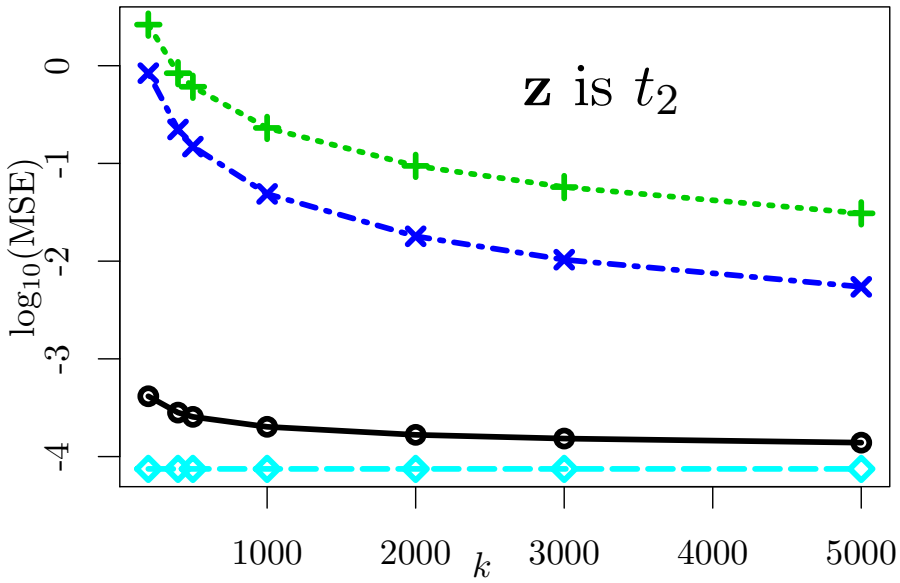


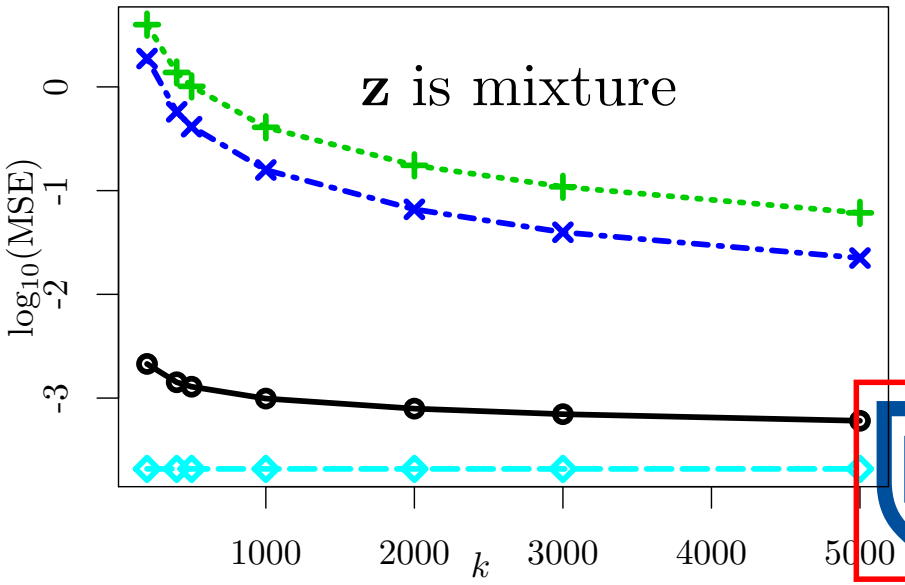
MSE of the **slope** estimator with **n = 10**

Back









Some highlights

For the mixture covariate distribution:

- The IBOSS strategy with $k = 1000$ and $n = 10^6$ is about 2.4 times as accurate as the full data analysis with $n = 10^5$.
- When $n = 10^6$, the IBOSS approach with $k = 200$ is about 10 times as accurate the Leveraging method with $k = 5000$.

CPU times for different n, p combination

Table: CPU times (seconds) for different n with $p = 500$

n	IBOSS-D	UNI	LEV	FULL
5×10^3	1.19	0.33	0.88	1.44
5×10^4	1.36	0.29	2.20	13.39
5×10^5	8.89	0.31	21.23	132.04

Table: CPU times (seconds) for different p with $n = 5 \times 10^5$

p	IBOSS-D	UNI	LEV	FULL
10	0.19	0.00	1.94	0.21
100	1.74	0.02	4.66	6.55
500	9.30	0.31	21.94	132.47

Thank you!

August 16, 2016