Fast Optimal Subsampling Probability Approximation for Generalized Linear Models

JooChul Lee, Elizabeth D. Schifano, HaiYing Wang*

Department of Statistics, University of Connecticut, Storrs, CT 06269, USA

Abstract

For massive data, subsampling techniques are popular to mitigate computational burden by reducing the data size. In a subsampling approach, subsampling probabilities for each data point are specified to obtain an informative sub-data, and then estimates based on the sub-data are obtained to approximate estimates from the full data. Assigning subsampling probabilities based on minimization of the asymptotic mean squared error of the estimator from a general subsample (A-optimality criteria) is a popular approach, however, it is still computationally demanding to calculate the probabilities under this setting. To efficiently approximate the A-optimal subsampling probabilities for generalized linear models, randomized algorithms are proposed. To develop the algorithms, the Johnson-Lindenstrauss Transform and Subsampled Randomized Hadamard Transform are used. Additionally, optimal subsampling probabilities are derived for the Gaussian linear model in the case where both the regression coefficients and dispersion parameter are of interest, and algorithms are developed to approximate the optimal subsampling probabilities. Simulation studies indicate that the estimators based on the developed algorithms have excellent performance for statistical inference and have substantial savings in computing time compared to the direct calculation of the A-optimal subsampling probabilities.

Keywords: Generalized linear models, Massive data, Optimal subsampling, Randomized algorithm.

1. Introduction

Due to scientific and technological advances, large datasets are being collected across many fields and require proper analysis. Applying conventional statistical methods to such big data can strain both computer memory and computational efficiency, with even very simple tasks causing inordinate computational burden. There are several statistical and computational approaches to address this challenge: divide-and-conquer approach [e.g., 10, 5, 15], online updating approach [e.g., 12, 13, 14, 20, 21, 9], and subsampling-based approach [e.g., 7, 11, 18, 1, 16, 17].

*Corresponding author: Department of Statistics, University of Connecticut, 215 Glenbrook Road U4120, Storrs, CT 06269, USA; Tel: 1.860.486.3414; Fax: 1.860.486.4113.

Email address: haiying.wang@uconn.edu (HaiYing Wang)

[1]Supplementary materials for this article are attached as annexes.

[2]Codes are available on GitHub, https://github.com/pedigree07/FASA.

The subsampling-based approach involves using sub-data which are drawn from the full data to approximate results of interest from the full data. Clearly, using a subset of the full data can lessen the computational burden by reducing the data size. The main issue of this approach is how to specify subsampling probabilities to obtain informative data points. In the context of the linear regression model, Drineas et al. [7] developed an algorithm to approximate the least squares estimates by preprocessing a randomized Hadamard transform on the covariate matrix and then sampling uniformly at random, and Ma et al. [11] used statistical leverage scores of the covariate matrix to allocate the subsampling probabilities. For logistic regression, Wang et al. [18] developed an optimal subsampling procedure. They derived an asymptotic distribution of the general subsampling estimator and then obtained optimal subsampling probabilities based on an A- or L-optimality criterion. The A-optimality criterion seeks to minimize the trace of the variance-covariance matrix of the parameter estimator, and the L-optimality criterion seeks to minimize the trace of the variance-covariance matrix based on some linear transformation of the parameter estimator. Based on the optimal subsampling algorithm developed by Wang et al. [18], Ai et al. [1] considered optimal subsampling for generalized linear models (GLMs) and Wang and Ma [17] investigated optimal subsampling for quantile regression.

In this paper, we propose algorithms to approximate the optimal subsampling probabilities under GLMs. Computing time for subsampling probabilities obtained from the A-optimality criterion is $O(Np^2)$ where N and p are the full data size and the number of covariates, respectively. Thus, efficient algorithms for approximating subsampling probabilities are developed to reduce the computing time. We use a Johnson-Lindenstrauss Transform (JLT) and a Subsampled Randomized Hadamard Transform (SRHT) which are techniques to downsize matrix volume. Furthermore, we derive A-optimal subsampling probabilities for the Gaussian linear model when the dispersion parameter is also of interest. The asymptotic distribution of the subsampling estimators is established, and optimal subsampling probabilities are obtained using this asymptotic distribution. We also suggest randomized algorithms to approximate these subsampling probabilities.

The rest of this paper is organized as follows. In Section 2, we explain the optimal subsampling probabilities based on the A-optimality criterion for GLMs, and propose the optimal subsampling probabilities for the Gaussian linear model considering the coefficients and dispersion parameter together. In Section 3, we develop algorithms to approximate optimal subsampling probabilities using a JLT and SRHT. Simulation studies and two real data analyses are provided in Section 4 to demonstrate the empirical performance and applicability of our algorithms. Section 5 concludes the paper and technical proofs for theoretical results are provided in Supplementary materials.

2. Models and Optimal Subsampling Probabilities

In this section, we review optimal subsampling probabilities developed for approximating the maximum likelihood estimator (MLE) of regression coefficients from the full data under GLMs, and derive optimal subsampling probabilities for the Gaussian linear model when the dispersion parameter is additionally of interest. In the following presentation, denote the full data as $\mathcal{D}_N = \{(\mathbf{x}_i, y_i) : i = 1, ..., N\}$ from a model where $\mathbf{x}_i$ is a p dimensional vector of covariates whose first element is one corresponding to an intercept term, and y_i is the response for the ith observation. Let $\pi_1, ..., \pi_N$ be subsampling probabilities assigned to all observations such that $\sum_{i=1}^{N} \pi_i = 1$. Using subsampling with replacement, a random subsample of size n is drawn based on the subsampling probabilities $\{\pi_i\}_{i=1}^{N}$ from the full data. Denote $\mathbf{x}_i^*, y_i^*$ and π_i^* for $i = 1, ..., n$ as covariates, responses, and subsampling probabilities in the subsample, respectively.

2.1. Optimal Subsampling probability with known dispersion parameter in generalized linear models

Suppose that y comes from a distribution with the following mass or density function in the subclass of the general exponential family,

$$f(y|\theta,\phi) = g(y)\exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right), \tag{1}$$

where θ is a natural parameter and $\phi(>0)$ is a dispersion parameter.

Let y be the response variable and $\mathbf{x}$ be the p dimensional covariate. Considering θ as a function of the linear predictor, $\theta = u(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x})$, and assuming that the dispersion parameter, ϕ, is known, Ai et al. [1] studied optimal subsampling under the following generalized linear regression model,

$$f(y|\boldsymbol{\beta},\mathbf{x}) = h(y)\exp[yu(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}) - \psi\{u(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x})\}], \tag{2}$$

where $\boldsymbol{\beta}$ is a $p\times 1$ vector of unknown regression coefficients including the intercept. For GLMs, they derived asymptotic distributions of the subsampling-based estimator, and then developed optimal subsampling strategies based on A- and L-optimality criteria. For completeness, we now briefly review the subsampling probabilities based on the A-optimality criterion for GLMs.

Suppose the full data $\mathcal{D}_N$ is from the model (2). The subsample estimator $\tilde{\boldsymbol{\beta}}_G$ is obtained by maximizing the following weighted objective function,

$$l^*(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\pi_i^*}[y_i^* u(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^*) - \psi\{u(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^*)\}]. \tag{3}$$

Let $\dot{\psi}(t)$ and $\dot{u}(t)$ be the first derivatives of $\psi(t)$ and $u(t)$, and $\ddot{\psi}(t)$ and $\ddot{u}(t)$ be the second derivatives of $\psi(t)$ and $u(t)$, respectively. Let $\hat{\boldsymbol{\beta}}_G$ be the MLE of $\boldsymbol{\beta}$ based on the full data under model (2). Given the full data, $\sqrt{n}(\tilde{\boldsymbol{\beta}}_G - \hat{\boldsymbol{\beta}}_G)$ converges in distribution to a normal distribution with mean $\mathbf{0}$ and variance-covariance matrix $\mathbf{V}_G = \mathbf{M}_G^{-1}\mathbf{V}_G^c\mathbf{M}_G^{-1}$, where

$$\mathbf{V}_G^c = \frac{1}{N^2}\sum_{i=1}^{N}\frac{w_i^c\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\pi_i}, \mathbf{M}_G = \frac{1}{N}\sum_{i=1}^{N}w_i\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}, w_i^c = [y_i - \dot{\psi}\{u(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)\}]^2\dot{u}(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i),$$

and $w_i = \ddot{u}(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)[\dot{\psi}\{u(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)\} - y_i] + \ddot{\psi}\{u(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)\}\dot{u}^2(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)$. To determine the optimal subsampling probabilities based on the A-optimality criterion, Ai et al. [1] minimized the asymptotic mean squared error (MSE) of $\tilde{\boldsymbol{\beta}}_G$, which is the same idea proposed in Wang et al. [18]. The resulting subsampling probabilities are

$$\pi_i^G = \frac{|y_i - \dot{\psi}\{u(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)\}|\|\mathbf{M}_G^{-1}\dot{u}^2(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)\mathbf{x}_i\|}{\sum_{j=1}^{N}|y_j - \dot{\psi}\{u(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_j)\}|\|\mathbf{M}_G^{-1}\dot{u}(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_j)\mathbf{x}_j\|}, \quad i = 1,...,N. \tag{4}$$

2.2. Optimal subsampling probability in Gaussian linear model

In this subsection, we derive the optimal subsampling probabilities based on A-optimality in the Gaussian linear model when ϕ is also a parameter of interest. To describe the Gaussian model from (1), we assume that y follows a normal distribution with mean μ and variance σ^2, and the

canonical link function (identity link) is used, namely, $\mu = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}$. The conditional distribution of y given $\mathbf{x}$ is

$$f(y|\mathbf{x}, \boldsymbol{\beta}, \sigma) = \exp\left\{ \frac{y\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x} - (\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x})^2/2}{\sigma^2} - \left(\frac{y^2}{\sigma^2} + \frac{\log 2\pi\sigma^2}{2} \right) \right\}. \tag{5}$$

Suppose the full data $\mathcal{D}_N$ is from the model (5). Then, the subsample estimators, $(\tilde{\boldsymbol{\beta}}_L^{\mathrm{T}}, \tilde{\sigma}_L)^{\mathrm{T}}$, are obtained by maximizing the weighted objective function,

$$l_L^*(\boldsymbol{\beta}, \sigma) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\pi_i^*} \left(\frac{y_i^* \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^* - (\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^*)^2/2}{\sigma^2} - \left(\frac{(y_i^*)^2}{\sigma^2} + \frac{\log 2\pi\sigma^2}{2} \right) \right). \tag{6}$$

To establish the asymptotic properties of the subsample estimator, we need the following assumptions.

Assumption 1. $N^{-2}\sum_{i=1}^{N} \pi_i^{-1}\|\mathbf{x}_i\|^4 = O_P(1)$ *and* $N^{-2}\sum_{i=1}^{N} \pi_i^{-1} e_i^4 = O_P(1)$ *where* $e_i = y_i - \hat{\boldsymbol{\beta}}_L^{\mathrm{T}}\mathbf{x}_i$ *and* $\hat{\boldsymbol{\beta}}_L$ *is the MLE of* $\boldsymbol{\beta}$ *based on the full data under model* (5).

Assumption 2. $N^{-(2+\delta)}\sum_{i=1}^{N} \pi_i^{-(1+\delta)}\|\mathbf{x}_i\|^{2(2+\delta)} = O_P(1)$ *and* $N^{-(2+\delta)}\sum_{i=1}^{N} \pi_i^{-(1+\delta)}|e_i|^{2(2+\delta)} = O_P(1)$ *for some* $\delta > 0$.

Assumption 1 is a condition on subsampling probabilities and the covariate distribution. It essentially imposes some moment constraints. For example, with equal sampling probabilities $\pi_i = 1/N$, Assumption 1 holds if $\mathbf{E}(\mathbf{x}_i^4) < \infty$ and $\mathbf{E}(|y_i|^4) < \infty$. Assumption 2 is for the Lindeberg-Feller Central Limit Theorem, and a sufficient condition is that $\mathbf{E}\left(\mathbf{x}_i^{2(2+\delta)}\right) < \infty$ and $\mathbf{E}(|y_i|^{2(2+\delta)}) < \infty$ for equal sampling probabilities $\pi_i = 1/N$.

Theorem 1. *Under Assumptions 1 and 2, conditional on the full data* $\mathcal{D}_N$ *in probability,*

$$\begin{pmatrix} \tilde{\boldsymbol{\beta}}_L - \hat{\boldsymbol{\beta}}_L \\ \tilde{\sigma}_L - \hat{\sigma}_L \end{pmatrix} \longrightarrow N(0, \mathbf{V}), \tag{7}$$

in distribution as $n, N \to \infty$, *where* $\hat{\sigma}_L$ *is the MLE for* σ *based on the full data under model* (5), $\mathbf{V} = \mathbf{M}^{-1}\mathbf{V}^c\mathbf{M}^{-1}$, $\mathbf{M} = \begin{bmatrix} \sum_{i=1}^{N} \frac{\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{N} & \mathbf{0} \\ \mathbf{0} & 2\hat{\sigma}_L \end{bmatrix}$ *and* $\mathbf{V}^c = \frac{1}{nN^2}\sum_{i=1}^{N} \frac{1}{\pi_i} \begin{bmatrix} e_i^2\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} & (e_i^2 - \hat{\sigma}_L)e_i\mathbf{x}_i \\ (e_i^2 - \hat{\sigma}_L)e_i\mathbf{x}_i^{\mathrm{T}} & (e_i^2 - \hat{\sigma}_L^2)^2 \end{bmatrix}$.

In the following Theorem 2, the subsampling probabilities based on A-optimality criterion are obtained by minimizing the trace of $\mathbf{V}$.

Theorem 2. *The A-optimal subsampling probabilities that minimize the asymptotic mean squared error of* $(\tilde{\boldsymbol{\beta}}_L^{\mathrm{T}}, \tilde{\sigma}_L^{\mathrm{T}})^{\mathrm{T}}$ *are*

$$\pi_i^L = \frac{\sqrt{e_i^2\|\mathbf{M}_L^{-1}\mathbf{x}_i\|^2 + (e_i^2 - \hat{\sigma}_L^2)^2/(4N^2\hat{\sigma}_L^2)}}{\sum_{j=1}^{N}\sqrt{e_j^2\|\mathbf{M}_L^{-1}\mathbf{x}_j\|^2 + (e_j^2 - \hat{\sigma}_L^2)^2/(4N^2\hat{\sigma}_L^2)}} \quad i = 1, ..., N, \tag{8}$$

where $\mathbf{M}_L = \mathbf{X}^{\mathrm{T}}\mathbf{X}$ *and* $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^{\mathrm{T}}$.

3. Fast Approximation of Subsampling Probability

To compute the optimal subsampling probabilities in (4) and (8), $O(Np^2)$ time is needed because of $\|\mathbf{M}_G^{-1}\dot{u}(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)\mathbf{x}_i\|$ in (4) and $\|\mathbf{M}_L^{-1}\mathbf{x}_i\|$ in (8) for $i = 1, ..., N$. Thus, randomized algorithms to approximate them are proposed for saving the computing time by using a Fast Johnson-Lindenstrauss Transform (FJLT), and a Johnson-Lindenstrauss Transform (JLT). We call them **f**ast **A**-optimal **s**ubsampling probability **a**pproximation (FASA) algorithms. To construct a FJLT with high probability for any vector $z \in \mathbb{R}^N$, we can use a Subsampled Randomized Hadamard Transform (SRHT), $\mathbf{T}_1 = \frac{1}{\sqrt{r_1}}SHD$, where S is a $r_1 \times N$ linear sampling operator, H is the $N \times N$ Hadamard transform and D is a $N \times N$ diagonal matrix whose diagonal entries are +1 or -1 with probability 1/2 respectively, [e.g., 7]. The time complexity for performing $\mathbf{T}_1 z$ is $O(N \log r_1)$ [e.g., 3]. To construct a JLT with high probability for any vector $z^* \in \mathbb{R}^p$, we can use an $r_2 \times p$ matrix, denoted by $\mathbf{T}_2$, in which every entry is independently equal to $\pm\sqrt{3/r_2}$ with probability 1/6 each and zero with probability 2/3 [e.g., 2]. Then, $\mathbf{T}_2 z^*$ is performed in $O(r_2 p)$ time.

3.1. Subsampling probability approximation in generalized linear models

In this subsection, we approximate the subsampling probabilities in (4). Write $\mathbf{M}_G = \mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}$ where $\mathbf{W} = \mathrm{diag}(w_1, ..., w_N)$, and $\mathcal{U} = \mathrm{diag}[\dot{u}(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_1), ..., \dot{u}(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_N)]$. Then, $\|\mathbf{M}_G^{-1}\dot{u}(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)\mathbf{x}_i\|$ is expressed as $\|(\mathbf{M}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\|$ where $(\mathbf{M}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}$ is the ith column of $\mathbf{M}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U}$. We focus on approximating $\|(\mathbf{M}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\|$ for $i = 1, 2, ..., N$. Using the SRHT, we first approximate $\mathbf{M}_G$ as

$$\widetilde{\mathbf{M}}_G = (\mathbf{T}_1\mathbf{W}^{1/2}\mathbf{X})^{\mathrm{T}}\mathbf{T}_1\mathbf{W}^{1/2}\mathbf{X}, \tag{9}$$

where $\mathbf{W}^{1/2} = \mathrm{diag}(\sqrt{w_1}, ..., \sqrt{w_N})$. Since we consider the case where N is much larger than r_1 and p, computing $\widetilde{\mathbf{M}}_G$ takes $O(Np \log r_1)$ time. However, we still need $O(Np^2)$ time for $\widetilde{\mathbf{M}}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U}$. Thus, we further consider a JLT for $\widetilde{\mathbf{M}}_G^{-1}$ to reduce the computational burden. Based on the JLT for $\widetilde{\mathbf{M}}_G^{-1}$, say $\mathbf{T}_2\widetilde{\mathbf{M}}_G^{-1}$, we can use $(\mathbf{T}_2\widetilde{\mathbf{M}}_G^{-1})\mathbf{X}^{\mathrm{T}}\mathcal{U}$ instead of $\mathbf{M}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U}$. Then, $\|(\mathbf{M}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\|$ can be approximated as

$$\|(\mathbf{T}_2\widetilde{\mathbf{M}}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\|, \quad i = 1, ..., N. \tag{10}$$

This operation of $(\mathbf{T}_2\widetilde{\mathbf{M}}_G^{-1})\mathbf{X}^{\mathrm{T}}\mathcal{U}$ can be performed in $O(Npr_2)$ time.

Since $\mathbf{W}$ and $\mathcal{U}$ depend on $\hat{\boldsymbol{\beta}}_G$, a pilot sample can be considered to obtain an estimator to replace $\hat{\boldsymbol{\beta}}_G$. Let $\breve{\boldsymbol{\beta}}_G$ be the estimate for $\boldsymbol{\beta}$ in the model (2) based on the pilot sample. Then, the optimal subsampling probabilities can be approximated by using $\breve{\boldsymbol{\beta}}_G$ instead of $\hat{\boldsymbol{\beta}}_G$ in $\mathbf{W}$ and $\mathcal{U}$. The details of this projection based FASA algorithm for (4) are presented in Algorithm 1.

To construct the SRHT for any vector $z \in \mathbb{R}^N$, however, the size N needs to be a power of 2 which may not be the case in practice. Thus, we propose a more practical FASA algorithm using a random sampling matrix and the JLT. Let $\mathbf{R}$ be a $r_3 \times N$ random sampling matrix whose rows are chosen randomly from the rows of the $N \times N$ identity matrix. We first construct $\widehat{\mathbf{M}}_G = (\mathbf{R}\mathbf{W}^{1/2}\mathbf{X})^{\mathrm{T}}\mathbf{R}\mathbf{W}^{1/2}\mathbf{X}$ and then perform the JLT for $\widehat{\mathbf{M}}_G^{-1}$, $\mathbf{T}_2\widehat{\mathbf{M}}_G^{-1}$. After that, we replace $\mathbf{T}_2\widetilde{\mathbf{M}}_G^{-1}$ by $\mathbf{T}_2\widehat{\mathbf{M}}_G^{-1}$ in (10). Then, the subsampling probabilities in (4) can be approximated based on $(\mathbf{T}_2\widehat{\mathbf{M}}_G^{-1})\mathbf{X}^{\mathrm{T}}\mathcal{U}$. It is also conducted in $O(Npr_2)$ time. In the same manner as Algorithm 1, we use $\breve{\boldsymbol{\beta}}_G$ to replace $\hat{\boldsymbol{\beta}}_G$ in $\mathbf{W}$ and $\mathcal{U}$. The details of the practical sampling based FASA for (4) are summarized in Algorithm 2.

Algorithm 1 Random projection based FASA for (4)

1. Construct $\breve{\mathbf{M}}_{G_1} = (\mathbf{T}_1\breve{\mathbf{W}}^{1/2}\mathbf{X})^{\mathrm{T}}\mathbf{T}_1\breve{\mathbf{W}}^{1/2}\mathbf{X}$, where $\mathbf{T}_1$ is a SRHT of $\breve{\mathbf{W}}^{1/2}\mathbf{X}$, $\breve{\mathbf{W}} = \mathrm{diag}(\breve{w}_1, ..., \breve{w}_N)$ and $\breve{w}_i = \ddot{u}(\breve{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)[\dot{\psi}\{u(\breve{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)\} - y_i] + \ddot{\psi}\{u(\breve{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)\}\dot{u}^2(\breve{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)$ for $i = 1, ..., N$.

2. Let $\mathbf{T}_2$ be a JLT for $\breve{\mathbf{M}}_{G_1}^{-1}$. After performing $\mathbf{T}_2\breve{\mathbf{M}}_{G_1}^{-1}$, construct $(\mathbf{T}_2\breve{\mathbf{M}}_{G_1}^{-1})\mathbf{X}^{\mathrm{T}}\breve{\mathcal{U}}$, where $\breve{\mathcal{U}} = \mathrm{diag}[\dot{u}(\breve{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_1), ..., \dot{u}(\breve{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_N)]$.

3. Replacing $\|\mathbf{M}_G^{-1}\dot{u}(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)\mathbf{x}_i\|$ by $\|(\mathbf{T}_2\breve{\mathbf{M}}_{G_1}^{-1}\mathbf{X}^{\mathrm{T}}\breve{\mathcal{U}})_{(i)}\|$ in (4), approximate the optimal subsampling probabilities as

$$\breve{\pi}_i^{G_1} = \frac{|y_i - \dot{\psi}\{u(\breve{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)\}|\|(\mathbf{T}_2\breve{\mathbf{M}}_{G_1}^{-1}\mathbf{X}^{\mathrm{T}}\breve{\mathcal{U}})_{(i)}\|}{\sum_{j=1}^N |y_j - \dot{\psi}\{u(\breve{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_j)\}|\|(\mathbf{T}_2\breve{\mathbf{M}}_{G_1}^{-1}\mathbf{X}^{\mathrm{T}}\breve{\mathcal{U}})_{(j)}\|}, \quad i = 1, ..., N,$$

   where $(\mathbf{T}_2\breve{\mathbf{M}}_{G_1}^{-1}\mathbf{X}^{\mathrm{T}}\breve{\mathcal{U}})_{(i)}$ is the ith column of $\mathbf{T}_2\breve{\mathbf{M}}_{G_1}^{-1}\mathbf{X}^{\mathrm{T}}\breve{\mathcal{U}}$.

Algorithm 2 Random sampling based FASA for (4)

Let $\mathbf{R}$ be a $r_3 \times N$ random sampling matrix.

1. Construct $\breve{\mathbf{M}}_{G_2} = (\mathbf{R}\breve{\mathbf{W}}^{1/2}\mathbf{X})^{\mathrm{T}}\mathbf{R}\breve{\mathbf{W}}^{1/2}\mathbf{X}$.

2. Let $\mathbf{T}_2$ be a JLT for $\breve{\mathbf{M}}_{G_2}^{-1}$. After performing $\mathbf{T}_2\breve{\mathbf{M}}_{G_2}^{-1}$, construct $(\mathbf{T}_2\breve{\mathbf{M}}_{G_2}^{-1})\mathbf{X}^{\mathrm{T}}\breve{\mathcal{U}}$.

3. Replacing $\|\mathbf{M}_G^{-1}\dot{u}(\hat{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)\mathbf{x}_i\|$ by $\|(\mathbf{T}_2\breve{\mathbf{M}}_{G_2}^{-1}\mathbf{X}^{\mathrm{T}}\breve{\mathcal{U}})_{(i)}\|$ in (4), approximate the optimal subsampling probabilities as

$$\breve{\pi}_i^{G_2} = \frac{|y_i - \dot{\psi}\{u(\breve{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_i)\}|\|(\mathbf{T}_2\breve{\mathbf{M}}_{G_2}^{-1}\mathbf{X}^{\mathrm{T}}\breve{\mathcal{U}})_{(i)}\|}{\sum_{j=1}^N |y_j - \dot{\psi}\{u(\breve{\boldsymbol{\beta}}_G^{\mathrm{T}}\mathbf{x}_j)\}|\|(\mathbf{T}_2\breve{\mathbf{M}}_{G_2}^{-1}\mathbf{X}^{\mathrm{T}}\breve{\mathcal{U}})_{(j)}\|}, \quad i = 1, ..., N,$$

   where $(\mathbf{T}_2\breve{\mathbf{M}}_{G_2}^{-1}\mathbf{X}^{\mathrm{T}}\breve{\mathcal{U}})_{(i)}$ is the ith column of $\mathbf{T}_2\breve{\mathbf{M}}_{G_2}^{-1}\mathbf{X}^{\mathrm{T}}\breve{\mathcal{U}}$.

We establish the following theoretical result to examine the effect of subsample sizes on approximation accuracy of the subsampling probability. Let $\sigma_{min}(A)$ and $\sigma_{max}(A)$ be the largest and smallest non-zero singular values of a matrix A.

Theorem 3. *Assume that* $\nu_1 \in (0, 1/3)$, $\sigma_{min}\left(\widehat{\mathbf{M}}_G\right) \geq \gamma\sigma_{min}\left(\mathbf{M}_G\right)$ *for some* $\gamma \in (0, 1]$. *If* $r_2 \geq \dfrac{12\log n - 6\log\nu_1}{\epsilon_1^2}$ *for* $\epsilon_1 \in (0, 1/2]$, *with probability at least* $1-(\nu_1+\nu_2)-\nu_1*\nu_2$, *we have*

$$\|(\mathbf{M}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)} - (\mathbf{T}_2\widehat{\mathbf{M}}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\| \leq \frac{\|(\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\|}{\gamma\sigma_{min}\left(\mathbf{M}_G\right)}\left(\frac{\sigma_{max}^2\left(\mathbf{W}^{1/2}\mathbf{X}\right)}{\sigma_{min}\left(\mathbf{M}_G\right)}\frac{4p^2\sqrt{\log 1/\nu_2}}{\alpha\sqrt{r_3}} + (1+\epsilon_1)\sqrt{p}\right).$$

Theorem 3 indicates that the approximation accuracy of FASA.RS-π^G is improved as r_2 and r_3 are larger. However, this comes at the expense of longer computational time.

3.2. Subsampling probability approximation for Gaussian linear model

In this subsection, we develop specific FASA algorithms for the optimal subsampling probabilities in (8). We can apply Algorithm 1 and Algorithm 2 to approximate the subsampling probabilities by letting $\breve{\mathbf{W}} = \breve{\mathcal{U}} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix.

Since $\{e_i\}_{i=1}^N$ and $\hat{\sigma}_L^2$ in (8) depend on $\hat{\boldsymbol{\beta}}_L$, we use a pilot sample to obtain an estimate to replace $\hat{\boldsymbol{\beta}}_L$. Let $\breve{\boldsymbol{\beta}}_L$ be the estimate based on the pilot sample. Based on $\breve{\boldsymbol{\beta}}_L$, we calculate residuals and mean squared error, denoted as $\{\breve{e}_i\}_{i=1}^N$ and $\breve{\sigma}_L^2$, where $\breve{e}_i = y_i - \breve{\boldsymbol{\beta}}_L^{\mathrm{T}}\mathbf{x}_i$ and $\breve{\sigma}_L^2 = \sum_{i=1}^N \breve{e}_i^2/N$. Then, we replace $\{e_i\}_{i=1}^N$ and $\hat{\sigma}_L^2$ by $\{\breve{e}_i\}_{i=1}^N$ and $\breve{\sigma}_L^2$, respectively. The detailed procedures are given in Algorithms 3 and 4.

Algorithm 3 Random projection based FASA for (8)

1. Construct $\widetilde{\mathbf{M}}_L = (\mathbf{T}_1\mathbf{X})^{\mathrm{T}}\mathbf{T}_1\mathbf{X}$ where $\mathbf{T}_1$ is a SRHT of $\mathbf{X}$.
2. Let $\mathbf{T}_2$ be a JLT for $\widetilde{\mathbf{M}}_L^{-1}$. After performing $\mathbf{T}_2\widetilde{\mathbf{M}}_L^{-1}$, construct $(\mathbf{T}_2\widetilde{\mathbf{M}}_L^{-1})\mathbf{X}^{\mathrm{T}}$.
3. Replacing $\|\mathbf{M}_L^{-1}\mathbf{x}_i\|$ by $\|(\mathbf{T}_2\widetilde{\mathbf{M}}_L^{-1}\mathbf{X}^{\mathrm{T}})_{(i)}\|$ in (8), approximate the optimal subsampling probabilities as

$$\breve{\pi}_i^{L_1} = \frac{\sqrt{\breve{e}_i^2\|(\mathbf{T}_2\widetilde{\mathbf{M}}_L^{-1}\mathbf{X}^{\mathrm{T}})_{(i)}\|^2 + (\breve{e}_i^2 - \breve{\sigma}_L^2)^2/(4n^2\breve{\sigma}_L^2)}}{\sum_{j=1}^N \sqrt{\breve{e}_j^2\|(\mathbf{T}_2\widetilde{\mathbf{M}}_L^{-1}\mathbf{X}^{\mathrm{T}})_{(j)}\|^2 + (\breve{e}_j^2 - \breve{\sigma}_L^2)^2/(4n^2\breve{\sigma}_L^2)}} \quad i = 1, ..., N,$$

   where $(\mathbf{T}_2\widetilde{\mathbf{M}}_L^{-1}\mathbf{X}^{\mathrm{T}})_{(i)}$ is the ith column of $\mathbf{T}_2\widetilde{\mathbf{M}}_L^{-1}\mathbf{X}^{\mathrm{T}}$.

Remark 1. *In* $\mathbf{T}_1$ *and* $\mathbf{T}_2$, r_1 *and* r_2 *should be appropriately specified for efficient computing time. The formulas* $r_1 = \Omega\left(\dfrac{p\log N}{\epsilon^2}\log\left(\dfrac{p\log N}{\epsilon^2}\right)\right)$ *and* $r_2 = O\left(\dfrac{\log N}{\epsilon^2}\right)$ *suggested by [6] can guide the choices for* r_1 *and* r_2 *to generate a JLT and a FJLT respectively, for* $\epsilon \in (0, 1/2]$. *Based on simulation results in Appendix E.2, we prefer that as a general rule,* $\dfrac{p\log N}{10}\log(p\log N) \leq r_1, r_3 \leq p(\log N)\log(p\log N)$ *and* $2\log p \leq r_2 < p$ *in practice.*

Algorithm 4 Random sampling based FASA for (8)

Let $\mathbf{R}$ be $r_3 \times n$ a random sampling matrix.

1. Construct $\widehat{\mathbf{M}}_L = (\mathbf{RX})^{\mathrm{T}}\mathbf{RX}$.
2. Let $\mathbf{T}_2$ be a JLT for $\widehat{\mathbf{M}}_L^{-1}$. After performing $\mathbf{T}_2\widehat{\mathbf{M}}_L^{-1}$, construct $(\mathbf{T}_2\widehat{\mathbf{M}}_L^{-1})\mathbf{X}^{\mathrm{T}}$.
3. Replacing $\|(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{x}_i\|$ by $\|(\mathbf{T}_2\widehat{\mathbf{M}}_L^{-1}\mathbf{X}^{\mathrm{T}})_{(i)}\|$ in (8), approximate the optimal subsampling probabilities as

$$\breve{\pi}_i^{L_2} = \frac{\sqrt{\breve{e}_i^2\|(\mathbf{T}_2\widehat{\mathbf{M}}_L^{-1}\mathbf{X}^{\mathrm{T}})_{(i)}\|^2 + (\breve{e}_i^2 - \breve{\sigma}_L^2)^2/(4n^2\breve{\sigma}_L^2)}}{\sum_{j=1}^N \sqrt{\breve{e}_j^2\|(\mathbf{T}_2\widehat{\mathbf{M}}_L^{-1}\mathbf{X}^{\mathrm{T}})_{(j)}\|^2 + (e_j^2 - \breve{\sigma}_L^2)^2/(4n^2\breve{\sigma}_L^2)}} \quad i = 1, ..., N,$$

   where $(\mathbf{T}_2\widehat{\mathbf{M}}_L^{-1}\mathbf{X}^{\mathrm{T}})_{(i)}$ is the ith column of $\mathbf{T}_2\widehat{\mathbf{M}}_L^{-1}\mathbf{X}^{\mathrm{T}}$.

4. Numerical examples

In this section, numerical experiments are conducted in linear regression and logistic regression to evaluate the performance of our proposed methods. Using the FASA algorithms, we approximate the optimal subsampling probabilities and then calculate estimates from the subsample which is taken based on the approximated subsampling probabilities.

4.1. Simulation studies

4.1.1. Linear Regression

We generated full data with size $N = 2^{17}$ according to the following model,

$$y_i = \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta} + \varepsilon_i \quad i = 1, \ldots, N,$$

where ε_i's are independent error terms and follow a normal distribution with mean zero and variance $\sigma^2 = 9$, and $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_{29})$ is a 30 dimensional vector including the intercept, β_0. Distributions of the covariates are considered in the following four scenarios.

Case 1. Covariates follow a multivariate normal distribution with mean $\mathbf{0}$ and variance-covariance matrix $\boldsymbol{\Sigma}$, $N(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}_{jk} = 0.5^{I(j\neq k)}$ for $j, k = 1, \ldots, 29$ and $I()$ is the indicator function. The true values of coefficients are $\boldsymbol{\beta} = (1, 0.25, 0.3, 0.35, 0, ..., 0)^{\mathrm{T}}$.

Case 2. This case is the same as the first case except that $\boldsymbol{\Sigma}_{jk} = 0.8^{I(j\neq k)}$. The true coefficients are $\boldsymbol{\beta} = (1, 0.4, 0.5, 0.6, 0, ..., 0)^{\mathrm{T}}$.

Case 3. Covariates follow a multivariate t distribution with degree of freedom 2, $t_2(\mathbf{0}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma}_{jk} = 0.5^{I(j\neq k)}$ for $j, k = 1, \ldots, 29$. The true coefficients are $\boldsymbol{\beta} = (1, 0.07, 0.10, 0.13, 0, ..., 0)^{\mathrm{T}}$.

Case 4. This case is the same as the third case except that $\boldsymbol{\Sigma}_{jk} = 0.8^{I(j\neq k)}$. The true coefficients are $\boldsymbol{\beta} = (1, 0.12, 0.15, 0.18, 0, ..., 0)^{\mathrm{T}}$.

The different nonzero coefficients in each case are considered to examine empirical power between roughly 0.3 and 1. Based on $B = 1000$ subsamples from the full data, we calculate the empirical mean squared errors of the resultant estimator using MSE $= \sum_{b=1}^{B} \|\tilde{\boldsymbol{\theta}}^{(b)} - \hat{\boldsymbol{\theta}}_{MLE}\|^2/B$ where $\tilde{\boldsymbol{\theta}}^{(b)}$ and $\hat{\boldsymbol{\theta}}_{MLE}$ are the estimates of $\boldsymbol{\theta} = (\boldsymbol{\beta}, \sigma)$ obtained from the bth subsample and the full data, respectively. To explore the performance for statistical tests ($H_0 : \beta_j = 0$ vs. $H_A : \beta_j \neq 0$), we calculate the empirical type I error for β_4 and power for β_1, β_2 and β_3.

We consider different subsampling probabilities to compare the performance: subsampling probabilities based on Algorithm 3 and Algorithm 4 (FASA.RP-π^L and FASA.RS-π^L, respectively), A-optimal subsampling probabilities based on (8) (ASP-π^L), and uniform subsampling probabilities (UNIF). We also present the results using subsampling probabilities obtained from Algorithm 5 in Appendix D (FASA.SVD-π^L). This algorithm is developed based on the ideas suggested in [7].

For FASA.RP-π^L, FASA.RS-π^L, FASA.SVD-π^L and ASP-π^L, a uniform pilot sample of size n_0 is taken to obtain estimates $\breve{\boldsymbol{\beta}}_L$ and $\breve{\sigma}_L^2$ for calculating the optimal subsampling probabilities. Then, a subsample of size n is taken based on the calculated subsampling probabilities, and combined with the pilot sample to estimate the coefficients. For UNIF, we use the total subsample sizes $n_0 + n$. We set $n_0 = 400$, $n = 400$, 600, 800, 1000. Also, we choose $r_1 = 1000$ and $r_2 = 10$, and $\widehat{\mathbf{M}}_L$ in Algorithm 4 is performed using the pilot sample without an additional sample.

The results for MSE are in Figure 1. For all cases, the MSEs for FASA.RP-π^L, FASA.RS-π^L, FASA.SVD-π^L and ASP-π^L are smaller than that of UNIF, and the MSE for FASA.RP-π^L is close to that for ASP-π^L method. In Case 1 and 2, the performance of FASA.RP-π^L and FASA.RS-π^L are similar, but better than FASA.SVD-π^L. When the covariates follow t_2 distribution, FASA.RP-π^L results in smaller MSE than FASA.RS-π^L and FASA.SVD-π^L. Also, we have additional simulation results to compare the performance in terms of MSEs between the A-optimal subsampling probabilities in (4) and (8) using Algorithms 3, 4 and 5. Detailed results are in Appendix E.1.

Figure 2 and 3 show the results from statistical testing. The empirical type I errors from all methods and cases are close to the nominal value of 0.05. Generally, the empirical power for β_1, β_2 and β_3 based on ASP-π^L, FASA.RP-π^L, FASA.RS-π^L and FASA.SVD-π^L are larger than that based on UNIF. Also, we observe that the power from FASA.RP-π^L and FASA.RS-π^L are comparable with that of ASP-π^L in most cases. When the covariates are highly correlated (Case 2 and 4), FASA.RP-π^L and FASA.RS-π^L outperform FASA.SVD-π^L. Additionally, we investigate the performance of the suggested methods for different pilot sample sizes, r_1, r_2 and r_3 in Appendix E.2. As expected, MSEs for the suggested algorithms decrease as n_0 increases, and they result in smaller MSE and longer computing time as r_1 and r_2 increase.

To evaluate the performance of computational efficiency for the algorithms, we report the CPU time (in seconds) for the different methods using data from Case 1. The R programming language (R Core Team, 2015) is used, along with the Rcpp package to interface with C++ for the SRHT in FASA.RP-π^L and FASA.SVD-π^L. All computations are carried out on a MacBook Pro with 2.5 GHz Intel Core i7 processor and 16 GB memory. Table 1 shows the results including the computing time for using the full data. We observe that FASA.RP-π^L, FASA.RS-π^L and FASA.SVD-π^L require less computing time than ASP-π^L because FASA algorithms save the computing time by approximating the optimal subsampling probabilities. As expected, the computing time for UNIF is the least since the additional step for calculating the subsampling probabilities is not required. FASA.RS-π^L, which does not perform the SRHT, takes less computing time than FASA.RP-π^L and FASA.SVD-π^L. ASP requires more computing time than using the full data. This is because the extra computing time to calculate the A-optimal subsampling probabilities is $O(Np^2)$, which is the same as that for computing the MLE based on full data in the linear model.

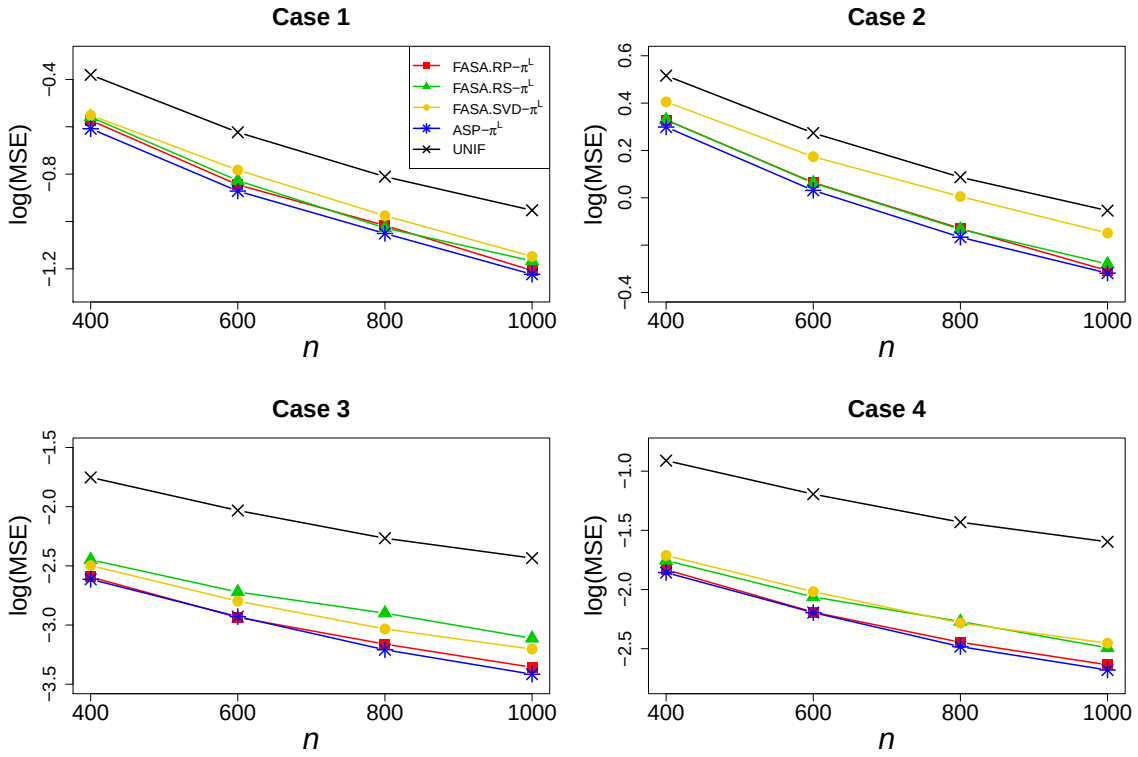
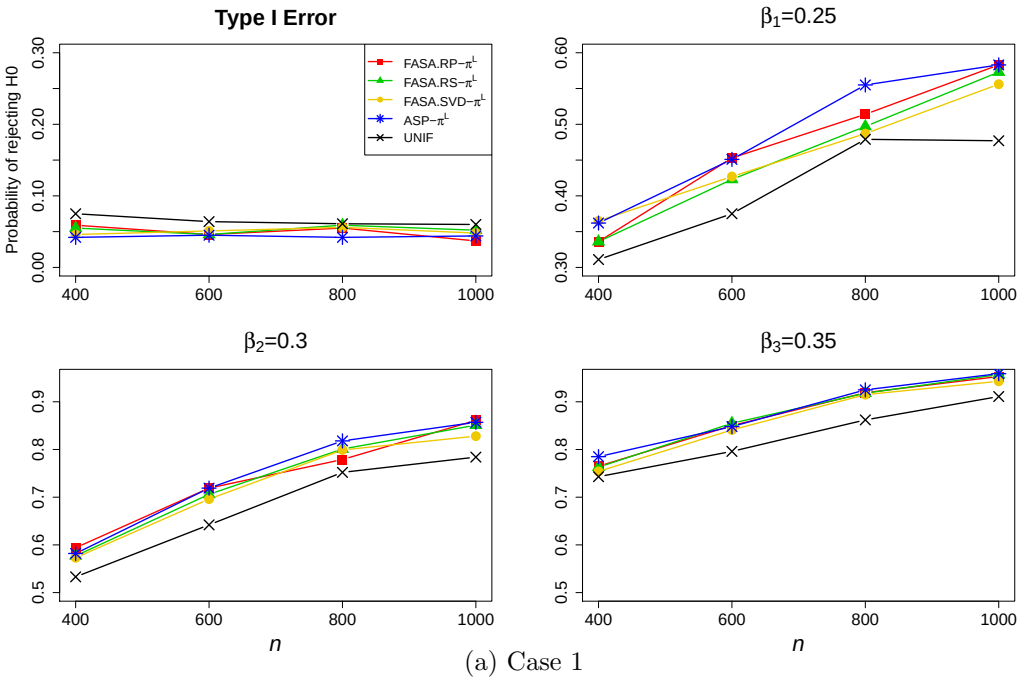
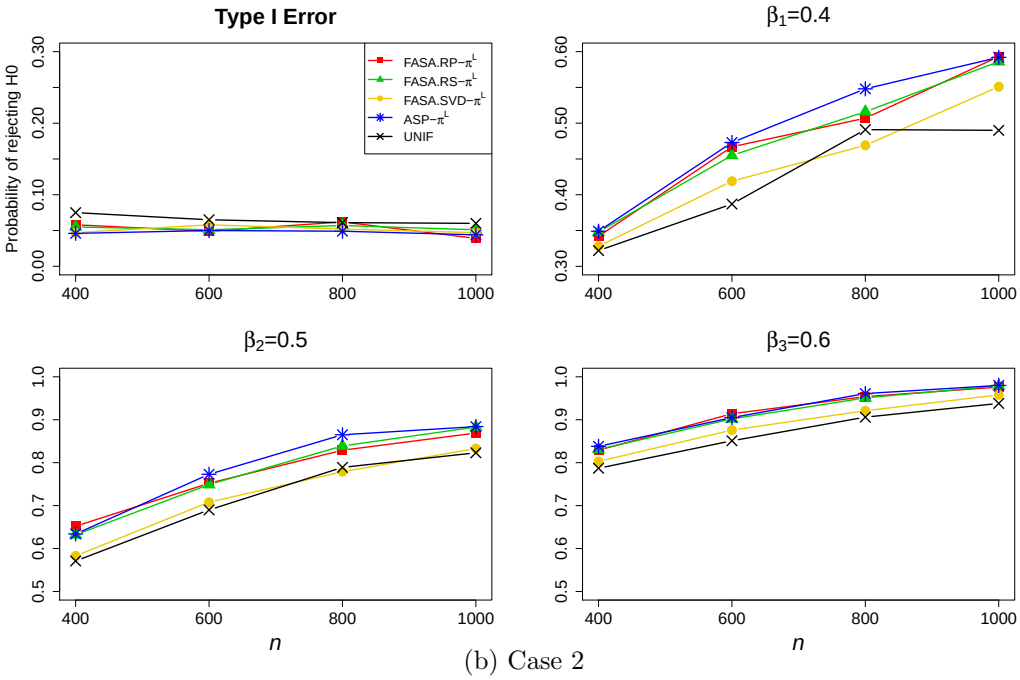


Figure 1: Logarithm of MSEs for varied subsample size n at a fixed $n_0 = 400$ in the linear model setting. FASA3, FASA.RS-π^L, and FASA.SVD-π^L use fast A-optimal subsampling probability approximation based on the Algorithm 3, Algorithm 4 and Algorithm 5, respectively, ASP-π^L uses A-optimal subsampling probabilities based on (8), and UNIF uses uniform subsampling probabilities.

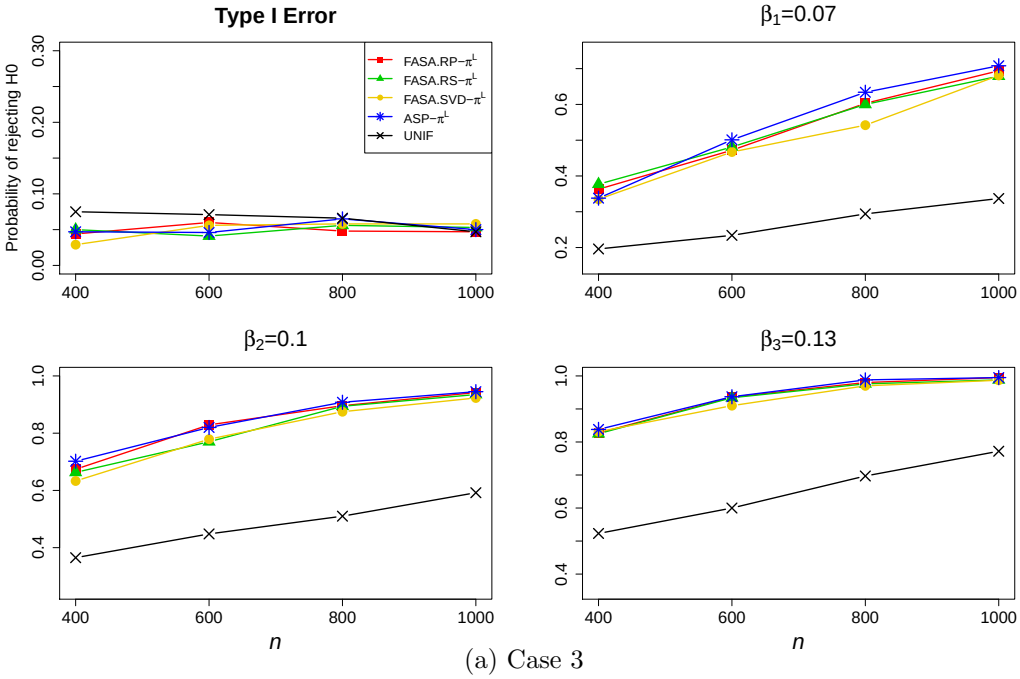


(a) Case 1

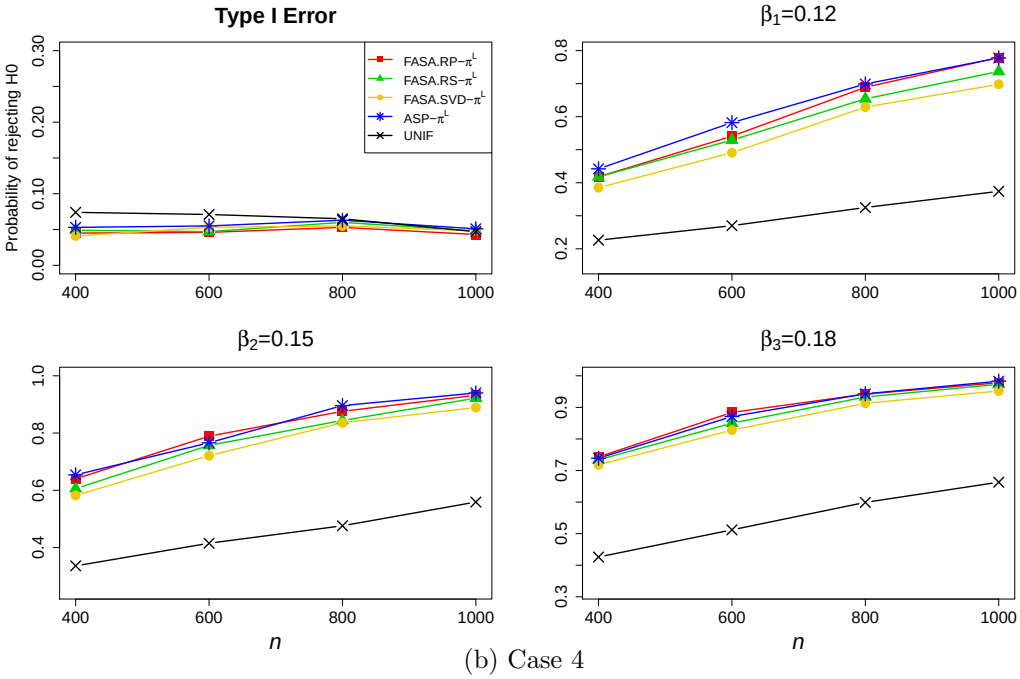


(b) Case 2

Figure 2: Empirical type I error and power in Cases 1 and 2 for different subsample size n at a fixed $n_0 = 400$ in linear model setting.



(a) Case 3



(b) Case 4

Figure 3: Empirical type I error and power in Cases 3 and 4 for varied subsample size n at fixed $n_0 = 400$ in the linear model setting.

Table 1: Average of CPU time (in seconds) in linear model using data from Case 1 with different subsample size n at a fixed $n_0 = 400$. The average of CPU time (in seconds) for the full data is provided in the last row. Repetition is 1000.

	n			
	400	600	800	1000
FASA.RP-π^L	0.2007	0.2011	0.2013	0.2021
FASA.RS-π^L	0.0533	0.0535	0.0539	0.0546
FASA.SVD-π^L	0.1944	0.1958	0.1960	0.1962
ASP-π^L	0.3939	0.3944	0.3945	0.3950
UNI	0.0020	0.0024	0.0028	0.0034
FULL data CPU seconds : 0.3394				

Table 2: Average of CPU time (in seconds) in linear model using data from Case 1 with different full data size N and number of covariates p at fixed $n = 2000$ and $n_0 = 1000$. The average of CPU time (in seconds) for the full data is provided in the last row. Repetition is 300.

	$N = 2^{17}$			$N = 2^{20}$		
	$p = 50$	$p = 80$	$p = 150$	$p = 50$	$p = 80$	$p = 150$
FASA.RP-π^L	0.3518	0.5946	1.2743	3.6518	5.8931	13.0123
FASA.RS-π^L	0.0975	0.1722	0.4051	0.8381	1.1997	2.3502
FASA.SVD-π^L	0.3388	0.5711	1.3163	3.6618	5.9331	13.2159
ASP-π^L	0.9546	2.2112	7.2071	9.0927	19.0304	60.3314
UNIF	0.0147	0.0312	0.0995	0.0159	0.0325	0.1040
FULL	0.8984	2.1314	6.9981	8.5794	18.4633	59.2473

To further explore the computational efficiency for more massive data, we consider $p = 50, 80, 150$ and $N = 2^{17}, 2^{20}$. We use data from Case 1 and set $n_0 = 1000$ and $n = 2000$ to record the computing time for 300 repetitions. Table 2 presents the results. As N and p increases, the computing based on the FASA algorithms becomes more efficient, compared to ASP-π^L and the full data approaches. Table A.1 in Appendix E.3 provides the results of the MSE, empirical type I error and power. The MSEs for FASA.RP-π^L and FASA.RS-π^L are closer to that for ASP-π^L than FASA.SVD-π^L in all cases. Overall, all methods show similar results for empirical type I error and power for β_1, β_2 and β_3.

4.1.2. Logistic Regression

In this section, a simulation study for logistic regression is performed. A full data with size $N = 2^{17}$ is generated from the model $y_i \sim Bernoulli(\theta_i)$ with $\text{logit}(\theta_i) = \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}$. We consider four cases for distributions of the covariates that are the same as the cases in the linear regression example, but we use different true values of coefficients: $\boldsymbol{\beta} = (1, 0.2, 0.25, 0.3, 0, ..., 0)^{\mathrm{T}}$ for Case 1, $\boldsymbol{\beta} = (1, 0.3, 0.35, 0.4, 0, ..., 0)^{\mathrm{T}}$ for Case 2, $\boldsymbol{\beta} = (1, 0.08, 0.10, 0.12, 0, ..., 0)^{\mathrm{T}}$ for Case 3, and $\boldsymbol{\beta} = (1, 0.18, 0.20, 0.22, 0, ..., 0)^{\mathrm{T}}$ for Case 4. We set the different nonzero coefficients to investigate empirical power between roughly 0.3 and 1. The $B = 1000$ subsamples from the full data are used to calculate MSE $= \sum_{b=1}^{B} \|\tilde{\boldsymbol{\beta}}^{(b)} - \hat{\boldsymbol{\beta}}_{MLE}\|^2/B$ where $\tilde{\boldsymbol{\beta}}^{(b)}$ is the estimate from the bth subsample and $\hat{\boldsymbol{\beta}}_{MLE}$ is the MLE based on the full data, and the empirical type I error for β_4 and power for

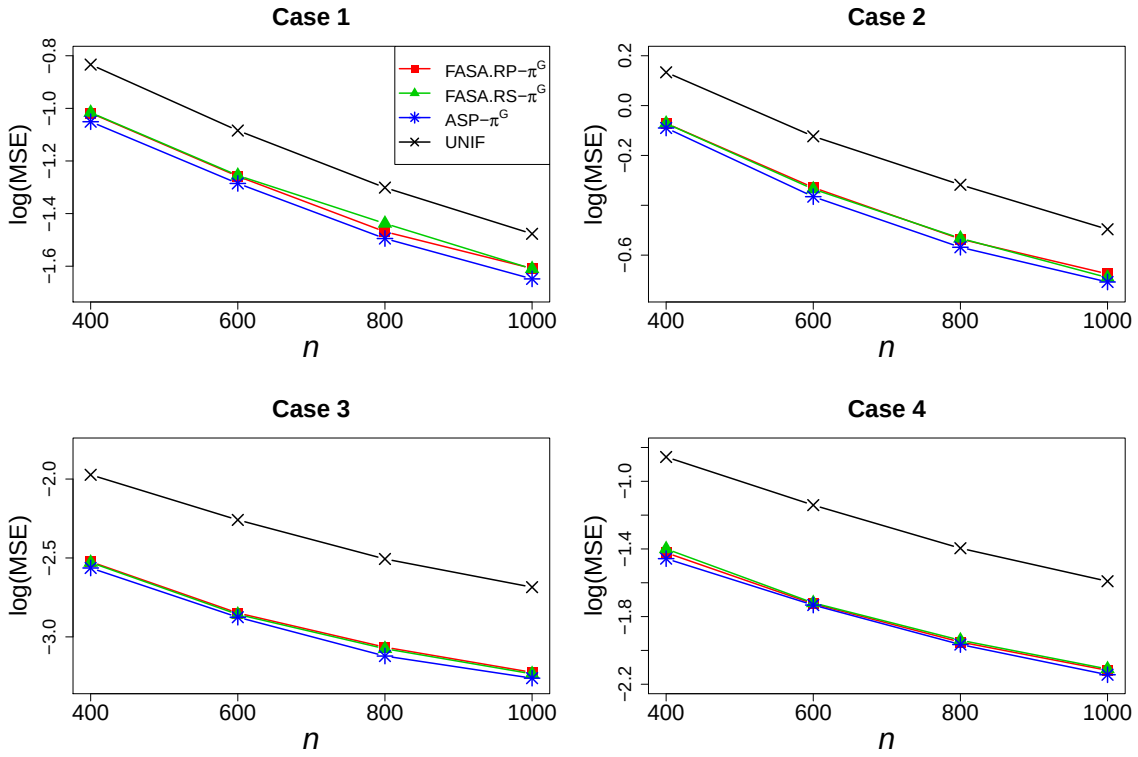
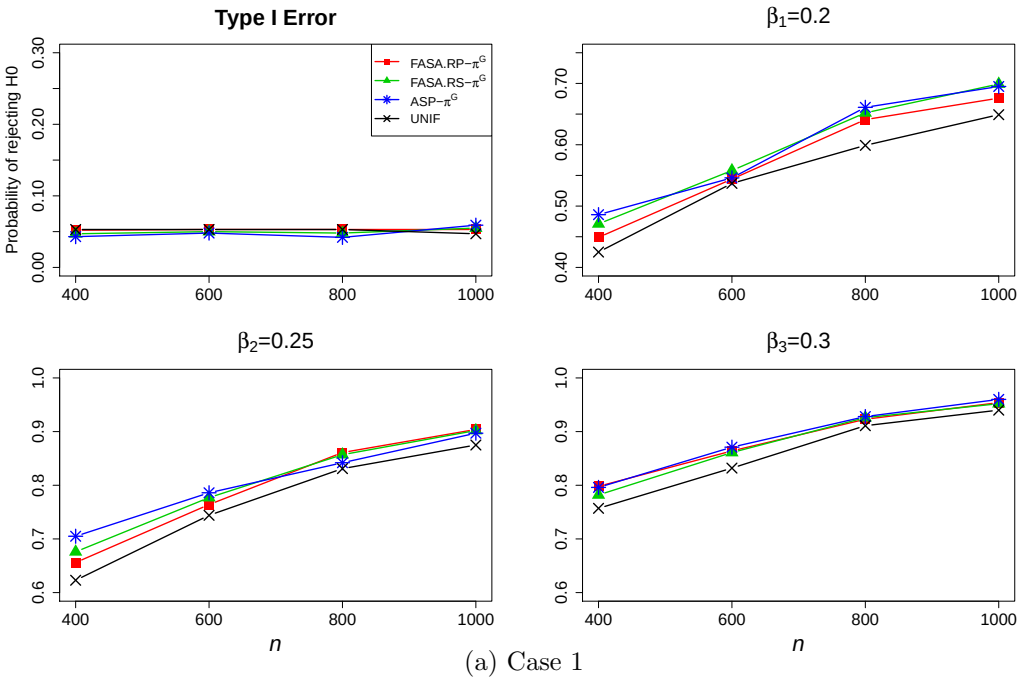
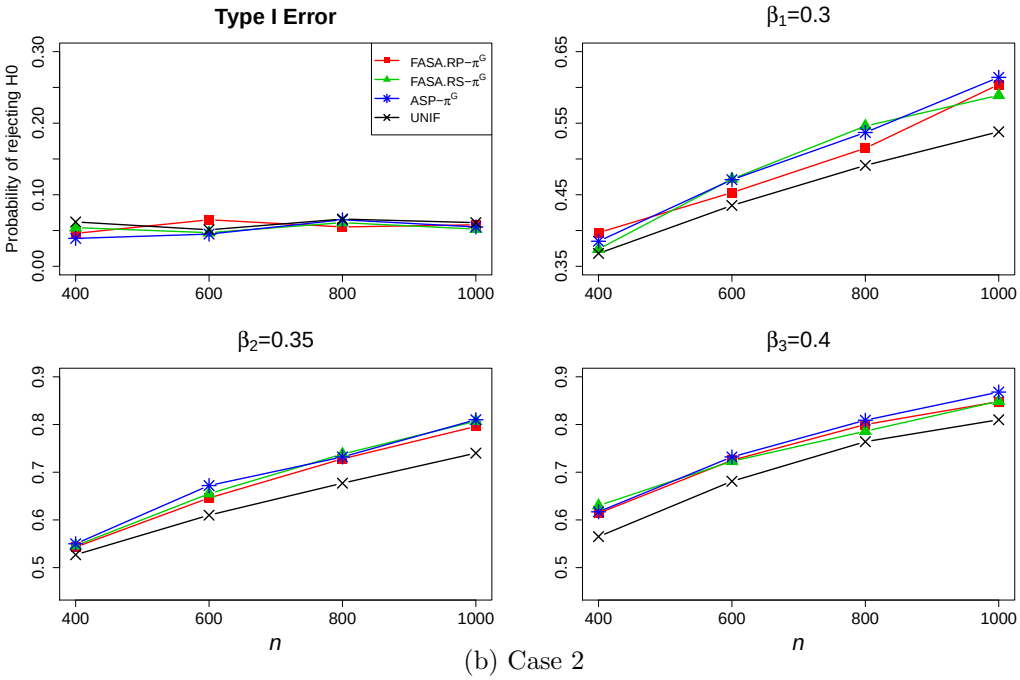


Figure 4: Logarithm of MSEs for varied subsample size n at a fixed $n_0 = 400$ in the logistic model setting. FASA.RP-π^G and FASA.RS-π^G use fast A-optimal subsampling probability approximation based on Algorithm 1 and Algorithm 2, respectively, ASP-π^G uses subsampling probabilities based on (4), and UNIF uses uniform subsampling probabilities.

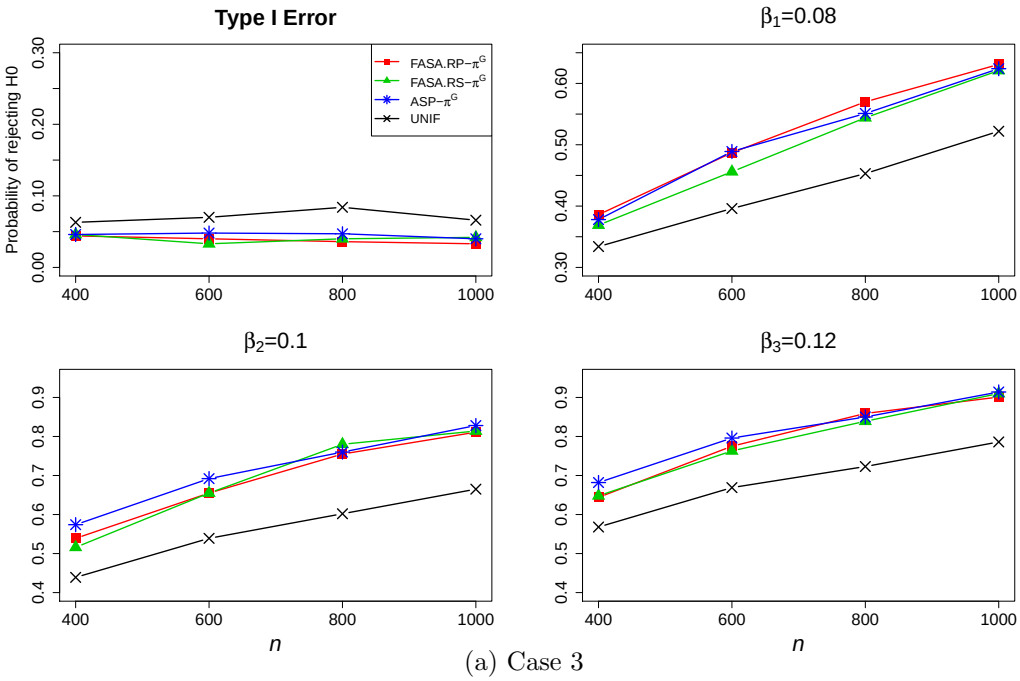


(a) Case 1

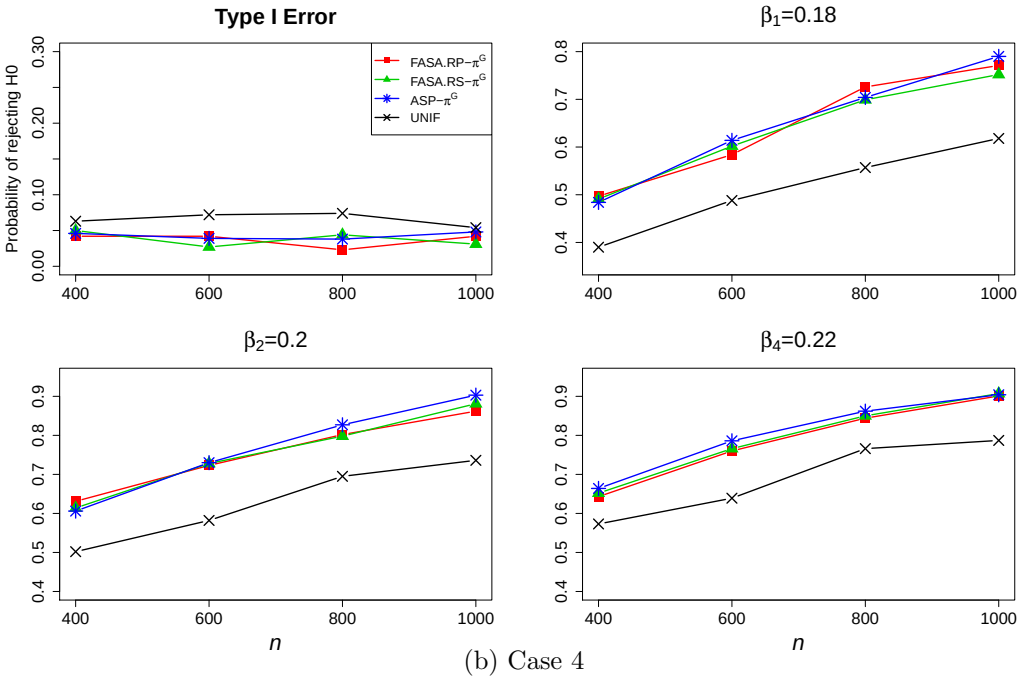


(b) Case 2

Figure 5: Empirical type I error and power in Cases 1 and 2 for different subsample size n at a fixed $n_0 = 400$ in logistic model setting.



(a) Case 3



(b) Case 4

Figure 6: Empirical type I error and power in Cases 3 and 4 for different subsample size n at a fixed $n_0 = 400$ in logistic model setting.

Table 3: Average of CPU time (in seconds) in logistic model using data from Case 1 with different subsample size n at a fixed $n_0 = 400$. The average of CPU time (in seconds) for using the full data is provided in the last row. Repetition is 1000.

	n			
	400	600	800	1000
FASA.RP-π^G	0.2188	0.2362	0.2366	0.2381
FASA.RS-π^G	0.0597	0.0605	0.0612	0.0623
ASP-π^G	0.2873	0.2880	0.2886	0.2905
UNI	0.0030	0.0037	0.0044	0.0055
FULL data CPU seconds : 0.6375				

Table 4: Average of CPU time (in seconds) in logistic model using data from Case 1 with different full data size N and number of covariates p at fixed $n = 2000$ and $n_0 = 1000$. The average of CPU time (in seconds) for the full data is provided in the last row. Repetition is 300.

	$N = 2^{17}$			$N = 2^{20}$		
	$p = 50$	$p = 80$	$p = 150$	$p = 50$	$p = 80$	$p = 150$
FASA.RP-π^G	0.3936	0.6697	1.4172	3.7703	6.4313	13.3619
FASA.RS-π^G	0.1166	0.1875	0.4417	0.8226	1.2236	2.3259
ASP-π^G	0.7303	1.5707	5.4712	6.5127	14.9707	48.5209
UNIF	0.0286	0.0624	0.1882	0.0282	0.0633	0.1937
FULL	1.5412	3.1870	11.0800	14.8427	32.5135	96.6608

β_1, β_2 and β_3 for testing $H_0 : \beta_j = 0$ vs. $H_A : \beta_j \neq 0$. Five different subsampling probabilities are compared: subsampling probabilities based on Algorithm 1 and Algorithm 2 (FASA.RP-π^G and FASA.RS-π^G, respectively), A-optimal subsampling probabilities on (4) (ASP-π^G), and uniform subsampling probabilities (UNIF). Other settings for the simulation are the same as the linear regression example.

Figure 4 presents the results for MSE. The MSEs for FASA.RP-π^G and FASA.RS-π^G are close to that for ASP-π^G in all cases. The UNIF method always has the worst performance. We also provide the results for the statistical testing in Figures 5 and 6. The empirical type I errors from all methods and cases are close to the nominal value of 0.05, except for UNIF in Cases 3 and 4. Compared to the proposed methods, UNIF tends to underestimate the variance of the estimator, which caused the inflation of type I errors. In general, ASP-π^G, FASA.RP-π^G and FASA.RS-π^G have better performance than UNIF for the empirical power.

Results on the CPU time (in seconds) for the different methods are reported in Tables 3 and 4. The settings for the simulation are the same as for the linear regression examples. In both tables, FASA.RP-π^G and FASA.RS-π^G are faster than ASP-π^G, and the UNIF always yields the fastest computing time. As N and p increase in Table 4, the computational efficiency of the FASA algorithms are more significant compared to the full data approach. As shown in Table A.2 in Appendix E.3, FASA.RP-π^G and FASA.RS-π^G show comparable performance with ASP-π^G in terms of MSE and statistical testing.

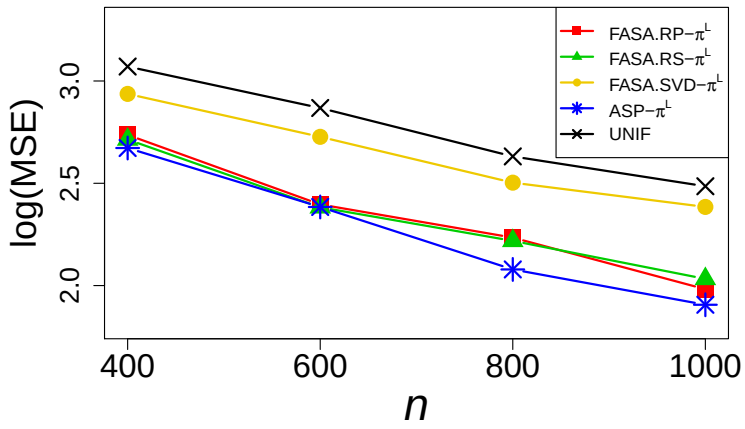


Figure 7: Logarithm of MSEs calculated from 1000 subsamples of the Online News Popularity Data for varied subsample size n at a fixed $n_0 = 400$.

4.2. Online News Popularity Data Analysis

In this section, we use the FASA algorithms to analyze online news popularity data. The data was collected from the contents of articles in Mashable, which is a news website, for two years. The detailed information of the data is described in Fernandes et al. [8], and the dataset is available at the UCI Machine Learning repository (`https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity`). For the linear regression model, the number of articles Mashable shares is used as the response. We consider 13 covariates: number of links, number of links to other articles published by Mashable, number of images, number of videos, average length of the words, rate of positive (negative) words, average (minimum, maximum) of polarity of positive words, and average (minimum, maximum) of polarity of negative words. The full data includes $N = 39,797$ observations. To apply FASA.RP-π^L and FASA.SVD-π^L to the dataset, we randomly draw a subset of data with the size $2^{15}(= 32,768)$ from the full data since the SRHT in the algorithms requires the size to be powers of 2. The responses are log-transformed, and we set $n_0 = 400$, $r_1 = 500$, and $r_2 = 5$.

Figure 7 shows the results for MSE $(= \sum_{b=1}^{1000} \|\tilde{\boldsymbol{\theta}}^{(b)} - \hat{\boldsymbol{\theta}}_{MLE}\|^2/1000)$ based on 1000 subsamples of the size $n_0 + n$ from the full data. ASP-π^L, FASA.RP-π^L, and FASA.RS-π^L have similar performance and they outperform the other methods. While the results of FASA.SVD-π^L are not close to those of ASP-π^L compared to other FASA algorithms, FASA.SVD-π^L gives smaller MSEs than UNIF.

To identify computational benefits for the suggested algorithms, we compute the average CPU times to obtain the estimates when $n = 1000$. The average computing times are 0.035 and 0.028 seconds for ASP-π^L and Full data, respectively, but 0.024, 0.008, and 0.023 seconds for FASA.RP-π^L, FASA.RS-π^L and FASA.SVD-π^L, respectively.

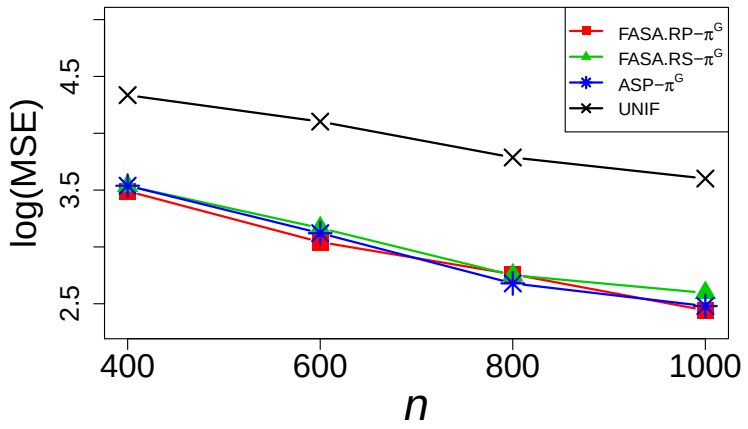


Figure 8: Logarithm of MSEs calculated from 1000 subsamples of the Forest Cover Type Data for varied subsample size n at a fixed $n_0 = 400$.

4.3. Forest Cover Type Data Analysis

For the logistic regression model, we analyze a forest cover type data (Blackard [4]). The data is available on the UCI Machine Learning repository: `https://archive.ics.uci.edu/ml/datasets/Covertype`. The dataset contains seven forest cover type classes, but two classes (Lodgepole Pine =0, spruce/fir = 1) are used for the logistic regression model. The percentage of Lodgepole Pine in the responses is about 57.22%. The following covariates are considered in the model: elevation, aspect, slope, horizontal (vertical) distance to the nearest surface water feature, horizontal distance to the nearest roadway, a relative measure of incident sunlight at 9:00 (at noon and at 15:00), and horizontal distance to the nearest historic wildfire ignition point. The full data sample size is $N = 495,141$. A sub-data of the size $2^{18}(= 262,144)$ is chosen randomly from the full data for FASA.RP-π^G and FASA.RS-π^G, and we consider $n_0 = 400$, $r_1 = 1,000$, and $r_2 = 5$.

The results of MSE $(= \sum_{b=1}^{1000} \|\tilde{\boldsymbol{\beta}}^{(b)} - \hat{\boldsymbol{\beta}}_{MLE}\|^2/1000)$ obtained from 1000 subsamples are in Figure (8). FASA.RP-π^G and FASA.RS-π^G yield smaller MSEs than UNIF, and they provide the results similar to those of ASP-π^G. We also observe that the FASA algorithms save computing time for obtaining the estimates, compared to ASP-π^G. The average CPU times for 1000 subsamples of the size $n_0 + n = 1400$ are 0.177 and 0.089 seconds for FASA.RP-π^G and FASA.RS-π^G, while it takes 0.231 seconds for ASP-π^G.

5. Conclusion

In this paper, we developed the FASA algorithms that approximate A-optimal subsampling probabilities for GLMs by performing the JLT and FJLT with high probabilities. In addition, we investigated the optimal subsampling method for the Gaussian linear model by taking into account both the regression coefficients and the dispersion parameter. Asymptotic results of the subsample estimators were examined and the A-optimal subsampling probabilities were derived by minimizing

the trace of the variance-covariance matrix of the estimators. To mitigate the computing burden, algorithms to approximate the optimal subsampling probabilities were also proposed. We have demonstrated the performance of the suggested algorithms through the simulation studies. In the linear regression setting, FASA.RP-π^L (algorithm constructed by the JLT and SRHT) showed comparable performance with ASP-π^L (A-optimal subsampling probability based on (8)) in terms of the empirical MSE and power for the coefficient parameter. FASA.RS-π^L (algorithm based on a random sampling matrix and the JLT) saved significantly the computing time compared to ASP-π^L and FASA.RP-π^L although it did not give better results for the empirical MSE and power than FASA.RP-π^L in some cases. In the logistic regression setting, both of FASA.RP-π^G (algorithm performed by the JLT and SRHT) and FASA.RS-π^G (algorithm based on a random sampling matrix and the JLT) provided similar MSEs and empirical power as ASP-π^G (A-optimal subsampling probability based on (4)) .

Based on Theorem 1, Theorem 3, and simulation results, we observe that the optimal subsampling methods give better performance in terms of MSE and power for statistical testing when r_1, r_2, r_3, pilot sample size, and subsample size increase. However, time complexities are $O(Np \log r_1 + Npr_2)$ for Algorithms 1 and 3, and $O(Npr_3 + Npr_2)$ for Algorithms 2 and 4. Moreover, the required times to obtain $\breve{\boldsymbol{\beta}}_G$ and $\breve{\boldsymbol{\beta}}_L$ are $O(n_0p^2\zeta_0)$ and $O(n_0p^2)$ where ζ_0 is the number of iterations in an iterative procedure. Lastly, the computing time to obtain estimates from the subsample is $O(np^2\zeta)$ for GLMs and $O(np^2)$ for linear regression models where ζ, where ζ is the number of iterations in an iterative procedure. It indicates that the computing time goes up for larger r_1, r_2, r_3, pilot sample size and subsample size. We face a tradeoff between computing time and desired accuracy (e.g., MSE and power). If better statistical accuracy is desired, more subsample sizes are required, although the computational efficiency is reduced. On the other hand, fewer subsample sizes can be considered for the benefit of computing time, but less accuracy.

There is an important question left to investigate in the future. When we approximated the optimal subsampling probabilities, we only considered JLT and FJLT. We can further consider other transformations such as a subsampled random Fourier transform [19]. To develop highly efficient algorithms, other transformation techniques are worthy of future investigation.

Acknowlegdements

The authors sincerely thank two reviewers and the associate editor for their comments, which greatly helped improve this manuscript. Wang's work research was partially supported by NSF grant DMS-1812013.

References

[1] Ai, M., Yu, J., Zhang, H., Wang, H., . Optimal subsampling algorithms for big data regressions. Statistica Sinica doi:10.5705/ss.202018.0439.

[2] Ailon, N., Chazelle, B., 2006. Approximate nearest neighbors and the fast johnson-lindenstrauss transform, in: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing, ACM. pp. 557–563.

[3] Ailon, N., Chazelle, B., 2009. The fast johnson–lindenstrauss transform and approximate nearest neighbors. SIAM Journal on computing 39, 302–322.

[4] Blackard, J., 1998. Comparison of Neural Networks and Discriminant Analysis in Predicting Forest Cover Types. Colorado State University. URL: https://books.google.com/books?id=0z0_NwAACAAJ.

[5] Chen, X., Xie, M.g., 2014. A split-and-conquer approach for analysis of extraordinarily large data. Statistica Sinica , 1655–1684.

[6] Drineas, P., Magdon-Ismail, M., Mahoney, M.W., Woodruff, D.P., 2012. Fast approximation of matrix coherence and statistical leverage. Journal of Machine Learning Research 13, 3475–3506.

[7] Drineas, P., Mahoney, M.W., Muthukrishnan, S., Sarlós, T., 2011. Faster least squares approximation. Numerische mathematik 117, 219–249.

[8] Fernandes, K., Vinagre, P., Cortez, P., 2015. A proactive intelligent decision support system for predicting the popularity of online news, in: Portuguese Conference on Artificial Intelligence, Springer. pp. 535–546.

[9] Lee, J., Wang, H., Schifano, E.D., 2020. Online updating method to correct for measurement error in big data streams. Computational Statistics & Data Analysis 149, 106976.

[10] Lin, N., Xi, R., 2011. Aggregated estimating equation estimation. Statistics and Its Interface 4, 73–83.

[11] Ma, P., Mahoney, M.W., Yu, B., 2015. A statistical perspective on algorithmic leveraging. The Journal of Machine Learning Research 16, 861–911.

[12] Schifano, E.D., Wu, J., Wang, C., Yan, J., Chen, M.H., 2016. Online updating of statistical inference in the big data setting. Technometrics 58, 393–403.

[13] Wang, C., Chen, M.H., Schifano, E., Wu, J., Yan, J., 2016. Statistical methods and computing for big data. Statistics and its interface 9, 399.

[14] Wang, C., Chen, M.H., Wu, J., Yan, J., Zhang, Y., Schifano, E., 2018a. Online updating method with new variables for big data streams. Canadian Journal of Statistics 46, 123–146.

[15] Wang, H., 2019a. Divide-and-conquer information-based optimal subdata selection algorithm. Journal of Statistical Theory and Practice 13, 46.

[16] Wang, H., 2019b. More efficient estimation for logistic regression with optimal subsamples. Journal of Machine Learning Research 20, 1–59.

[17] Wang, H., Ma, Y., 2020. Optimal subsampling for quantile regression in big data. Biometrika , DOI:10.1093/biomet/asaa043doi:10.1093/biomet/asaa043.

[18] Wang, H., Zhu, R., Ma, P., 2018b. Optimal subsampling for large sample logistic regression. Journal of the American Statistical Association 113, 829–844.

[19] Woolfe, F., Liberty, E., Rokhlin, V., Tygert, M., 2008. A fast randomized algorithm for the approximation of matrices. Applied and Computational Harmonic Analysis 25, 335–366.

[20] Wu, J., Chen, M.H., Schifano, E.D., Yan, J., 2018. Online Updating of Survival Analysis. Technical Report 18-30. University of Connecticut, Department of Statistics.

[21] Xue, Y., Wang, H., Yan, J., Schifano, E.D., 2020. An online updating approach for testing the proportional hazards assumption with streams of survival data. Biometrics 76, 171–182.

Supplementary Materials

Fast Optimal Subsampling Probability Approximation for Generalized Linear Models

JooChul Lee, Elizabeth D. Schifano, and HaiYing Wang

University of Connecticut

Appendix A. Proof of Theorem 1

In this section, we prove Theorem 1. We first investigate asymptotic properties of subsample estimators, $(\tilde{\boldsymbol{\beta}}_L^{\mathrm{T}}, \tilde{\sigma}_L^2)^{\mathrm{T}}$.

$$
\begin{aligned}
\begin{pmatrix} \tilde{\boldsymbol{\beta}}_L - \hat{\boldsymbol{\beta}}_L \\ \tilde{\sigma}_L^2 - \hat{\sigma}_L^2 \end{pmatrix} &= \begin{bmatrix} \left(\sum_{i=1}^n \frac{\mathbf{x}_i^* \mathbf{x}_i^{*\mathrm{T}}}{\pi_i^*}\right)^{-1} \sum_{i=1}^n \frac{\mathbf{x}_i^* y_i^*}{\pi_i^*} - \hat{\boldsymbol{\beta}}_L \\ \left(\sum_{i=1}^n \frac{1}{\pi_i^*}\right)^{-1} \sum_{i=1}^n \frac{(y_i^* - \tilde{\boldsymbol{\beta}}_L^{\mathrm{T}} \mathbf{x}_i^*)^2}{\pi_i^*} - \hat{\sigma}_L^2 \end{bmatrix} \\
&= \begin{bmatrix} \sum_{i=1}^n \frac{\mathbf{x}_i^* \mathbf{x}_i^{*\mathrm{T}}}{\pi_i^*} & \mathbf{0} \\ \mathbf{0} & \sum_{i=1}^n \frac{1}{\pi_i^*} \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^n \left(\frac{\mathbf{x}_i^* y_i^*}{\pi_i^*} - \frac{\mathbf{x}_i^* \mathbf{x}_i^{*\mathrm{T}}}{\pi_i^*} \hat{\boldsymbol{\beta}}_L \right) \\ \sum_{i=1}^n \left(\frac{(y_i^* - \tilde{\boldsymbol{\beta}}_L^{\mathrm{T}} \mathbf{x}_i^*)^2}{\pi_i^*} - \frac{1}{\pi_i^*} \hat{\sigma}_L^2 \right) \end{bmatrix} \\
&= \begin{bmatrix} \sum_{i=1}^n \frac{\mathbf{x}_i^* \mathbf{x}_i^{*\mathrm{T}}}{\pi_i^*} & \mathbf{0} \\ \mathbf{0} & \sum_{i=1}^n \frac{1}{\pi_i^*} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \sum_{i=1}^n \left(\frac{\mathbf{x}_i^* y_i^*}{\pi_i^*} - \frac{\mathbf{x}_i^* \mathbf{x}_i^{*\mathrm{T}}}{\pi_i^*} \hat{\boldsymbol{\beta}}_L \right) \\ \sum_{i=1}^n \left(\frac{(y_i^* - \hat{\boldsymbol{\beta}}_L^{\mathrm{T}} \mathbf{x}_i^*)^2}{\pi_i^*} - \frac{1}{\pi_i^*} \hat{\sigma}_L^2 \right) \end{bmatrix} \right. \\
&\quad \left. + \begin{bmatrix} \mathbf{0} \\ \tilde{\boldsymbol{\beta}}_L^{\mathrm{T}} \sum_{i=1}^n \frac{\mathbf{x}_i^* \mathbf{x}_i^{*\mathrm{T}}}{\pi_i^*} \tilde{\boldsymbol{\beta}}_L - \hat{\boldsymbol{\beta}}_L^{\mathrm{T}} \sum_{i=1}^n \frac{\mathbf{x}_i^* \mathbf{x}_i^{*\mathrm{T}}}{\pi_i^*} \hat{\boldsymbol{\beta}}_L + 2(\hat{\boldsymbol{\beta}}_L - \tilde{\boldsymbol{\beta}}_L)^{\mathrm{T}} \sum_{i=1}^n \frac{\mathbf{x}_i^* y_i^*}{\pi_i^*} \end{bmatrix} \right\} \\
&= \widetilde{\mathbf{M}}_L^{-1} \frac{1}{n} \sum_{i=1}^n \mathbf{A}_i + \widetilde{\mathbf{M}}_L^{-1} \mathbf{B}, \qquad \text{(A.1)}
\end{aligned}
$$

where

$$
\widetilde{\mathbf{M}}_L = \sum_{i=1}^n \begin{bmatrix} \frac{\mathbf{x}_i^* \mathbf{x}_i^{*\mathrm{T}}}{nN\pi_i^*} & \mathbf{0} \\ \mathbf{0} & \frac{1}{nN\pi_i^*} \end{bmatrix}, \mathbf{A}_i = \begin{bmatrix} \frac{\mathbf{x}_i^* y_i^* - \mathbf{x}_i^* \mathbf{x}_i^{*\mathrm{T}} \hat{\boldsymbol{\beta}}_L}{N\pi_i^*} \\ \frac{(y_i^* - \hat{\boldsymbol{\beta}}_L^{\mathrm{T}} \mathbf{x}_i^*)^2 - \hat{\sigma}_L^2}{N\pi_i^*} \end{bmatrix},
$$

and

$$\mathbf{B} = \sum_{i=1}^{n} \begin{bmatrix} \mathbf{0} \\ (\tilde{\boldsymbol{\beta}}_L - \hat{\boldsymbol{\beta}}_L)^{\mathrm{T}} \frac{\mathbf{x}_i^* \mathbf{x}_i^{*^{\mathrm{T}}}}{nN\pi_i^*} (\tilde{\boldsymbol{\beta}}_L - \hat{\boldsymbol{\beta}}_L) - 2(\tilde{\boldsymbol{\beta}}_L - \hat{\boldsymbol{\beta}}_L)^{\mathrm{T}} \left(\frac{\mathbf{x}_i^* y_i^*}{nN\pi_i^*} - \frac{\mathbf{x}_i^* \mathbf{x}_i^{*^{\mathrm{T}}}}{nN\pi_i^*} \hat{\boldsymbol{\beta}}_L \right) \end{bmatrix}.$$

We first discuss $\widetilde{\mathbf{M}}_L$ in (A.1). By direct calculation, we know that

$$\mathbf{E}(\widetilde{\mathbf{M}}_L | \mathcal{D}_N) = \widetilde{\mathbf{M}}, \tag{A.2}$$

where $\widetilde{\mathbf{M}} = \begin{bmatrix} \sum_{i=1}^{N} \frac{\mathbf{x}_i \mathbf{x}_i^{\mathrm{T}}}{N} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$.

Let $\widetilde{\mathbf{M}}_L^{j_1,j_2}$ be the (j_1, j_2) entry of the matrix $\widetilde{\mathbf{M}}$. For $1 \le j_1, j_2 \le p$,

$$\begin{aligned}
\mathbf{V}\left(\widetilde{\mathbf{M}}_L^{j_1,j_2} \middle| \mathcal{D}_N\right) &= \frac{1}{n} \sum_{i=1}^{N} \pi_i \left(\frac{x_{ij_1} x_{ij_2}}{N\pi_i} - \widetilde{\mathbf{M}}^{j_1,j_2} \right)^2 \\
&= \frac{1}{n} \sum_{i=1}^{N} \pi_i \left(\frac{x_{ij_1} x_{ij_2}}{N\pi_i} \right)^2 - \frac{1}{n} (\widetilde{\mathbf{M}}^{j_1,j_2})^2 \\
&\le \frac{1}{n} \sum_{i=1}^{N} \pi_i \left(\frac{x_{ij_1} x_{ij_2}}{N\pi_i} \right)^2 \\
&\le \frac{1}{n} \sum_{i=1}^{N} \frac{\|\mathbf{x}_i\|^4}{N^2 \pi_i} \\
&= O_P(n^{-1}).
\end{aligned} \tag{A.3}$$

For $j_1, j_2 = p + 1$,

$$\begin{aligned}
\mathbf{V}\left(\widetilde{\mathbf{M}}_L^{j_1,j_2} \middle| \mathcal{D}_N\right) &= \frac{1}{n} \sum_{i=1}^{N} \pi_i \left(\frac{1}{N\pi_i^*} - \widetilde{\mathbf{M}}_L^{j_1,j_2} \right)^2 \\
&= \frac{1}{n} \sum_{i=1}^{N} \pi_i \left(\frac{1}{N\pi_i} \right)^2 - \frac{1}{n} (\widetilde{\mathbf{M}}_L^{j_1,j_2})^2 \\
&\le \frac{1}{n} \sum_{i=1}^{N} \frac{1}{N^2 \pi_i} \\
&= O_P(n^{-1}).
\end{aligned} \tag{A.4}$$

In (A.3) and (A.4), the last equalities are from Assumption 1. Then, from Markov's inequality, (A.2), (A.3), and (A.4), conditionally on $\mathcal{D}_N$ in probability,

$$\widetilde{\mathbf{M}}_L - \widetilde{\mathbf{M}} = O_{P|\mathcal{D}_N}(n^{-1/2}). \tag{A.5}$$

Now, we discuss $\frac{1}{n}\sum_{i=1}^{n}\mathbf{A}_i$ in (A.1). Given $\mathcal{D}_N, \mathbf{A}_1, ..., \mathbf{A}_n$ are i.i.d with mean

$$\mathbf{E}(\mathbf{A}_i|\mathcal{D}_N) = \sum_{i=1}^{N}\pi_i \begin{bmatrix} \frac{\mathbf{x}_i y_i - \mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_L}{N\pi_i} \\ \frac{(y_i - \hat{\boldsymbol{\beta}}_L^{\mathrm{T}}\mathbf{x}_i)^2 - \hat{\sigma}_L^2}{N\pi_i} \end{bmatrix} = \mathbf{0}, \tag{A.6}$$

and variance

$$\mathbf{V}(\mathbf{A}_i|\mathcal{D}_N) = \sum_{i=1}^{N}\begin{bmatrix} \frac{(y_i - \hat{\boldsymbol{\beta}}_L^{\mathrm{T}}\mathbf{x}_i)^2\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{N^2\pi_i} & \frac{(e_i^2 - \hat{\sigma}_L^2)e_i\mathbf{x}_i}{N^2\pi_i} \\ \frac{(e_i^2 - \hat{\sigma}_L^2)e_i\mathbf{x}_i^{\mathrm{T}}}{N^2\pi_i} & \frac{(e_i^2 - \hat{\sigma}_L^2)^2}{N^2\pi_i} \end{bmatrix}, \tag{A.7}$$

where $e_i = y_i - \hat{\boldsymbol{\beta}}_L^{\mathrm{T}}\mathbf{x}_i$.
Let $\mathbf{V}(\mathbf{A}_i|\mathcal{D}_N)^{j_1,j_2}$ be the (j_1, j_2) entry of the matrix $\mathbf{V}(\mathbf{A}_i|\mathcal{D}_N)$. For $1 \le j_1, j_2 \le p$,

$$\begin{aligned}\mathbf{V}(\mathbf{A}_i|\mathcal{D}_N)^{j_1,j_2} &= \sum_{i=1}^{N}\frac{e_i^2 x_{ij_1}x_{ij_2}}{N^2\pi_i} \\ &\le \sum_{i=1}^{N}\frac{e_i^2\|\mathbf{x}_i\|^2}{N^2\pi_i} \\ &\le \sqrt{\sum_{i=1}^{N}\frac{e_i^4}{N^2\pi_i}\sum_{i=1}^{N}\frac{\|\mathbf{x}_i\|^4}{N^2\pi_i}} \\ &= O_p(1).\end{aligned} \tag{A.8}$$

In (A.8), the last inequality and equality are from Holder's inequality and Assumption 1, respectively. For $j_1, j_2 = p+1$,

$$\begin{aligned}\mathbf{V}(\mathbf{A}_i|\mathcal{D}_N)^{j_1,j_2} = \sum_{i=1}^{N}\frac{(e_i^2 - \hat{\sigma}_L^2)^2}{N^2\pi_i} &\le \sum_{i=1}^{N}\frac{e_i^4}{N^2\pi_i} + 2\hat{\sigma}_L^2\left(\sum_{i=1}^{N}\frac{e_i^4}{N^2\pi_i}\right)^{\frac{1}{2}}\left(\sum_{i=1}^{N}\frac{1}{N^2\pi_i}\right)^{\frac{1}{2}} + \hat{\sigma}_L^2\sum_{i=1}^{N}\frac{1}{N^2\pi_i} \\ &= O_p(1).\end{aligned} \tag{A.9}$$

In (A.9), the last inequality and equality are from Holder's inequality and Assumption 1, respectively.
For $1 \le j_1 \le p$ and $j_2 = p+1$,

$$\begin{aligned}\mathbf{V}(\mathbf{A}_i|\mathcal{D}_n)^{j_1,j_2} &= \sum_{i=1}^{N}\frac{(e_i^2 - \hat{\sigma}_L^2)e_i x_{ij_1}}{N^2\pi_i} \\ &= \sum_{i=1}^{N}\frac{e_i^3\|\mathbf{x}_i\| - \hat{\sigma}_L^2 e_i\|\mathbf{x}_i\|}{N^2\pi_i} \\ &\le \left(\sum_{i=1}^{N}\frac{e_i^4}{N^2\pi_i}\right)^{\frac{3}{4}}\left(\sum_{i=1}^{N}\frac{\|\mathbf{x}_i\|^4}{N^2\pi_i}\right)^{\frac{1}{4}} - \hat{\sigma}_L^2\left(\sum_{i=1}^{N}\frac{e_i^4}{N^2\pi_i}\right)^{\frac{1}{4}}\left(\sum_{i=1}^{N}\frac{\|\mathbf{x}_i\|^4}{N^2\pi_i}\right)^{\frac{1}{4}}\left(\sum_{i=1}^{N}\frac{1}{N^2\pi_i}\right)^{\frac{1}{2}} \\ &= O_p(1).\end{aligned} \tag{A.10}$$

In (A.10), the last inequality and equality are from Holder's inequality and Assumption 1, respectively. From (A.8), (A.9), and (A.10), we have

$$\mathbf{V}(\mathbf{A}_i|\mathcal{D}_N) = O_p(1). \tag{A.11}$$

For every $\epsilon > 0$ and some $\delta > 0$,

$$\begin{aligned}
\sum_{i=1}^{n} \mathbf{E}\left(\left\| \frac{1}{n^2}\mathbf{A}_i \right\|^2 I(\|\mathbf{A}_i\| > n^{\frac{1}{2}}\varepsilon)|\mathcal{D}_N \right) &\leq \frac{1}{n^{1+\delta/2}\varepsilon^\delta} \sum_{i=1}^{n} \mathbf{E}\left(\|\mathbf{A}_i\|^{2+\delta} I(\|\mathbf{A}_i\| > n^{1/2}\varepsilon)|\mathcal{D}_N \right) \\
&\leq \frac{1}{n^{1+\delta/2}\varepsilon^\delta} \sum_{i=1}^{n} \mathbf{E}\left(\|\mathbf{A}_i\|^{2+\delta}|\mathcal{D}_N \right) \\
&= \frac{1}{n^{\delta/2}N^{2+\delta}\varepsilon^\delta} \sum_{i=1}^{N} \frac{|e_i|^{2+\delta}\|\mathbf{x}_i\|^{2+\delta} + |e_i^2 - \hat{\sigma}_L^2|^{2+\delta}}{\pi_i^{1+\delta}} \\
&= \frac{1}{n^{\delta/2}N^{2+\delta}\varepsilon^\delta} \left\{ \sum_{i=1}^{N} \frac{|e_i|^{2+\delta}\|\mathbf{x}_i\|^{2+\delta}}{\pi_i^{1+\delta}} + \sum_{i=1}^{N} \frac{|e_i^2|^{2+\delta}}{\pi_i^{1+\delta}} \right\} \\
&= \frac{1}{n^{\delta/2}\varepsilon^\delta} \left\{ \sqrt{\sum_{i=1}^{N} \frac{|e_i|^{2(2+\delta)}}{N^{2+\delta}\pi_i^{1+\delta}} \sum_{i=1}^{N} \frac{\|\mathbf{x}_i\|^{2(2+\delta)}}{N^{2+\delta}\pi_i^{1+\delta}}} \right. \\
&\quad \left. + \sum_{i=1}^{N} \frac{|e_i^2|^{2+\delta}}{N^{2+\delta}\pi_i^{1+\delta}} \right\} \\
&= O_P(n^{-\delta/2}).
\end{aligned} \tag{A.12}$$

The first inequality and the last equality are from Holder's inequality and Assumption 2, respectively. Then, we have, as $n, N \to \infty$,

$$\sum_{i=1}^{n} \mathbf{E}\left(\|n^{-1/2}\mathbf{A}_i\|^2 I(\|\mathbf{A}_i\| > n^{1/2}\varepsilon)|\mathcal{D}_N \right) \to 0. \tag{A.13}$$

This result indicates that the Lindeberg-Feller conditions are satisfied. Thus, by the Lindeberg-Feller Central Limit Theorem (Proposition 2.27 of van der Vaart, 1998), we obtain conditionally on $\mathcal{D}_N$,

$$\frac{1}{n^{1/2}}\{\mathbf{V}(\mathbf{A}_i|\mathcal{D}_N)\}^{-1/2} \sum_{i=1}^{n} \mathbf{A}_i \to N(\mathbf{0}, \mathbf{I}). \tag{A.14}$$

We also discuss $\mathbf{B}$ in (A.1). We know that from (A.6),

$$\mathbf{E}\left\{ \frac{1}{n}\sum_{i=1}^{n} \left(\frac{\mathbf{x}_i^* y_i^* - \mathbf{x}_i^* \mathbf{x}_i^{*\mathrm{T}} \hat{\boldsymbol{\beta}}_L}{N\pi_i^*} \right) \middle| \mathcal{D}_N \right\} = 0, \tag{A.15}$$

and from (A.8),

$$\mathbf{V}\left\{ \frac{1}{n}\sum_{i=1}^{n} \left(\frac{\mathbf{x}_i^* y_i^* - \mathbf{x}_i^* \mathbf{x}_i^{*\mathrm{T}} \hat{\boldsymbol{\beta}}_L}{N\pi_i^*} \right) \middle| \mathcal{D}_N \right\} = O_p(n^{-1}). \tag{A.16}$$

From Markov's inequality, (A.15) and (A.16), conditionally on $\mathcal{D}_N$ in probability,

$$\frac{1}{n}\sum_{i=1}^{n}\frac{\mathbf{x}_i^* y_i^* - \mathbf{x}_i^*\mathbf{x}_i^{*\mathrm{T}}\hat{\boldsymbol{\beta}}_L}{N\pi_i^*} = o_{P|\mathcal{D}_N}(1). \tag{A.17}$$

Note that $\widetilde{\mathbf{M}}_L = O_{P|\mathcal{D}_N}(1)$ from (A.5). Combining this with (A.17), we have conditionally on $\mathcal{D}_N$ in probability,

$$\begin{aligned}\tilde{\boldsymbol{\beta}}_L - \hat{\boldsymbol{\beta}}_L &= \left(\sum_{i=1}^{n}\frac{\mathbf{x}_i^*\mathbf{x}_i^{*\mathrm{T}}}{nN\pi_i^*}\right)^{-1}\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\mathbf{x}_i^* y_i^* - \mathbf{x}_i^*\mathbf{x}_i^{*\mathrm{T}}\hat{\boldsymbol{\beta}}_L}{N\pi_i^*}\right)\\ &= o_{P|\mathcal{D}_N}(1).\end{aligned} \tag{A.18}$$

Thus,

$$\begin{aligned}\mathbf{B} &= \begin{bmatrix}\mathbf{0}\\ (\tilde{\boldsymbol{\beta}}_L - \hat{\boldsymbol{\beta}}_L)^{\mathrm{T}}\sum_{i=1}^{n}\frac{\mathbf{x}_i^*\mathbf{x}_i^{*\mathrm{T}}}{nN\pi_i^*}(\tilde{\boldsymbol{\beta}}_L - \hat{\boldsymbol{\beta}}_L) - 2(\tilde{\boldsymbol{\beta}}_L - \hat{\boldsymbol{\beta}}_L)^{\mathrm{T}}\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\mathbf{x}_i^* y_i^* - \mathbf{x}_i^*\mathbf{x}_i^{*\mathrm{T}}\hat{\boldsymbol{\beta}}_L}{N\pi_i^*}\right)\end{bmatrix}\\ &= o_{P|\mathcal{D}_N}(1) + o_{P|\mathcal{D}_N}(1)\\ &= o_{P|\mathcal{D}_N}(1).\end{aligned} \tag{A.19}$$

From (A.5),

$$\widetilde{\mathbf{M}}_L^{-1} - \widetilde{\mathbf{M}}^{-1} = -\widetilde{\mathbf{M}}^{-1}(\widetilde{\mathbf{M}}_L - \widetilde{\mathbf{M}})\widetilde{\mathbf{M}}_L^{-1} = O_{P|\mathcal{D}_N}(n^{-1/2}). \tag{A.20}$$

Then, from (A.19) and (A.20),

$$\begin{aligned}\begin{pmatrix}\tilde{\boldsymbol{\beta}}_L - \hat{\boldsymbol{\beta}}_L\\ \tilde{\sigma}_L^2 - \hat{\sigma}_L^2\end{pmatrix} &= \widetilde{\mathbf{M}}_L^{-1}\frac{1}{n}\sum_{i=1}^{n}\mathbf{A}_i + \widetilde{\mathbf{M}}_L^{-1}\mathbf{B}\\ &= \widetilde{\mathbf{M}}^{-1}\frac{1}{n}\sum_{i=1}^{n}\mathbf{A}_i + (\widetilde{\mathbf{M}}_L^{-1} - \widetilde{\mathbf{M}}^{-1})\frac{1}{n}\sum_{i=1}^{n}\mathbf{A}_i + o_{P|\mathcal{D}_N}(1)\\ &= \widetilde{\mathbf{M}}^{-1}\frac{1}{n^{1/2}}\{\mathbf{V}(\mathbf{A}_i|\mathcal{D}_N)\}^{1/2}\frac{1}{n^{1/2}}\{\mathbf{V}(\mathbf{A}_i|\mathcal{D}_N)\}^{-1/2}\sum_{i=1}^{n}\mathbf{A}_i + o_{P|\mathcal{D}_N}(1).\end{aligned}$$

By Slutsky's Theorem, we obtain conditional on $\mathcal{D}_N$ in probability,

$$\begin{pmatrix}\tilde{\boldsymbol{\beta}}_L - \hat{\boldsymbol{\beta}}_L\\ \tilde{\sigma}_L^2 - \hat{\sigma}_L^2\end{pmatrix} \to \mathbf{N}(\mathbf{0}, \widetilde{\mathbf{M}}^{-1}\mathbf{V}_\mathbf{L}^\mathbf{c}\widetilde{\mathbf{M}}^{-1}),$$

in distribution, where $\mathbf{V}_L^c = n^{-1}\{\mathbf{V}(\mathbf{A}_i|\mathcal{D}_N)\}$.
Let $\mathbf{g}(\mathbf{a}, b) = (\mathbf{a}^{\mathrm{T}}, \sqrt{b})^{\mathrm{T}}$ for a specific vector $\mathbf{a}$ and value b. By applying the Delta method, we show that conditional on $\mathcal{D}_N$ in probability,

$$\left(\mathbf{g}(\tilde{\boldsymbol{\beta}}_L, \tilde{\sigma}_L) - \mathbf{g}(\hat{\boldsymbol{\beta}}_L, \hat{\sigma}_L)\right) = \begin{pmatrix}\tilde{\boldsymbol{\beta}}_L - \hat{\boldsymbol{\beta}}_L\\ \tilde{\sigma}_L - \hat{\sigma}_L\end{pmatrix} \to N(\mathbf{0}, \mathbf{V}),$$

in distribution, where $\mathbf{V}_\mathbf{L} = \nabla\mathbf{g}^{\mathrm{T}}\widetilde{\mathbf{M}}^{-1}\mathbf{V}^\mathbf{c}\widetilde{\mathbf{M}}^{-1}\nabla\mathbf{g} = \mathbf{M}^{-1}\mathbf{V}^\mathbf{c}\mathbf{M}^{-1}$ and $\nabla\mathbf{g} = \begin{bmatrix}\mathbf{I}_{p\times p} & \mathbf{0}\\ \mathbf{0} & \frac{1}{2\hat{\sigma}_L}\end{bmatrix}$.

Appendix B. Proof of Theorem 2

Note that $\mathrm{tr}(\mathbf{V}) = \mathrm{tr}(\mathbf{M}^{-1}\mathbf{V}^{\mathbf{c}}\mathbf{M}^{-1})$ and $e_i = y_i - \hat{\boldsymbol{\beta}}_L^{\mathrm{T}}\mathbf{x}_i$.

$$\begin{aligned}
\mathrm{tr}(\mathbf{M}^{-1}\mathbf{V}^{\mathbf{c}}\mathbf{M}^{-1}) &= \frac{1}{n}\left\{\mathrm{tr}\left((\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\sum_{i=1}^{N}\frac{e_i^2\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\pi_i}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\right) + \mathrm{tr}\left(\sum_{i=1}^{N}\frac{(e_i^2-\hat{\sigma}_L^2)^2}{4\pi_i N^2\hat{\sigma}_L^2}\right)\right\} \\
&= \frac{1}{n}\sum_{i=1}^{N}\frac{1}{\pi_i}\left\{\mathrm{tr}\left((\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}e_i^2\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\right) + \frac{(e_i^2-\hat{\sigma}_L^2)^2}{4N^2\hat{\sigma}_L^2}\right\} \\
&= \frac{1}{n}\sum_{i=1}^{N}\frac{1}{\pi_i}\left\{e_i^2\|(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{x}_i\|^2 + \frac{(e_i^2-\hat{\sigma}_L^2)^2}{4N^2\hat{\sigma}_L^2}\right\}\sum_{i=1}^{N}\pi_i \\
&\geq \frac{1}{n}\sum_{i=1}^{N}\left\{e_i^2\|(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{x}_i\|^2 + \frac{(e_i^2-\hat{\sigma}_L^2)^2}{4N^2\hat{\sigma}_L^2}\right\}.
\end{aligned}$$

The last inequality is from the Cauchy-Schwarz inequality and the equality holds if and only if $\pi_i \propto \sqrt{e_i^2\|(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{x}_i\|^2 + \frac{(e_i^2-\hat{\sigma}_L^2)^2}{4N^2\hat{\sigma}_L^2}}$.

Appendix C. Proof of Theorem 3

The following Lemmas are needed to derive Theorem 3. Let $\|\cdot\|_F$ be the Frobenius norm.

Lemma 1. *(Theorem 2.1 of Drineas et al. [4]) Let $A^{(i)}$ be the i-th row of A and $B_{(j)}$ be the i-th column of B. Suppose sampling probabilities p_i are such that*

$$p_i \geq \alpha\frac{\|A^{(i)}\|\|B_{(i)}\|}{\sum_{j=1}^{n}\|A^{(j)}\|\|B_{(j)}\|},$$

for some $\alpha \in (0,1]$. Construct C and R with Algorithm 1 in Drineas et al. [4], and assume that $\nu_1 \in (0,1/3)$. Then with probability at least $1-\nu_1$, we have

$$\|AB - CR\|_F \leq \frac{4\sqrt{\log 1/\nu_1}}{\alpha\sqrt{s}}\|A\|_F\|B\|_F.$$

Lemma 2. *(Theorem 1.1 of Achlioptas [1]) Let $x_1, ..., x_n$ be an arbitrary set of points, where $x_i \in \mathbb{R}^p$ and let $\epsilon \in (0,1/2]$ be an accuracy parameter. If $r_2 \geq \frac{1}{\epsilon_1^2}(12\log n + 6\log 1/\nu_2)$, with probability at least $1-\nu_2$,*

$$(1-\epsilon_1)\|x_i\|^2 \leq \|\mathbf{T}_2 x_i\|^2 \leq (1+\epsilon_1)\|x_i\|^2.$$

Now, we establishing Theorem 3.

$$
\begin{aligned}
&\|(\mathbf{M}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)} - (\mathbf{T}_2\widehat{\mathbf{M}}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\| \\
&= \|(\mathbf{M}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)} - (\widehat{\mathbf{M}}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)} + (\widehat{\mathbf{M}}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)} - (\mathbf{T}_2\widehat{\mathbf{M}}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\| \\
&\le \|(\mathbf{M}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)} - (\widehat{\mathbf{M}}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\| \\
&\quad + \|(\widehat{\mathbf{M}}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)} - (\mathbf{T}_2\widehat{\mathbf{M}}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\| \\
&\le \|(\mathbf{M}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)} - (\widehat{\mathbf{M}}_G^{-1}\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\| + (1+\epsilon_1)\|\widehat{\mathbf{M}}_G^{-1}\|_F\|(\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\| \\
&\le \|\mathbf{M}_G^{-1}\|_F\|\widehat{\mathbf{M}}_G^{-1}\|_F\|\mathbf{M}_G - \widehat{\mathbf{M}}_G\|_F\|(\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\| \\
&\quad + (1+\epsilon_1)\|\widehat{\mathbf{M}}_G^{-1}\|_F\|(\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\| \\
&\le \sigma_{max}\left(\mathbf{M}_G^{-1}\right)\sigma_{max}\left(\widehat{\mathbf{M}}_G^{-1}\right)\sigma_{max}^2\left(\mathbf{W}^{1/2}\mathbf{X}\right)p^2\frac{4\sqrt{\log 1/\nu_2}}{\alpha\sqrt{r_3}}\|(\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\| \\
&\quad + (1+\epsilon_1)\sigma_{max}\left(\widehat{\mathbf{M}}_G^{-1}\right)\sqrt{p}\|(\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\| \\
&\le \sigma_{min}^{-1}\left(\mathbf{M}_G\right)\sigma_{min}^{-1}\left(\widehat{\mathbf{M}}_G\right)\sigma_{max}^2\left(\mathbf{W}^{1/2}\mathbf{X}\right)\frac{4p^2\sqrt{\log 1/\nu_2}}{\alpha\sqrt{r_3}}\|(\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\| \\
&\quad + (1+\epsilon_1)\sigma_{min}^{-1}\left(\widehat{\mathbf{M}}_G\right)\sqrt{p}\|(\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\| \\
&\le \frac{\|(\mathbf{X}^{\mathrm{T}}\mathcal{U})_{(i)}\|}{\gamma\sigma_{min}\left(\mathbf{M}_G\right)}\left(\frac{\sigma_{max}^2\left(\mathbf{W}^{1/2}\mathbf{X}\right)}{\sigma_{min}\left(\mathbf{M}_G\right)}\frac{4p^2\sqrt{\log 1/\nu_2}}{\alpha\sqrt{r_3}} + (1+\epsilon_1)\sqrt{p}\right),
\end{aligned}
$$

where the second inequality is from Lemma 2, the last third inequality is from Lemma 1 by letting $A = B = \mathbf{W}^{1/2}\mathbf{X}$, $C = D = \mathbf{R}\mathbf{W}^{1/2}\mathbf{X}$, and $\alpha = 1/N$, and the last inequality is from the assumption $\sigma_{min}\left(\widehat{\mathbf{M}}_G\right) \ge \gamma\sigma_{min}\left(\mathbf{M}_G\right)$ for some $\gamma \in (0, 1]$.

Appendix D. Algorithm via Singular Value Decomposition Approach for Gaussian linear model

Denote the Singular Value Decomposition (SVD) of $\mathbf{X}$ as $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}$, where $\mathbf{U}$ is a $N \times p$ matrix, $\mathbf{D}$ is a $p \times p$ diagonal matrix and $\mathbf{V}$ is a $p \times p$ matrix. Since $\mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-2}\mathbf{X}^{\mathrm{T}} = \mathbf{U}\mathbf{D}^{-2}\mathbf{U}^{\mathrm{T}} = \mathbf{U}\mathbf{D}^{-1}(\mathbf{U}\mathbf{D}^{-1})^{\mathrm{T}}$, we can express

$$\|(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{x}_i\| = \|(\mathbf{D}^{-1}\mathbf{U}^{\mathrm{T}})_{(i)}\|, \quad i = 1, ..., N,$$

where $(\mathbf{D}^{-1}\mathbf{U}^{\mathrm{T}})_{(i)}$ is the ith column of $\mathbf{D}^{-1}\mathbf{U}^{\mathrm{T}}$. Thus, we focus on approximating $\|(\mathbf{D}^{-1}\mathbf{U}^{\mathrm{T}})_{(i)}\|$ for all i by using the idea for approximating statistical leverage scores proposed in [3]. Denote the SVD of $\mathbf{T}_1\mathbf{X}$ as $\mathbf{T}_1\mathbf{X} = \mathbf{U}_{\mathbf{T}_1\mathbf{X}}\mathbf{D}_{\mathbf{T}_1\mathbf{X}}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}$ where $\mathbf{T}_1$ is a SRHT of $\mathbf{X}$. We approximate $\mathbf{D}^{-1}\mathbf{U}^{\mathrm{T}}$ as

$$\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}} = \mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}\mathbf{V}\mathbf{D}\mathbf{U}^{\mathrm{T}}. \tag{D.1}$$

Since the computing time for $\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}$ requires $O(Np^2)$, we further consider a JLT for the columns of $\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}$, say $\mathbf{T}_2\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}$. After constructing $(\mathbf{T}_2\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}})\mathbf{X}^{\mathrm{T}}$, we approximate $\|(\mathbf{D}^{-1}\mathbf{U}^{\mathrm{T}})_{(i)}\|$ as

$$\|(\mathbf{T}_2\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}})_{(i)}\|, \quad i = 1, ..., N.$$

Algorithm 5 Random projection based FASA for (8)

1. Construct $\mathbf{T}_1\mathbf{X}$ where $\mathbf{T}_1$ is a SRHT of $\mathbf{X}$. Let its SVD be $\mathbf{T}_1\mathbf{X} = \mathbf{U}_{\mathbf{T}_1\mathbf{X}}\mathbf{D}_{\mathbf{T}_1\mathbf{X}}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}$.
2. Construct $\mathbf{T}_2\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}$ where $\mathbf{T}_2$ is a JLT for the rows of $\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}$. After that, perform $(\mathbf{T}_2\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}})\mathbf{X}^{\mathrm{T}}$.
3. Replacing $\|(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{x}_i\|$ by $\|(\mathbf{T}_2\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}})_{(i)}\|$ in (8), approximate the optimal subsampling probability as

$$\check{\pi}_i^{L3} = \frac{\sqrt{\check{e}_i^2\|(\mathbf{T}_2\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}})_{(i)}\|^2 + (\check{e}_i^2 - \check{\sigma}_L^2)^2/(4n^2\check{\sigma}_L^2)}}{\sum_{j=1}^N \sqrt{e_j^2\|(\mathbf{T}_2\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}})_{(j)}\|^2 + (\check{e}_j^2 - \check{\sigma}_L^2)^2/(4n^2\check{\sigma}_L^2)}} \quad i = 1, ..., N,$$

where $(\mathbf{T}_2\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}})_{(i)}$ is the ith column of $\mathbf{T}_2\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}$.

where $(\mathbf{T}_2\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}})_{(i)}$ is the ith column of $\mathbf{T}_2\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}$. The computing time to perform $(\mathbf{T}_2\mathbf{D}_{\mathbf{T}_1\mathbf{X}}^{-2}\mathbf{V}_{\mathbf{T}_1\mathbf{X}}^{\mathrm{T}})\mathbf{X}^{\mathrm{T}}$ is $O(Npr_2)$ time.

Appendix E. Additional numerical results

Appendix E.1. Comparison of MSEs between π_i^G and π_i^L in linear model setting.

In this section, we further compare performance between ASP-π^L and the A-optimal subsampling probabilities in (4) (ASP-π^G). Since Ai et al. [2] considered generalized linear regression assuming the dispersion parameter is known, ASP-π^G use the A-optimal subsampling probabilities determined without considering σ^2 in this example. The model setup is the same as that in Section 4.1.1. Figure A.1 provides the results of MSE. It is seen that ASP-π^L and ASP-π^G show similar performance in Cases 1 and 2, while ASP-π^L gives better results in Case 3 and 4. Also, we approximate π_i^G in (4) by applying Algorithm 3 and 4, and the first two steps in Algorithm 5 (FASA.RP-π^G, FASA.RS-π^G and FASA.SVD-π^G), and compare them with FASA.RP-π^L, FASA.RS-π^L and FASA.SVD-π^L. Figure A.2 presents that FASA.RP-π^L and FASA.RP-π^G have similar MSEs for Case 1 and 2, but FASA.RP-π^L results in smaller MSEs than FASA.RP-π^G for Cases 3 and 4. Likewise, FASA.SVD-π^L compared with FASA.SVD-π^G tends to give smaller MSEs in Cases 3 and 4. The results for FASA.RS-π^L and FASA.RS-π^G are similar in all cases.

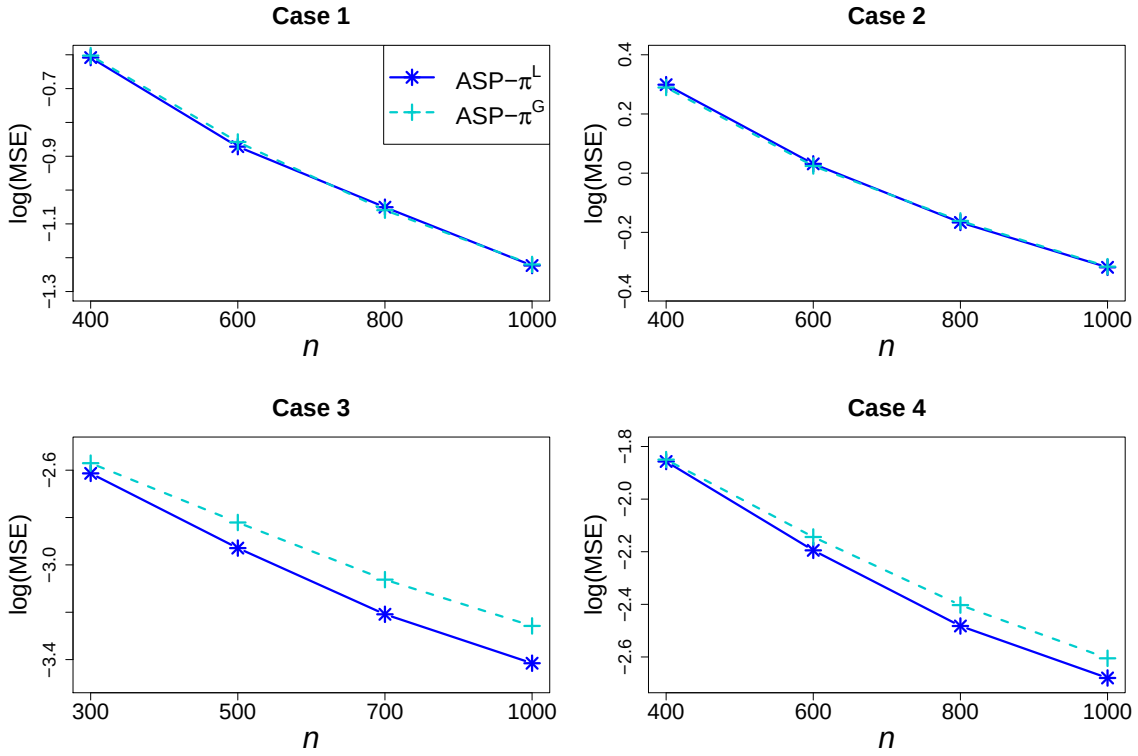


Figure A.1: Logarithm of MSEs for varied subsample size n at a fixed $n_0 = 400$ in the linear model setting. ASP-π^L uses A-optimal subsampling probabilities based on (8), and ASP-π^G uses A-optimal subsampling probabilities based on (4).

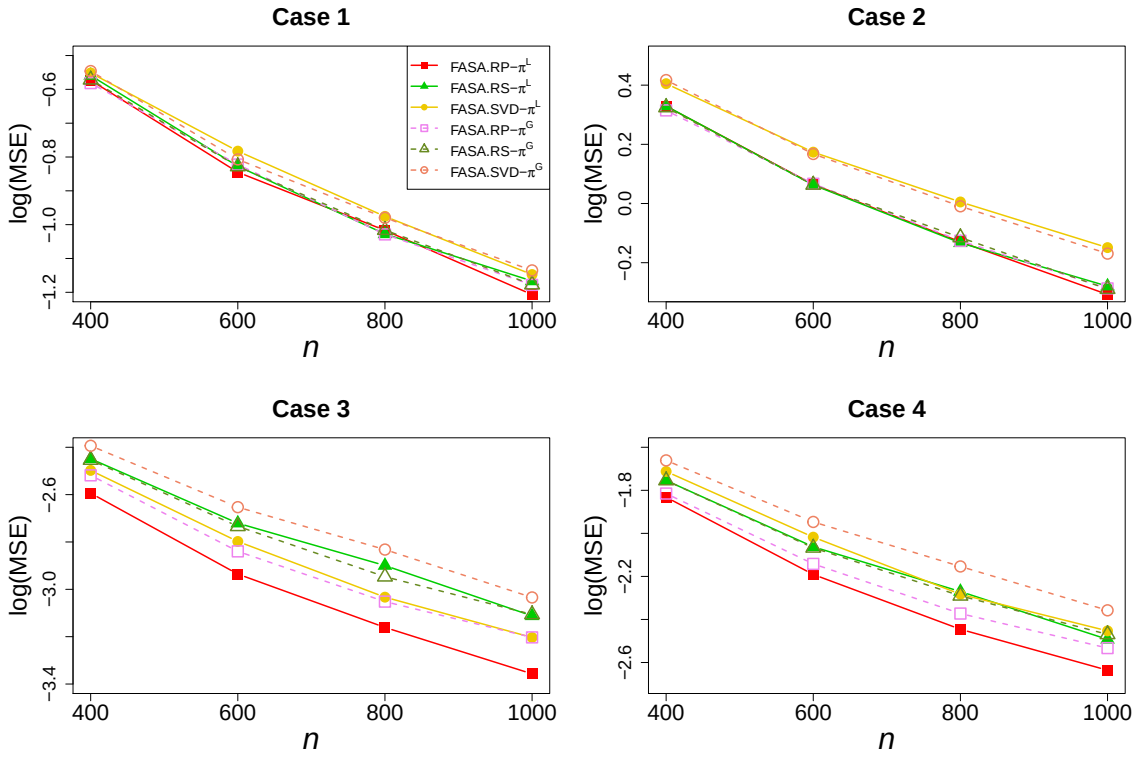


Figure A.2: Logarithm of MSEs for varied subsample size n at a fixed $n_0 = 400$ in the linear model setting. FASA.RP-π^L, FASA.RS-π^L and FASA.SVD-π^L use subsampling probabilities approximated based on the Algorithm 3, Algorithm 4 and Algorithm 5, respectively, for the A-optimal subsampling probabilities in (8). FASA.RP-π^G, FASA.RS-π^G and FASA.SVD-π^G use subsampling probabilities approximated by Algorithm 1 and 2, and the first two steps in Algorithm 5, respectively, for the A-optimal subsampling probabilities in (4).

Appendix E.2. MSEs, empirical sizes, and empirical powers for different pilot sample sizes, r_1, r_2 and r_3

We conduct extra simulations to examine the performance for different pilot sample sizes using datasets from Case 1 in linear regression and logistic regression examples. Subsample size was fixed at $n = 1000$. As shown in the left panel of Figure A.3, MSEs for ASA.RP-π^L, FASA.RS-π^L, FASA.SVD-π^L and ASP-π^L decrease as n_0 increases. The right panel of Figure A.3 presents the performance under logistic regression. ASA.RP-π^G, FASA.RS-π^G and ASP-π^L are worse than UNIF when $n_0 = 300$, but they give better results for the MSE when larger pilot sample sizes are used.

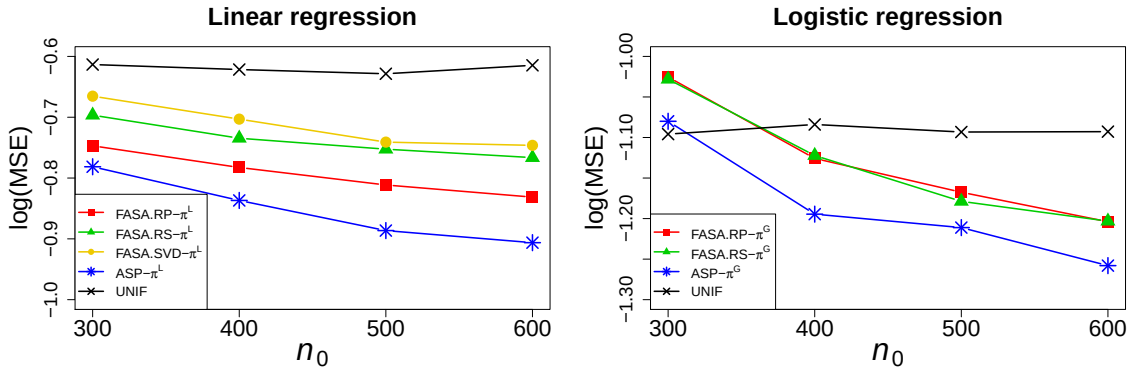


Figure A.3: Logarithm of MSEs for different pilot sample size n_0 at a fixed $n = 1000$ in the linear model and logistic model settings.

For different r_1, r_2 and r_3, we consider datasets from Case 1 in linear regression and logistic regression examples and $(r_1, r_2) = (10^2, 7), (10^3, 10), (10^4, 13)$, and we set $r_1 = r_3$.

Figure A.4 shows the results for MSE and CPU time in the linear regression example. As expected, the proposed algorithms result in smaller MSE and longer computing time as r_1 and r_2 increase. For FASA.RP-π^L and FASA.SVD-π^L, the MSE declines about 3.3 and 1 percent when (r_1, r_2) increase from $(10^2, 7)$ to $(10^3, 10)$ and from $(10^3, 10)$ to $(10^4, 13)$, respectively. However, the computing time rise about 1.2 and 1.3-fold, respectively. We also find that for FASA.RS-π^L, the MSE decreases about 5 and 0.6 percent whereas the computing time takes 1.3 and 1.4 fold longer, when (r_1, r_2) increase from $(10^2, 7)$ to $(10^3, 10)$ and from $(10^3, 10)$ to $(10^4, 13)$, respectively.

As shown in Figure A.5, similar performance is presented in the logistic regression example. When $(r_1, r_2) = (10^2, 7)$ increase to $(10^3, 10)$, FASA.RP-π^G and FASA.RS-π^G give 3.9 and 4.4 percent drop for MSE, but 1.3 and 1.2 fold increase for the computing time, respectively. When $(r_1, r_2) = (10^3, 10)$ increase to $(10^4, 13)$, the computing time for FASA.RP-π^G and FASA.RS-π^G decrease 1.2 and 1.4 fold, but the MSE decreases only 1.2 and 1.5 percent, respectively.

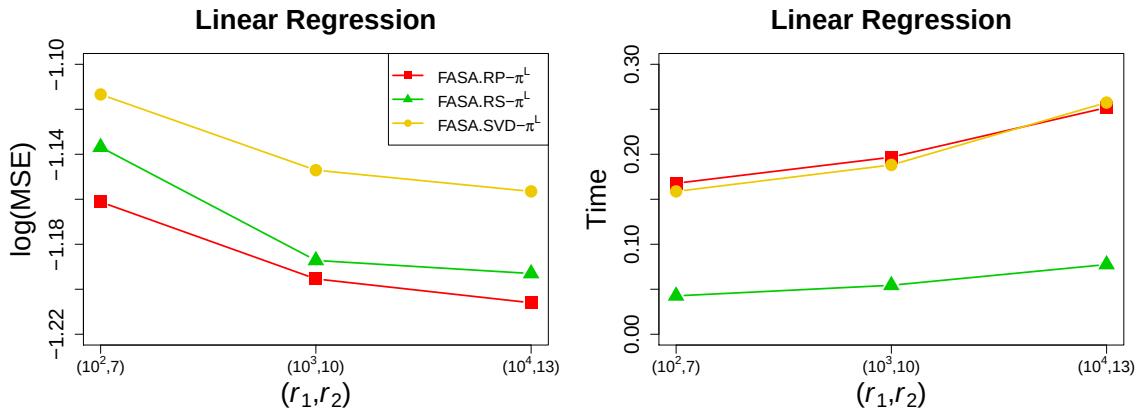


Figure A.4: Logarithm of MSEs and CPU time for varied r_1 and r_2 at a fixed $n_0 = 400, n = 1000$ in the linear model setting. For FASA.RS-π^G, r_3 is the same size as r_1.

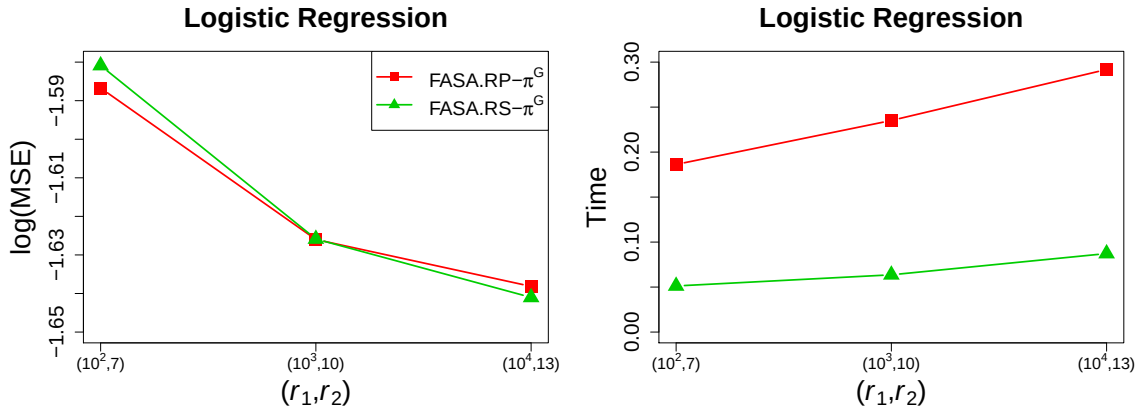


Figure A.5: Logarithm of MSEs and CPU time for varied r_1 and r_2 at a fixed $n_0 = 400, n = 1000$ in the logistic model setting. For FASA.RS-π^G, r_3 is the same size as r_1.

Appendix E.3. MSEs and empirical statistical tests for larger N and p

Table A.1: Average of MSE, and the empirical type I error for β_4 and power for β_1, β_2 and β_3 in linear model using data from Case 1 with different full data size N and number of covariates p at fixed $n = 2000$ and $n_0 = 1000$. Repetition is 300.

	$N = 2^{17}$			$N = 2^{20}$		
	$p = 50$	$p = 80$	$p = 150$	$p = 50$	$p = 80$	$p = 150$
MSE						
FASA.RP-π^L	0.2371	0.3893	0.7759	0.2345	0.3933	0.7842
FASA.RS-π^L	0.2365	0.3937	0.7795	0.2317	0.3963	0.7810
FASA.SVD-π^L	0.2465	0.4069	0.8026	0.2411	0.4137	0.8240
ASP-π^L	0.2280	0.3751	0.7552	0.2301	0.3813	0.7618
Empirical type I error						
FASA.RP-π^L	0.0600	0.0467	0.0400	0.0500	0.0233	0.0367
FASA.RS-π^L	0.0433	0.0600	0.0367	0.0267	0.0433	0.0400
FASA.SVD-π^L	0.0567	0.0567	0.0433	0.0500	0.0333	0.0600
ASP-π^L	0.0800	0.0500	0.0633	0.0267	0.0567	0.0500
Empirical power for β_1						
FASA.RP-π^L	0.3700	0.3067	0.3467	0.3800	0.3867	0.4333
FASA.RS-π^L	0.3533	0.3200	0.3400	0.2967	0.3700	0.3633
FASA.SVD-π^L	0.4000	0.2867	0.3133	0.4000	0.3967	0.3667
ASP-π^L	0.3833	0.3467	0.3467	0.4000	0.3933	0.4500
Empirical power for β_2						
FASA.RP-π^L	0.6467	0.7400	0.6667	0.7433	0.7133	0.6633
FASA.RS-π^L	0.6567	0.7400	0.6733	0.7400	0.7533	0.6600
FASA.SVD-π^L	0.6633	0.7167	0.6100	0.6833	0.7233	0.6467
ASP-π^L	0.6700	0.7633	0.6700	0.7467	0.7733	0.6800
Empirical power for β_3						
FASA.RP-π^L	0.8967	0.9300	0.8700	0.9367	0.9500	0.8900
FASA.RS-π^L	0.9333	0.9533	0.8867	0.9567	0.9500	0.9067
FASA.SVD-π^L	0.9200	0.9167	0.8667	0.9233	0.9267	0.9033
ASP-π^L	0.9033	0.9433	0.9133	0.9533	0.9367	0.9233

Table A.2: Average of MSE, and the empirical type I error for β_4 and power for β_1, β_2 and β_3 in logistic model using data from Case 1 with different full data size N and number of covariates p at fixed $n = 2000$ and $n_0 = 1000$. Repetition is 300

	$N = 2^{17}$			$N = 2^{20}$		
	$p = 50$	$p = 80$	$p = 150$	$p = 50$	$p = 80$	$p = 150$
MSE						
FASA.RP-π^G	0.1560	0.2572	0.5327	0.1502	0.2535	0.5271
FASA.RS-π^G	0.1529	0.2565	0.5385	0.1525	0.2516	0.5215
ASP-π^G	0.1503	0.2452	0.5165	0.1462	0.2446	0.5086
Empirical type I error						
FASA.RP-π^G	0.0867	0.0367	0.0600	0.0433	0.0400	0.0400
FASA.RS-π^G	0.0467	0.0467	0.0333	0.0267	0.0467	0.0400
ASP-π^G	0.0667	0.0333	0.0467	0.0433	0.0367	0.0467
Empirical power for β_1						
FASA.RP-π^G	0.3700	0.3800	0.2767	0.2967	0.2800	0.2533
FASA.RS-π^G	0.3400	0.3833	0.2700	0.2800	0.2433	0.1367
ASP-π^G	0.2967	0.3767	0.2333	0.2667	0.2533	0.2567
Empirical power for β_2						
FASA.RP-π^G	0.6633	0.6133	0.7167	0.6567	0.7067	0.6067
FASA.RS-π^G	0.6767	0.6500	0.7100	0.7000	0.6467	0.6767
ASP-π^G	0.6800	0.6733	0.7167	0.7433	0.6833	0.6700
Empirical power for β_3						
FASA.RP-π^G	0.9067	0.9467	0.9200	0.9400	0.9400	0.9067
FASA.RS-π^G	0.9267	0.9267	0.9333	0.9533	0.9333	0.9133
ASP-π^G	0.9333	0.9600	0.9533	0.9500	0.9500	0.9067

References

References

[1] Achlioptas, D., 2003. Database-friendly random projections: Johnson-lindenstrauss with binary coins. Journal of computer and System Sciences 66, 671–687.

[2] Ai, M., Yu, J., Zhang, H., Wang, H., . Optimal subsampling algorithms for big data regressions. Statistica Sinica doi:10.5705/ss.202018.0439.

[3] Drineas, P., Magdon-Ismail, M., Mahoney, M.W., Woodruff, D.P., 2012. Fast approximation of matrix coherence and statistical leverage. Journal of Machine Learning Research 13, 3475–3506.

[4] Drineas, P., Mahoney, M.W., Muthukrishnan, S., 2006. Sampling algorithms for l 2 regression and applications, in: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, pp. 1127–1136.