THE A-OPTIMAL SUBSAMPLING APPROACH TO THE ANALYSIS OF COUNT DATA OF MASSIVE SIZE

A PREPRINT

Fei Tan
Department of Mathematical Sciences, IUPUI
402 N Blackford St., LD 270, Indianapolis, IN 46202, USA
feitan@iu.edu

Xiaofeng Zhao
School of Mathematics and Statistics
North China University of Water Resources and Electric Power
136 Jinshui Road, Zhengzhou City, Henan Province, P.R. China
zxfstats@ncwu.edu.cn

Hanxiang Peng
Department of Mathematical Sciences, IUPUI
402 N Blackford St., LD 270, Indianapolis, IN 46202, USA
hanxpeng@iu.edu

March 25, 2024

ABSTRACT

The uniform and the statistical leverage-scores-based (nonuniform) distributions are frequently used in the analysis of data of massive size. Both distributions, however, are not effective in extraction of important information in data. In this article, we construct the A-optimal subsampling estimators of parameters in generalized linear models (GLM) to approximate the full-data estimators, and derive the A-optimal distributions based on the criterion of minimizing the sum of the component variances of the subsampling estimators. As the distributions have the same running time as the full-data estimator, we generalize the Scoring Algorithm introduced in Zhang, *et al.*(2023) in a Big Data linear model to GLM using the iterative weighted least squares. The paper presents a comprehensive numerical evaluation of our approach using the simulated and real data through the comparison of its performance with the uniform and the leverage-scores- subsamplings. The results exhibited that our approach substantially outperformed the uniform and the leverage-scores subsamplings and the Algorithm significantly reduced the computing time required for implementing the full-data estimator.

***K*eywords** A-optimality · Big Data · Generalized Linear Models · Negative Binomial Regression · Optimal Subsampling · Poisson Regression

1 Introduction

Big Data are data on a massive scale with regard to volume, velocity, variety, and veracity that exceed both the capacity of the conventional software tools and the memory limit of computers, see e.g. Ma, *et al.*(2015) and Fan, *et al.*(2013). Big Data pose two computational bottlenecks: (1) the sizes exceed a computer's memory, and (2) the computing task requires too long time to finish. The two bottlenecks can be simultaneously addressed by *judiciously* choosing a subsample as a surrogate for the full sample and completing the data analysis.

While the Divide-and-Conquer method easily overcomes the memory limit and often routinely amalgamates the sectional results such as by averaging (which, however, is not always mathematically justifiable), the Subsampling approach breaks the limit and saves computing time in the meantime, and gives the result with no necessity of amalgamation. Due to its mathematical simplicity and computational ease, the uniform sampling is often used for intensive computing, for the development of fast randomized algorithms, and for Monte Carlo and bootstrap. The uniform sampling, however, is not effective in extracting information in data. In this article, nonuniform sampling distributions will be sought based on the criterion of A-optimality, specifically, minimizing the trace norm of the asymptotic variance-covariance matrix (equivalently, the sum of the component variances) of the subsampling estimator.

Mathematicians, computer scientists and statisticians have already made important progress in this area. Drineas, *et al.*(2006a) constructed fast Monte Carlo algorithms to approximate matrix multiplication. Drineas, *et al.*(2006b) presented a sampling algorithm for the least squares fit problem and studied its algorithmic properties. A key feature of the foregoing algorithms is the nonuniform sampling. Ma and Sun (2014) and Ma, *et al.*(2015) explored the leverage-scores-based distribution in a Big Data linear regression model. Xu, *et al.*(2016) presented subsampled newton methods with nonuniform sampling. Liang, *et al.*(2013) constructed a resampling-based stochastic approximation for large geostatistical data. Kleiner, *et al.*(2014) proposed a scalable bootstrap for data of massive. See also the monograph by Mahoney (2011) on nonuniform random subsampling for matrix based machine learning.

Recently, Wang, *et al.*(2018) proposed the A-optimal Subsampling approach to the Big Data large logistic regression. Wang, *et al.*(2019) introduced information-based subdata selection for large linear regression. Wang, *et al.*(2022) showed that the unweighted subsampling estimators are more efficient than the weighted estimators. Ma, *et al.*(2022) conducted asymptotic analysis of sampling estimators for randomized numerical linear algebra algorithms. Zhang, *et al.*(2023) presented a systematic treatment of the A-optimal Subsampling in the framework of Big Data linear regression. The authors gave three types of the A-optimal distributions, studied the relationship to the leverage-scores-based distribution, suggested truncation which is useful for inverse probability weighted estimation, and constructed the Scoring Algorithm for fast computing–an analogue of the Scoring Method for improving estimation efficiency. Motivated by the computational burden in fitting single index models caused by high parameter dimensionality and possibly compounded by data of massive size, Smithson, *et al.*(2024) constructed the A-optimal subsampling estimators to approximate the full data estimators. The authors studied dimension asymptotics and established higher efficiency of the unweighted subsampling estimators than that of the weighted estimators.

Count data are observations of the number of occurrences of a behavior in a fixed period of time. Count data are common, for example, hospital visits, blog comments, car/bike renters, and questionnaire respondents. The scope of count data is very wide, including sociology, marketing, demographic economics, accident insurance, manufacturing defects, etc. The analysis of count data has drawn a lot of attention and been an influential part in statistical modeling. Linear regression is not an appropriate technique for count data, as it fails to take into account the limited number of possible values of the count response variable. Standard regression methods include the Poisson, the Overdispersed Poisson, the Negative Binomial, and the Zero-Inflated Poisson regressions, as well as truncated methods and the quasi-likelihood approach.

The Poisson regression and the Negative Binomial regression are often used, motivated by the ordinary consideration for regression analysis, meanwhile, seek to protect and exploit the nonnegative integer-valued characteristic of the outcome as much as possible. The Poisson regression requires distributional assumptions, which restricts its use in reality because real count data usually exhibit over-dispersion, an inflated number of zeros, an absence of certain counts, censoring counts, and missing counts. Overdisperson can be handled by generalizing the Poisson models to, for instance, quasi-Poisson models. Another useful approach is the Negative Binomial model. These models constitute the main components of generalized linear models, see, e.g., McCullagh and Nelder (1984).

The above models can deal with over-dispersion rather well, but are not enough for modeling excessive zeros. To address this problem, researchers have developed methods for zero-inflated data by including another model component to capture zero counts. This is done by a mixture model that combines a count component with a point mass at zero, see Cameron and Trivedi (2005).

The article is organized as follows. In Section 2, we construct the weighted and unweighted subsampling estimators and compare their efficiencies, derive the A-optimal distributions, and present the Scoring Algorithm. In Section 3, we review a few count regression models used in our simulations and real data applications, followed by the simulation results. Section 4 reports a real data application. Section 5 contains some supplementary tables.

Alg.1 The (Weighted) Subsampling Estimator $\hat{\boldsymbol{\beta}}^*_r$

1. Construct a distribution $\boldsymbol{\pi}$ on the data points $(\mathbf{x}_i, Y_i)$'s, use it to draw a subsample $(\mathbf{X}^*, \mathbf{Y}^*)$ of size $r << n$ and formulate the diagonal matrix $\mathbf{W}^* = \mathrm{diag}(1/r\boldsymbol{\pi}^*)$ with $\boldsymbol{\pi}^*$ the corresponding probability vector.
2. Calculate the (weighted) subsampling estimator $\hat{\boldsymbol{\beta}}^*_r$ as a solution to the weighted GEE,

$$\sum_{j=1}^{r} \frac{Y_j^* - \mu_j^*(\boldsymbol{\beta})}{\pi_j^* V_j^*(\boldsymbol{\beta})} \frac{\mathbf{x}_j^*}{g_j'^*(\boldsymbol{\beta})} = 0, \tag{2.4}$$

where $\mu_j^*(\boldsymbol{\beta}) = h(\mathbf{x}_j^{*\top}\boldsymbol{\beta}), V_j^*(\boldsymbol{\beta}) = V(\mu_j^*(\boldsymbol{\beta}))$ and $g_j'^*(\boldsymbol{\beta}) = g_j'(\mu_j^*(\boldsymbol{\beta}))$.

2 The A-optimal Subsampling In Big Data GLM

In a generalized linear regression model (GLM), the response variable Y_i and covariate vector $\mathbf{x}_i$ satisfy

$$Y_i = h(\mathbf{x}_i^\top \boldsymbol{\beta}) + \varepsilon_i, \quad i = 1, \ldots, n, \tag{2.1}$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ is an unknown parameter vector, h is the inverse of a link function g, and ε_i's are independent random errors with zero mean $E(\varepsilon_i) = 0$ and finite variance $\mathrm{Var}(\varepsilon_i) = V(\mu_i)$ for some variance function $V(\cdot)$ of the mean $\mu_i = E(Y_i)$ of Y_i. Assume that $\mathbf{x}_i$ are non-random. If $\mathbf{x}_i$'s are random, we replace the relevant assumptions with the conditional versions given $\mathbf{x}_i$'s, and the results typically hold. Let $\mu_i(\boldsymbol{\beta}) = h(\mathbf{x}_i^\top \boldsymbol{\beta})$ and $g_i(\boldsymbol{\beta}) = g(\mu_i(\boldsymbol{\beta}))$.

The parameter $\boldsymbol{\beta}$ can be estimated by the solution $\hat{\boldsymbol{\beta}}_n$ to the generalized estimating equation (GEE),

$$\sum_{i=1}^{n} \frac{Y_i - \mu_i(\boldsymbol{\beta})}{V_i(\boldsymbol{\beta})} \frac{\mathbf{x}_i}{g_i'(\boldsymbol{\beta})} = 0, \quad g_i'(\boldsymbol{\beta}) = g'(\mu_i(\boldsymbol{\beta})). \tag{2.2}$$

2.1 The (Weighted) Subsampling Estimator

When n is of massive size (often accompanied with large p), it becomes a challenging task to compute the usual $\hat{\boldsymbol{\beta}}_n$ using the conventional computers and software tools. We now take a random subsample $(\mathbf{X}^*, \mathbf{Y}^*)$ of size $r << n$ as surrogate and construct a (weighted) subsampling estimator $\hat{\boldsymbol{\beta}}^*_r$ in the algorithm in Alg. 1 to approximate $\hat{\boldsymbol{\beta}} =: \hat{\boldsymbol{\beta}}_n$.

Notation $\varepsilon_i(\boldsymbol{\beta}) = Y_i - \mu_i(\boldsymbol{\beta})$, $V_i(\boldsymbol{\beta}) = \mathrm{Var}(\varepsilon_i(\boldsymbol{\beta}))$, $\Sigma(\boldsymbol{\beta}) = \mathrm{Diag}(V_i(\boldsymbol{\beta}))$, $\dot{g}(m) = g'(m)$, $\mu_i = \mu_i(\boldsymbol{\beta}_0)$, $V_i = V_i(\boldsymbol{\beta}_0)$, $g_i = g_i(\boldsymbol{\beta}_0)$, $\varepsilon_i = Y_i - \mu_i$, $\Sigma = \mathrm{Diag}(V_i)$, $\hat{\mu}_i = \mu_i(\hat{\boldsymbol{\beta}})$, $\hat{\varepsilon}_i = Y_i - \hat{\mu}_i$, $\hat{V}_i = V_i(\hat{\boldsymbol{\beta}})$, $\hat{g}_i = g_i(\hat{\boldsymbol{\beta}})$, and $\hat{\Sigma} = \Sigma(\hat{\boldsymbol{\beta}})$. Denote by $\tilde{\eta}$ the "standarization" of η such as $\tilde{\varepsilon}_i = (Y_i - \mu_i)/\sqrt{V_i}$, and the "hat" version $\hat{\tilde{\varepsilon}}_i = (Y_i - \hat{\mu}_i)/\sqrt{\hat{V}_i}$ of $\tilde{\varepsilon}_i$.

In GLM, the hat matrix is defined as $\mathbf{H}(\boldsymbol{\beta}) = \Sigma^{1/2}(\boldsymbol{\beta})\mathbf{X}(\mathbf{X}^\top \Sigma(\boldsymbol{\beta})\mathbf{X})^{-1}\mathbf{X}^\top \Sigma^{1/2}(\boldsymbol{\beta})$. As $\mathbf{H} = \mathbf{H}(\boldsymbol{\beta}_0)$ contains the unknown parameter $\boldsymbol{\beta}_0$, one estimates it by the plug-in estimate $\hat{\mathbf{H}} = \mathbf{H}(\hat{\boldsymbol{\beta}})$. When Σ is the identity matrix, $\mathbf{H}$ simplifies to the hat matrix in a linear regression model. The hat matrix $\mathbf{H}$ in GLM possesses similar properties as the hat matrix in a linear model. Like in a linear model, the diagonal entries $h_{i,i}$ of $\mathbf{H}$ induce a sampling distribution $\boldsymbol{\ell} = (\ell_i)$ as follows:

$$\ell_i \propto h_{i,i}, \quad i = 1, \ldots, n, \tag{2.3}$$

where $\mathbf{b} \propto c_i$ denote $b_i = c_i / \sum_j c_j$ for all i. Clearly, $\boldsymbol{\ell} = (h_{i,i}/p)$ as in a linear model.

The $\hat{A}$-optimal Distributions Under suitable conditions, the subsampling estimator $\hat{\boldsymbol{\beta}}^*_r$ is asymptotically normal, i.e.,

$$\mathbf{V}_0^{-1/2}(\boldsymbol{\pi})\sqrt{r}(\hat{\boldsymbol{\beta}}^*_r - \hat{\boldsymbol{\beta}}) \Rightarrow N(0, \mathbf{I}_p), \quad \text{a.s.} \quad r \to \infty, \tag{2.5}$$

where $\hat{\mathbf{V}}(\boldsymbol{\pi})$ is the asymptotic variance-covariance matrix (abusing a bit of the parlance) given by

$$\hat{\mathbf{V}}(\boldsymbol{\pi}) = \mathrm{AVar}^*(\hat{\boldsymbol{\beta}}^*_r) = (\mathbf{X}^\top \hat{\Sigma}\mathbf{X})^{-1}(\hat{\Sigma}^{1/2}\mathbf{X})^\top \mathrm{Diag}(\hat{\tilde{\boldsymbol{\varepsilon}}}^2/r\boldsymbol{\pi})\hat{\Sigma}^{1/2}\mathbf{X}(\mathbf{X}^\top \hat{\Sigma}\mathbf{X})^{-1}. \tag{2.6}$$

Let $\mathbf{A}$ be a nonsingular $q \times p$ matrix. The plug-in estimator $\mathbf{A}\hat{\boldsymbol{\beta}}^*_r$ of the linear transformation $\mathbf{A}\boldsymbol{\beta}$ of $\boldsymbol{\beta}$ then has the asymptotic variance-covariance matrix $\mathbf{A}\hat{\mathbf{V}}(\boldsymbol{\pi})\mathbf{A}^\top$. The criterion of A-optimality is to seek a sampling distribution $\boldsymbol{\pi}$ on the data points $\{(\mathbf{x}_i, Y_i)\}$ which minimizes the trace norm $\hat{\tau}_{\mathbf{A}}(\boldsymbol{\pi}) = \mathrm{Tr}(\mathbf{A}\hat{\mathbf{V}}(\boldsymbol{\pi})\mathbf{A}^\top)$ of the matrix. Equivalently, the criterion seeks $\boldsymbol{\pi}$ to minimize the sum of the variances of the components of $\mathbf{A}\hat{\boldsymbol{\beta}}^*_r$. It is not difficult to see that

$$\hat{\tau}_{\mathbf{A}}(\boldsymbol{\pi}) =: \mathrm{Tr}(\mathbf{A}\hat{\mathbf{V}}(\boldsymbol{\pi})\mathbf{A}^\top) = \frac{1}{r}\sum_{i=1}^{n} \frac{\|\mathbf{a}_i\|^2 \hat{\tilde{\varepsilon}}_i^2}{\pi_i}, \tag{2.7}$$

where $\mathbf{a}_i = \mathbf{A}(\mathbf{X}^\top \hat{\Sigma}\mathbf{X})^{-1}\hat{\Sigma}^{1/2}\mathbf{x}_i$. Using Lagrange's multipliers, we derive

Theorem 1 *Suppose that* $\mathbf{A}$ *is independent of* $\boldsymbol{\pi}$. *Assume that* $\mathbf{X}^\top\hat{\Sigma}\mathbf{X}$ *is invertible such that* $\mathbf{A}(\mathbf{X}^\top\hat{\Sigma}\mathbf{X})^{-1}\hat{\Sigma}^{1/2}\mathbf{x}_i \neq 0$ *and the diagonal entries* $\hat{h}_{i,i}$ *of* $\hat{\mathbf{H}}$ *satisfy* $\hat{h}_{i,i} \neq 1$ *for* $i = 1, \ldots, n$. *Then there exists a unique A-optimal distribution* $\hat{\boldsymbol{\pi}}_{\mathbf{A}} = (\hat{\boldsymbol{\pi}}_{\mathbf{A},i})$ *for* $\mathbf{A}\hat{\boldsymbol{\beta}}_r^*$ *to approximate the best linear unbiased estimate* $\mathbf{A}\hat{\boldsymbol{\beta}}$ *of* $\mathbf{A}\boldsymbol{\beta}$, *which is given by*

$$\hat{\boldsymbol{\pi}}_{\mathbf{A}} \propto (\|\mathbf{a}_i\|\,|\hat{\tilde{\varepsilon}}_i|). \tag{2.8}$$

This shall be referred to as the $\hat{A}$-optimal as in Zhang, *et al.*(2023). Let

$$\mathbf{H}_\alpha = (h_{\alpha,i,j}) = \Sigma^{1/2}\mathbf{X}(\mathbf{X}^\top\Sigma\mathbf{X})^{-\alpha}\mathbf{X}^\top\Sigma^{1/2}, \quad \alpha = 0, 1, 2.$$

It can be estimated by $\hat{\mathbf{H}}_\alpha = \mathbf{H}_\alpha(\hat{\boldsymbol{\beta}})$. One has $\mathbf{H}_1 = \mathbf{H}$ and $\mathbf{H}_0 = \Sigma^{1/2}\mathbf{X}\mathbf{X}^\top\Sigma^{1/2}$. For $\mathbf{A} = (\mathbf{X}^\top\hat{\Sigma}\mathbf{X})^{1-\alpha/2}$,

$$\hat{\boldsymbol{\pi}}_{\mathbf{A}} =: \hat{\boldsymbol{\pi}}_\alpha \propto \big(\hat{h}_{\alpha,i,i}^{1/2}|\hat{\tilde{\varepsilon}}_i|\big).$$

Thus, the $\hat{A}$-optimal distribution for $\hat{\boldsymbol{\beta}}_r^*$ to approximate $\hat{\boldsymbol{\beta}}$ is the case of $\alpha = 2$, i.e.,

$$\hat{\boldsymbol{\pi}}_2 \propto \big(\hat{h}_{2,i,i}^{1/2}\,|\hat{\tilde{\varepsilon}}_i|\big). \tag{2.9}$$

Another two $\hat{A}$-optimal sampling distributions of possibly computational ease are

$$\hat{\boldsymbol{\pi}}_0 \propto \big(\|\hat{V}_i^{1/2}\mathbf{x}_i\|\,|\hat{\tilde{\varepsilon}}_i|\big), \qquad \hat{\boldsymbol{\pi}}_1 \propto \big(\hat{h}_{i,i}\,|\hat{\tilde{\varepsilon}}_i|\big). \tag{2.10}$$

In the simulated and real data, the Poisson (Poi), the Negative Binomial (NB) and the Quasipoisson (QPoi) models were used with the log-link $g(m) = \log(m)$, so that $\hat{\mu}_i = \exp(\mathbf{x}_i^\top\hat{\boldsymbol{\beta}})$ and $\hat{\tilde{\varepsilon}}_i = (Y_i - \hat{\mu}_i)/\hat{V}_i^{1/2}$, where $\hat{V}_i$ are equal to

$$\hat{\mu}_i(\text{Poi}), \quad \hat{\mu}_i(1+\hat{\alpha}\hat{\mu}_i)(\text{NB}), \quad \hat{\phi}\hat{\mu}_i(\text{QPoi}), \quad i = 1, 2, \ldots, n, \tag{2.11}$$

where $\hat{\phi}$, $\hat{\alpha}$ are estimates of ϕ, α such as the empirical estimators using the full sample in our analysis of real data.

The $\bar{A}$-optimal Distributions via Conditioning. Consider minimizing the trace norm of the conditional covariance matrix given $\mathbf{X}$. Write $\hat{\tau}_{\mathbf{A}}(\boldsymbol{\pi}) = \hat{\tau}_{\mathbf{A}}(\hat{\boldsymbol{\beta}}, \boldsymbol{\pi})$ to stress the dependence on $\hat{\boldsymbol{\beta}}$, and let let $\tilde{\tau}_{\mathbf{A}}(\boldsymbol{\pi}) = \mathrm{E}(\hat{\tau}_{\mathbf{A}}(\boldsymbol{\beta}_0, \boldsymbol{\pi})|\mathbf{X})$. We integrate out the standardized squared residuals in $\hat{\tau}_{\mathbf{A}}(\boldsymbol{\pi})$ (that is, $\mathrm{Var}(\tilde{\varepsilon}_i) = 1$) and get

$$\tilde{\tau}_{\mathbf{A}}(\boldsymbol{\pi}) = \frac{1}{r}\sum_{i=1}^n \frac{\|\mathbf{a}_{0,i}\|^2}{\pi_i} \approx \frac{1}{r}\sum_{i=1}^n \frac{\|\mathbf{a}_i\|^2}{\pi_i} =: \bar{\tau}_{\mathbf{A}}(\boldsymbol{\pi}), \tag{2.12}$$

where $\mathbf{a}_{0,i} = \mathbf{A}(\mathbf{X}^\top\Sigma\mathbf{X})^{-1}\Sigma^{1/2}\mathbf{x}_i$. Analogously, we minimize $\bar{\tau}(\boldsymbol{\pi})$ and achieve

Theorem 2 *Suppose that the assumptions in Theorem 1 hold. Then there exists a unique* $\bar{A}$*-optimal distribution* $\bar{\boldsymbol{\pi}}_{\mathbf{A}}$ *for* $\mathbf{A}\hat{\boldsymbol{\beta}}_r^*$ *to approximate the BLUE* $\mathbf{A}\hat{\boldsymbol{\beta}}$ *of the linear transformation* $\mathbf{A}\boldsymbol{\beta}$, *given by*

$$\bar{\boldsymbol{\pi}}_{\mathbf{A}} \propto (\|\mathbf{a}_i\|). \tag{2.13}$$

The $\bar{A}$-optimal distribution for $\hat{\boldsymbol{\beta}}_r^*$ to approximate the full-sample estimator $\hat{\boldsymbol{\beta}}$ is now given by

$$\bar{\boldsymbol{\pi}}_2 \propto (\hat{h}_{2,i,i}^{1/2}). \tag{2.14}$$

Another two alternative $\bar{A}$-optimal distributions of possibly computational ease are

$$\bar{\boldsymbol{\pi}}_0 \propto (\|\hat{V}_i^{1/2}\mathbf{x}_i\|), \quad \bar{\boldsymbol{\pi}}_1 \propto (\hat{h}_{i,i}^{1/2}). \tag{2.15}$$

Truncation Observe that (2.8) implies that the i-th data point $(\mathbf{x}_i, Y_i)$ must be drawn with probability $\hat{\pi}_{\mathbf{A},i}$ proportional to the i-th standardized residual $|\hat{\tilde{\varepsilon}}_i|$. Since each probability is inversely used in constructing $\hat{\boldsymbol{\beta}}_r^*$, $\hat{\boldsymbol{\pi}}_{\mathbf{A}}$ must be truncated from below in order to guarantee appropriate statistical properties for $\hat{\boldsymbol{\beta}}_r^*$. Here we follow Zhang, *et al.*(2023) to truncate $\hat{\boldsymbol{\pi}}_{\mathbf{A}} = (\hat{\boldsymbol{\pi}}_{\mathbf{A},i})$ from below by L/n, and define $\hat{\boldsymbol{\pi}}_{\mathbf{A}}(l)$ as

$$\hat{\boldsymbol{\pi}}_{\mathbf{A}}(l) \propto \big(\hat{\pi}_{\mathbf{A},i}\mathbf{1}[\hat{\pi}_{\mathbf{A},i} \geq L/n] + (l/n)\mathbf{1}[\hat{\pi}_{\mathbf{A},i} < L/n]\big),$$

where L is a threshold value, and typically $0 < L \leq 1$. As pointed out by the above authors, we may drop "unimportant" observations by taking $l = 0$ for fast computing, otherwise $l = L$. To determine the value of L, we must take it into consideration the desired running time and the accuracy. Our extensive simulated and real data exhibited that the truncation led to only a slight loss of efficiency.

Alg.2 The Scoring Algorithm

1. Take a uniform pre-subsample $(\mathbf{X}_0^*, \mathbf{y}_0^*)$ of size $r_0 \ll r \ll n$ from $(\mathbf{X}, \mathbf{y})$, and use it to compute $\mathbf{H}_{0,\alpha}^*$ and $\hat{\varepsilon}_0^*$ described in (2.18) and the distribution $\boldsymbol{\pi}_{0,\alpha}^* = (\mathrm{diag}(\mathbf{H}_{0,\alpha,k}^*) : k = 1, \ldots, K)$ for given α.
2. Call the algorithm in Alg. 1 with the subsample size r and the sampling distribution $\boldsymbol{\pi}_{0,\alpha}^*$.

The Scoring Algorithm Like a typical non-uniform subsampling, the optimal distributions $\hat{\boldsymbol{\pi}}_\alpha$ and $\bar{\boldsymbol{\pi}}_\alpha, \alpha = 0, 1, 2$ have the same running time as the full-data estimator $\hat{\boldsymbol{\beta}}$. Here we generalize it to GLM as described in Alg. 2 the Scoring Algorithm introduced for the A-optimal subsampling in a Big Data linear regression model by Zhang, *et al.*(2023) as follows.

One advantage of GLM is that the estimator $\hat{\boldsymbol{\beta}}$ can be found by the iterative weighted least squares estimate (IWLSE). Specifically, we rewrite (2.2) in a matrix form,

$$\mathbf{X}^\top \mathbf{W}(\boldsymbol{\beta}) \dot{\mathbf{g}}(\boldsymbol{\beta})(\mathbf{Y} - \boldsymbol{\mu}(\boldsymbol{\beta})) = 0, \tag{2.16}$$

where $\boldsymbol{\mu}(\boldsymbol{\beta}) = (\mu_1, \ldots, \mu_n)^\top|_{\boldsymbol{\beta}}$, $\dot{\mathbf{g}}(m) = \mathrm{Diag}(g_1', \ldots, g_n')|_m$ and $\mathbf{W}(\boldsymbol{\beta}) = \mathrm{Diag}(1/V_1 g_1^{'2}, \ldots, 1/V_n g_n^{'2})|_{\boldsymbol{\beta}}$. Let

$$\mathbf{Z}^{(0)} = \mathbf{X}\boldsymbol{\beta}^{(0)} + \dot{\mathbf{g}}^{(0)}(\mathbf{Y} - \boldsymbol{\mu}^{(0)}),$$

where $\boldsymbol{\beta}^{(0)}$ is an initial value (which is automatically provided in the *R* package), $\mathbf{W}^{(0)} = \mathbf{W}(\boldsymbol{\beta}^{(0)})$, $\dot{\mathbf{g}}^{(0)} = \dot{\mathbf{g}}(\boldsymbol{\beta}^{(0)})$ and $\boldsymbol{\mu}^{(0)} = \boldsymbol{\mu}(\boldsymbol{\beta}^{(0)})$. The estimate $\hat{\boldsymbol{\beta}}$ can now be obtained by a few iterations of the weighted least squares. Formally,

$$\boldsymbol{\beta}^{(1)} = (\mathbf{X}^\top \mathbf{W}^{(0)} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}^{(0)} \mathbf{Z}^{(0)}. \tag{2.17}$$

Since the computational bottleneck is to calculate the matrix $\mathbf{X}^\top \mathbf{W}^{(0)} \mathbf{X}$, which takes $O(n(p+d)^2)$ time, we shall approximate it by a subsampling matrix $\mathbf{X}_0^{*\top} \mathbf{W}_0^{(0)*} \mathbf{X}_0^*$ based on a computationally easy pre-subsample $(\mathbf{X}_0^*, \mathbf{y}_0^*)$ from the full data $(\mathbf{X}, \mathbf{Y})$. The same consideration applies to $\mathbf{X}^\top \mathbf{W}^{(0)} \mathbf{Z}^{(0)}$, resulting in the pre-subsample estimator,

$$\hat{\boldsymbol{\beta}}_0^* = (\mathbf{X}_0^{*\top} \mathbf{W}_0^{(0)*} \mathbf{X}_0^*)^{-1} \mathbf{X}_0^{*\top} \mathbf{W}_0^{(0)*} \mathbf{Z}_0^{(0)*}, \quad \hat{\boldsymbol{\Sigma}}_0 = \boldsymbol{\Sigma}(\hat{\boldsymbol{\beta}}_0^*).$$

While a uniform pre-subsample is almost immediate, more efficient pre-subsamples are in fact possible. When the $n \times p$ matrix $\mathbf{X}$ exceeds the memory limit, one may break $\mathbf{X}$ into K submatrices $\mathbf{X}_k$ of lower dimension $n_k \times p$, compute

$$\hat{\varepsilon}_{0,k}^* = \mathbf{Y}_k - \mathbf{X}_k \hat{\boldsymbol{\beta}}_0^*, \quad \mathbf{H}_{0,\alpha,k}^* = \hat{\boldsymbol{\Sigma}}_{0,k}^{1/2} \mathbf{X}_k (\mathbf{X}_0^{*\top} \hat{\boldsymbol{\Sigma}}_0^* \mathbf{X}_0^*)^{-\alpha} \mathbf{X}_k^\top \hat{\boldsymbol{\Sigma}}_{0,k}^{1/2}, \quad k = 1, \ldots, K, \tag{2.18}$$

and 'amalgamate' them to get $\hat{\varepsilon}_0^*$ and $\mathbf{H}_{0,\alpha}^*$ for computing the distributions in (2.9)-(2.10) and (2.14)-(2.15), where $\mathbf{Y}_k, \hat{\boldsymbol{\Sigma}}_{0,k}$ are defined in an obvious way and $\hat{\boldsymbol{\Sigma}}_0^*$ is the corresponding subsampling matrix from $\hat{\boldsymbol{\Sigma}}_0$. One doesn't really amalgamate, but extract the diagonal entries to obtain the sampling distribution. These details are summarized in Alg. 2. Our extensive simulated and real data in Sections 3–4 exhibited that the Scoring Algorithm in Alg. 2 worked well.

2.2 The Unweighted Subsampling Estimator and its Efficiency

Analogously, we construct the unweighted subsampling estimator $\tilde{\boldsymbol{\beta}}^*$ via calling the algorithm in Alg. 1 in which we set $\pi_j^* = 1$ for all j. In particular, $\tilde{\boldsymbol{\beta}}^*$ solves the unweighted GEE,

$$\mathbf{G}_n^*(\boldsymbol{\beta}) = \sum_{j=1}^r \mathbf{g}_j^*(\boldsymbol{\beta}) =: \sum_{j=1}^r \frac{Y_j^* - \mu_j^*(\boldsymbol{\beta})}{V_j^*(\boldsymbol{\beta})} \frac{\mathbf{x}_j^*}{g_j'^*(\boldsymbol{\beta})} = 0. \tag{2.19}$$

Let $\mathbf{G}_n(\boldsymbol{\beta}, \boldsymbol{\pi}) = \mathbf{E}^*(\mathbf{G}_n^*(\boldsymbol{\beta}))$, and let $\tilde{\boldsymbol{\beta}}$ be a solution to $\mathbf{G}_n(\boldsymbol{\beta}, \boldsymbol{\pi}) = 0$, that is, $\tilde{\boldsymbol{\beta}}$ solves the weighted GEE,

$$\mathbf{G}_n(\boldsymbol{\beta}, \boldsymbol{\pi}) = \sum_{i=1}^n \pi_i \mathbf{g}_i(\boldsymbol{\beta}) =: \sum_{i=1}^n \pi_i \frac{Y_i - \mu_i(\boldsymbol{\beta})}{V_i(\boldsymbol{\beta})} \frac{\mathbf{x}_i}{g_i'(\boldsymbol{\beta})} = 0. \tag{2.20}$$

The $\tilde{\boldsymbol{\beta}}$ is the *generalized bootstrap estimator* for the GEE studied by Chatterjee and Bose (2002), who proved the asymptotic normality for growing parameter dimension. Let $\tilde{\mathbf{H}}(\boldsymbol{\pi}) = \partial \mathbf{G}_n(\tilde{\boldsymbol{\beta}}, \boldsymbol{\pi})/\partial \boldsymbol{\beta}$ be the Hessian matrix, and write $\tilde{\mathbf{g}}_i = \mathbf{g}_i(\tilde{\boldsymbol{\beta}})$. Then $\tilde{\mathbf{G}}_n^* = \sum_{j=1}^r \tilde{\mathbf{g}}_j^*$ with $\mathbf{E}^*(\tilde{\mathbf{G}}_n^*) = \mathbf{G}_n(\tilde{\boldsymbol{\beta}}, \boldsymbol{\pi}) = 0$. As a result,

$$\tilde{\boldsymbol{\Sigma}} = \tilde{\boldsymbol{\Sigma}}(\boldsymbol{\pi}) = \mathrm{Var}^*(\tilde{\mathbf{G}}_n^*) = r \sum_{i=1}^n \pi_i \tilde{\mathbf{g}}_i^{\otimes 2}. \tag{2.21}$$

Let $\tilde{\mathbf{V}}(\boldsymbol{\pi}) = \mathrm{AVar}^*(\hat{\boldsymbol{\beta}}_r^*) = \tilde{\mathbf{H}}^{-1}(\boldsymbol{\pi})\tilde{\boldsymbol{\Sigma}}(\boldsymbol{\pi})\tilde{\mathbf{H}}^{-\top}(\boldsymbol{\pi})$. Under suitable conditions, for any sampling distribution $\boldsymbol{\pi}$,

$$\tilde{\mathbf{V}}^{-1/2}(\boldsymbol{\pi})\sqrt{r}(\tilde{\boldsymbol{\beta}}_r^* - \tilde{\boldsymbol{\beta}}) \Rightarrow N(0, \mathbf{I}_p), \quad a.s. \tag{2.22}$$

Wang, *et al.*(2023) proved that the unweighted subsampling estimators in GLM are more efficient than the weighted estimator. Smithson, *et al.*(2024) demonstrated that the unweighted subsampling estimators of parameters in the penalized single index model (SIM) are asymptotically more efficient than the weighted estimators on an event whose probability tends to one as r tends to infinity. As the SIM extends the GLM by allowing the link to be unknown, the latter is obviously a special case of the former. Thus, the result in Smithson, *et al.*(2024) applies to the GLM, specifically, for any $\mathbf{u}$, as $r \to \infty$,

$$\mathbf{u}^\top \big(\tilde{\mathbf{V}}^{-1}(\boldsymbol{\pi}, \tilde{\boldsymbol{\beta}}^*) - \hat{\mathbf{V}}^{-1}(\boldsymbol{\pi}, \hat{\boldsymbol{\beta}}^*)\big)\mathbf{u} \ge o_P(1),$$

for an arbitrary sampling distribution $\boldsymbol{\pi}$ which is independent of the random errors ε_i. Clearly, the $\bar{A}$-optimal distribution $\bar{\boldsymbol{\pi}}_{\mathbf{A}}$ in (2.13) including $\bar{\boldsymbol{\pi}}_2$ in (2.14) are independent of the random errors. The authors exhibited that the result holds with no restriction on the relationship of r and n under some typical boundedness conditions.

3 A Large Simulation Study

In this Section, we first review count data regression models used in our analysis of simulated and real data, followed by the simulation results.

3.1 Count Data Regression Models

The Poisson Model Let Y have a Poisson distribution with mean μ, $\mathrm{Poi}(\mu)$, i.e., the probability mass function (pmf) is

$$f_{\mathrm{poi}}(y; \mu) = e^{-\mu}\mu^y/y!, \quad y = 0, 1, 2, \ldots \tag{3.1}$$

The mean and variance are equal, $\mathrm{Var}(Y) = \mu = E(Y)$. In real-life data, however, the equality is usually not met, which is termed as *overdisperson* in the literature.

In the presence of overdispersion, the standard errors (SE) of the estimates in Poisson regression model are deflated, leading to exaggerated test statistic values for parameters and false significant findings accordingly. Overdispersion can often be tested by the usual goodness-of-fit statistic. In our real data analysis, we should perform such tests. An alternative option to handle overdispersion is

The Negative Binomial Model Let Y have a Negative Binomial with mean μ and overdispersion parameter $\alpha > 0$, $\mathrm{Nb}(\mu, \alpha)$, i.e., with the pmf,

$$f_{\mathrm{nb}}(y; \mu, \alpha) = \frac{\Gamma(y + 1/\alpha)}{\Gamma(1/\alpha)y!}(1 + \alpha\mu)^{-1/\alpha}(\mu/(\mu + 1/\alpha))^{-y}, \; y = 0, 1, 2, \ldots \tag{3.2}$$

Then $E(Y) = \mu$ and $\mathrm{Var}(Y) = \mu + \alpha\mu^2$ satisfy $\mathrm{Var}(Y) \ge E(Y)$, and $\mathrm{Var}(Y) = E(Y)$ if and only if $\alpha = 0$.

The Quasi-likelihood Model Another popular option to handle overdispersion is the *quasi-likelihood model*. This has the advantage of requiring only to specify the mean and variance but not a distribution for the response Y. Specifically, the statistical inference is based on the quasi-likelihood equation,

$$\sum_{i=1}^n \frac{y_i - \mu_i(\boldsymbol{\beta})}{V_i(\boldsymbol{\beta}, \phi)} h'(\mathbf{x}_i^\top \boldsymbol{\beta})\mathbf{x}_i = 0, \tag{3.3}$$

where $\mu_i(\boldsymbol{\beta}) = E(Y_i|\mathbf{x}_i)$ and $V_i(\boldsymbol{\beta}, \phi) = \mathrm{Var}(Y_i|\mathbf{x}_i)$ are the mean and variance functions to be specified, and ϕ is an overdisperson parameter,

The quasi-likelihood model has great flexibility and unifies several models in the sense that the maximum likelihood estimate (MLE) of the models are special cases. Setting $V_i = \mu_i$, Eqt (3.3) is the estimating equations for the MLE of the parameters in the Poisson model. Setting $V_i = \mu_i(1 + \alpha\mu_i)$ with $\phi = \alpha$, Eqt (3.3) is the estimating equations for the MLE of the parameters in the Negative Binomial model. Another frequent choice of the variance for overdispersion is $V_i = \phi\mu_i$ with $\phi > 0$. All the three cases can be unified with the form of $V_i = \mu_i + \alpha\mu_i^p$ for $p = 1, 2$.

The Zero-Inflated Poisson Model In many real count data, there is an excess of zero counts for which the Poisson model can not account. Consider a mixture model of a degenerate distribution at 0 and a Poisson distribution,

$$f_{\mathrm{zip}}(y; \mu, \rho) = \rho f_0(y) + (1 - \rho)f_{\mathrm{poi}}(y; \mu), \quad y = 0, 1, 2, \ldots, \tag{3.4}$$

where $f_0(y) = \mathbf{1}[y = 0]$ is the point mass at zero to account for structural zeros. Since

$$f_{\text{zip}}(0; \mu, \rho) = \rho + (1 - \rho) \exp(-\mu),$$

it follows from $0 \leq f_{\text{zip}}(0; \mu, \rho) \leq 1$ that $1/(1 - \exp(\mu)) \leq \rho \leq 1$. This shows that ρ can be negative. A positive ρ represents that the probability of structural zeros is above the expected number of zeros under the Poisson f_{poi}, and a negative ρ represents that the probability is below the expected number. The MLE $\hat{\boldsymbol{\beta}}$ solves the GEE,

$$\sum_{i=1}^{n} \frac{f_{\text{poi}}(y_i; \mu_i)}{f_{\text{zip}}(y_i; \mu_i, \rho)} \frac{y_i - \mu_i(\boldsymbol{\beta})}{\mu_i(\boldsymbol{\beta})} h'(\mathbf{x}_i^\top \boldsymbol{\beta}) \mathbf{x}_i = 0. \tag{3.5}$$

To estimate ρ, one can find another equation by differentiating the log likelihood w.r.t. ρ. For simplicity, we shall estimate ρ by the sample percentage $\hat{\rho}$ of the zero observations in data. Substituting $\hat{\rho}$ in (3.5), we then solve for $\hat{\boldsymbol{\beta}}$.

3.2 The Simulation Results

In our simulations, the covariates $\mathbf{x}_i$ were generated from each of next four distributions. (GA) The Gaussian $N(0, \boldsymbol{\Sigma})$ with $\Sigma_{i,j} = 0.3^{|i-j|}$; (MG) The Mixture Gaussian $\frac{1}{2}N(0, \Sigma) + \frac{1}{2}N(0, 3\Sigma)$; (LN) The Log-normal $LN(0, \frac{1}{2}\Sigma)$; (T_5) The student t with 5 degrees of freedom $\mathbf{T}_5(0, \frac{1}{2}\Sigma)$. The responses Y_i were generated from the Poisson and the Negative Binomial models with the variance structure $\text{Var}(Y_i) = \mu_i + 5\mu_i^2$. We used the logarithmic link, so that $\mu_i = E(Y_i) = \exp(\mathbf{x}_i^\top \boldsymbol{\beta}_0)$. We chose $n = 50,000$, $p = 50$, and $\boldsymbol{\beta}_0 = (0.1, -0.1 \times \mathbf{1}_{p/2}^\top, 0.1 \times \mathbf{1}_{p/2}^\top)^\top$.

For the Poisson and the Negative Binomial models, we used each of the $\hat{A}$-optimal distributions given in (2.9)–(2.10) together with the estimates of unknown quantities given in (2.11) and the $\bar{A}$-optimal distributions in (2.14)-(2.15). A subsample of size r was drawn, and the subsampling estimator $\hat{\boldsymbol{\beta}}_r^*$ then calculated using the algorithm in Alg. 1. We repeated the process $B = 1,000$ times and computed the empirical mean squared errors (EMSE) as follows:

$$\text{EMSE} = \frac{1}{B} \sum_{b=1}^{B} \|\hat{\boldsymbol{\beta}}_{r,b}^* - \hat{\boldsymbol{\beta}}\|^2. \tag{3.6}$$

where $\hat{\boldsymbol{\beta}}_{r,b}^*$ is the subsampling estimator calculated on the b^{th} subsample of size r. The results are reported as figures and tables below (in Section 5 as supplementary tables) for the Poisson model (the Negative Binomial model), where the symbols hpi0($\hat{\boldsymbol{\pi}}_0$), hpi1($\hat{\boldsymbol{\pi}}_1$), ..., bpi2($\bar{\boldsymbol{\pi}}_2$) are used. We summarized the results below.

Efficiency Comparison Reported in Table 1 are the simulated EMSE (and their ratios) of the weighted $\bar{\boldsymbol{\pi}}_2$- subsampling estimator $\hat{\boldsymbol{\beta}}^*$ and the unweighted estimator $\tilde{\boldsymbol{\beta}}^*$. We generated $\mathbf{X}$ from the four distributions and Y from the Poisson model. The MSE are calculated by the formula (3.6) in which $\hat{\boldsymbol{\beta}}$ is replaced with the true value $\boldsymbol{\beta}_0$. As pointed out in Subsection 2.2, $\bar{\boldsymbol{\pi}}_2$ given in (2.14) satisfies the condition for the unweighted subsampling estimator $\tilde{\boldsymbol{\beta}}^*$ to be more efficient than the weighted estimator $\hat{\boldsymbol{\beta}}^*$. We chose $r_0 = 500$ in the Scoring Algorithm in Alg. 2 to calculate an approximation to $\bar{\boldsymbol{\pi}}_2$. As the sample size $n = 50,000$ exceeds our laptop, we broke the full data $\mathbf{X}$ of dimension $50,000 \times 50$ into five chunks $\mathbf{X}_k$ of dimension $10,000 \times 50$ to calculate the approximation. The MSE were calculated for a few subsample sizes r based on 500 repetitions. One can see that the ratios of the simulated EMSE were significantly less than one with the smallest ratios about less than 40% with $\mathbf{X}$ generated from $\mathbf{T}_5(0, \frac{1}{2}\Sigma)$, suggesting that the unweighted estimator was substantially more efficient than the weighted estimator.

Variability of $\hat{A}$- and $\bar{A}$-optimal Distributions Reported in Fig. 1 (Fig. 5) are the boxplots of the probabilities of six optimal distributions using the responses Y_i generated from the Poisson model (the Negative Binomial model). In each plot, all $\hat{\boldsymbol{\pi}}_k$ were more spread out than all $\bar{\boldsymbol{\pi}}_k$, but the medians of $\bar{\boldsymbol{\pi}}_k$ were slightly bigger than those of $\hat{\boldsymbol{\pi}}_k$ for $k = 0, 1, 2$.

EMSE Reported in Fig. 2 (Fig.6) are the plots of the log (EMSE) against subsmaple size r in the Poisson model (the Negative Bionomial model). For the four datasets, the EMSE was decreasing with the increasing r. Both $\hat{\boldsymbol{\pi}}_k$ and $\bar{\boldsymbol{\pi}}_k$ had smaller EMSE than the uniform; the $\hat{A}$-sampling outperformed the $\bar{A}$-sampling; $\hat{\boldsymbol{\pi}}_2$ was the best among $\hat{\boldsymbol{\pi}}_k$ and $\bar{\boldsymbol{\pi}}_2$ was the best among $\bar{\boldsymbol{\pi}}_k$ for $k = 0, 1, 2$.

Coverage Probability Reported in Fig. 3 (Fig. 7) are the plots of the simulated percentages of the 95% confidence intervals catching the true value of the coefficient β_2 against the subsample size r based on $2,000$ repetitions, with the responses Y_i generated from the Poisson model (the Negative Binomial model). The confidence intervals were calculated using the formula $\hat{\beta}_{2,r}^* \pm Z_{0.975} SE(\hat{\beta}_{2,r}^*)$ with $SE(\hat{\beta}_{2,r}^*) = \sqrt{\hat{\mathbf{V}}_{22}}$. Fig. 3 exhibited that when the subsample size r was small, the coverage probabilities were lower than the nominal level 95%, and were closer to the nominal level with the increasing r. Except for GA and LN, the coverage probabilities under the $\hat{\boldsymbol{\pi}}_2$- and $\bar{\boldsymbol{\pi}}_2$- subsampling were closer to the nominal level than the uniform subsampling.

EMSE Ratio Reported in Table 3 (Table 18) are the ratios of the EMSE of the A-optimal subsampling estimators to the EMSE of the uniform subsampling estimator, with the responses Y_i generated from the Poisson model (the Negative Binomial model). (1) All the ratios in the Tables were less than one, indicating all the optimal subsampling outperformed the uniform. (2) $\hat{\boldsymbol{\pi}}_k$ outperformed $\bar{\boldsymbol{\pi}}_k$, and $\hat{\boldsymbol{\pi}}_2$ was superior to all others. (3) The EMSE ratios using the truncated $\hat{\boldsymbol{\pi}}_k$ and $\bar{\boldsymbol{\pi}}_k$ resulted in only slight loss of efficiency compared to those using the untruncated ones for $k = 0, 1, 2$. This property is useful in the Analysis of Big Data because the loss of efficiency would be small when dropping unimportant observations for fast computing according to whether the sampling probabilities are less than certain threshold value. (4) Truncation is necessary to guarantee the theoretical properties of the subsampling estimators, see Zhang, *et al.*(2023).

Reported in Table 8 (Table 19) are the EMSE ratios where the Scoring Algorithm was used. We first chose a uniform pre-subsample of size $r_0 = 500$; obtained an initial estimator $\hat{\boldsymbol{\beta}}^*_{r_0}$ to approximate $\hat{\boldsymbol{\beta}}$; then approximated the sampling distributions and used them to draw subsamples; calculated the subsampling estimator $\hat{\boldsymbol{\beta}}^*_r$ in the end. Observe that the Scoring Algorithm saved significant amount of time while the loss of efficiency was marginal.

Running Time Reported in Tables 4–5 are the running times for computing the $\hat{\boldsymbol{\pi}}_2$- and $\bar{\boldsymbol{\pi}}_2$-subsampling estimators $\hat{\boldsymbol{\beta}}^*_r$, using the statistical computing package *R*. Those times were computed on a desktop with Intel i5 processor and 8GB memory. We recorded the CPU times for 1000 repetitions, then took the average of the times for fair comparison. It is noteworthy that although $\hat{\boldsymbol{\pi}}_2$ spent longer computing time than $\bar{\boldsymbol{\pi}}_2$, all the proposed methods spent significant less computing time than for computing the full-sample estimator. One found in Table 2 that the proposed Subsampling approach had similar number of iterations, indicating that small subsample sizes did not necessarily increase the iterations in using Newton's algorithms.

Figure 1: The boxplots of log(probability) of the A-optimal distributions with Y_i generated from the Poisson model and the full-sample estimator $\hat{\boldsymbol{\beta}}$ with $n = 50,000$ & $p = 50$.

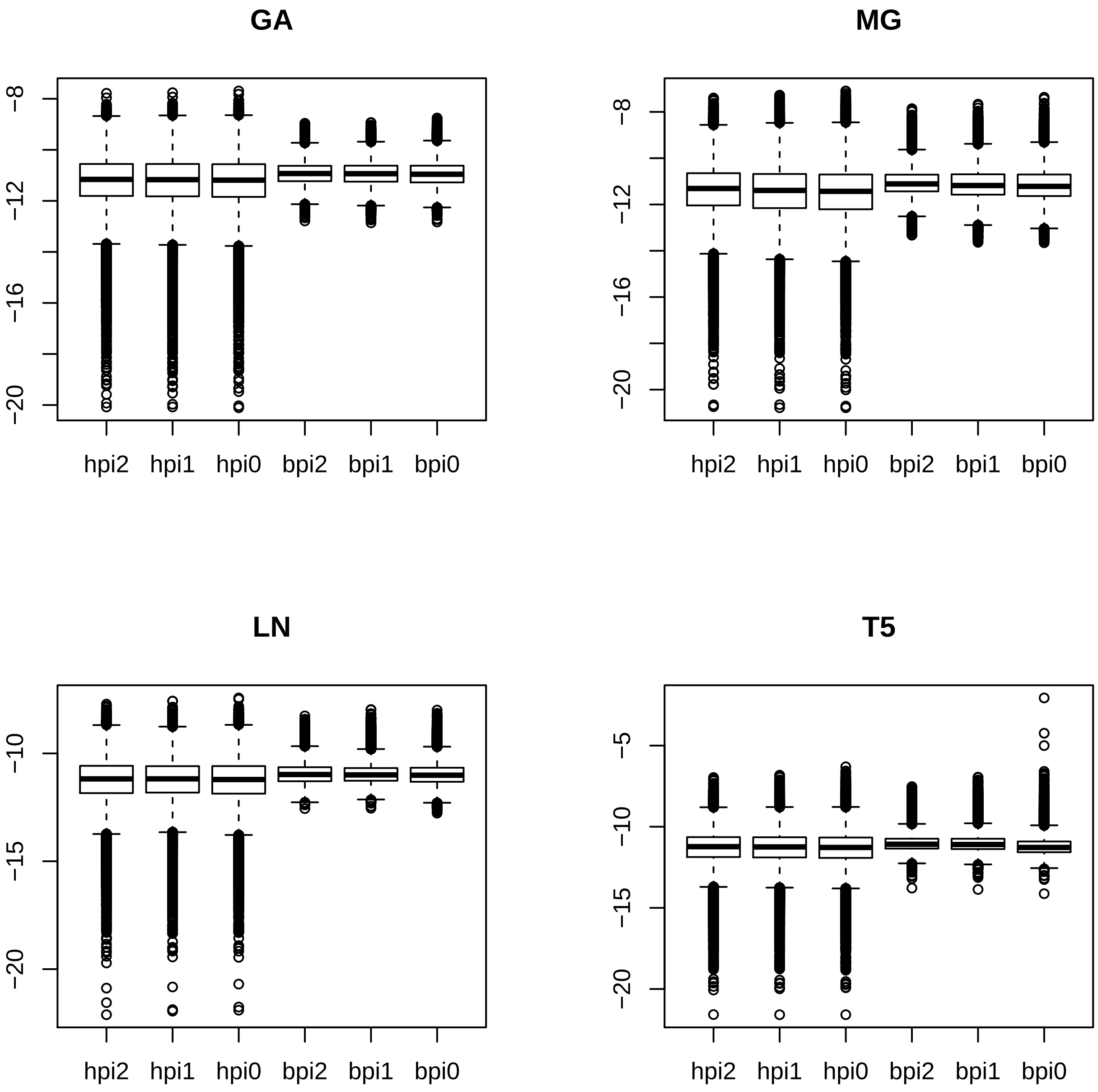


Figure 2: The plots of log (EMSE) of the subsampling estimate $\hat{\boldsymbol{\beta}}_r^*$ under different samplings against the subsample size r with Y_i generated from the Poisson model and the full-sample estimate $\hat{\boldsymbol{\beta}}$ with $n = 50,000$ & $p = 50$.

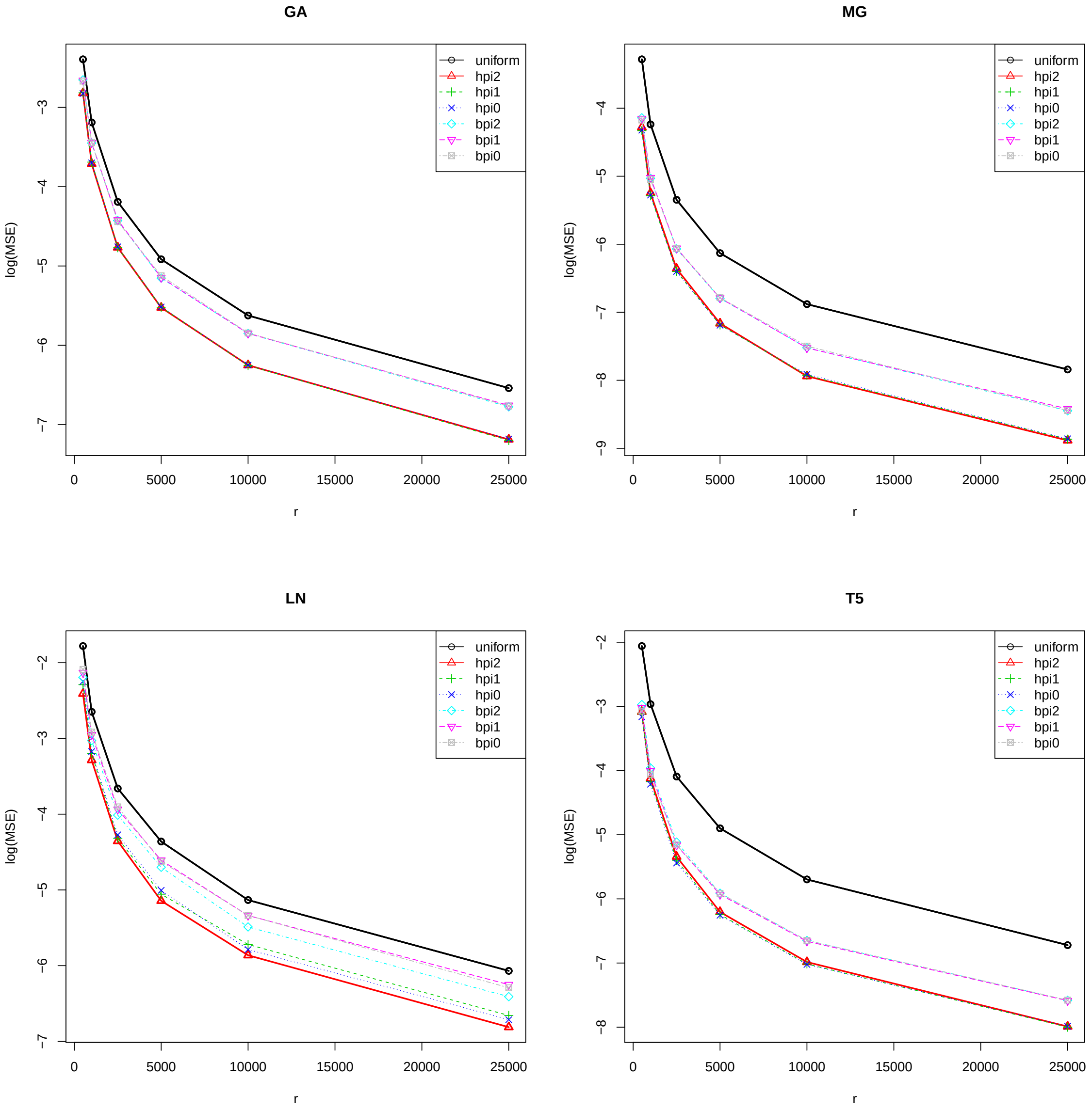


Table 1: The simulated EMSE (and their ratios) of the unweighted (UW) and weighted (W) $\bar{\boldsymbol{\pi}}_2$- subsampling estimators $\tilde{\boldsymbol{\beta}}^*$ and $\hat{\boldsymbol{\beta}}^*$ with the pre-subsample size $r_0 = 500$, $p = 50$ and $n = 50,000$.

r	1200	1400	1600	1800	2000	1200	1400	1600	1800	2000
	GA					MG				
UW	.4261	.3620	.3139	.2705	.2458	.0577	.0493	.0423	.0375	.0346
W	.5361	.4592	.3987	.3460	.3099	.1070	.0935	.0802	.0704	.0625
Ratio	.7948	.7883	.7873	.7818	.7932	.5396	.5275	.5271	.5334	.5547
	LN					T_5				
UW	.5507	.4887	.4304	.3706	.3617	.1173	.0994	.0791	.0709	.0630
W	.8203	.7078	.6421	.5668	.5206	.3031	.2580	.2166	.1920	.1793
Ratio	.6713	.6904	.6703	.6538	.6948	.3871	.3854	.3653	.3692	.3514

Figure 3: The plots of the simulated percentages of the 95% confidence intervals catching the true coefficient β_2 under different samplings against the subsample size r with Y_i generated from the Poisson model with pre-subsample size $r_0 = 500$, $n = 50,000$ & $p = 50$.

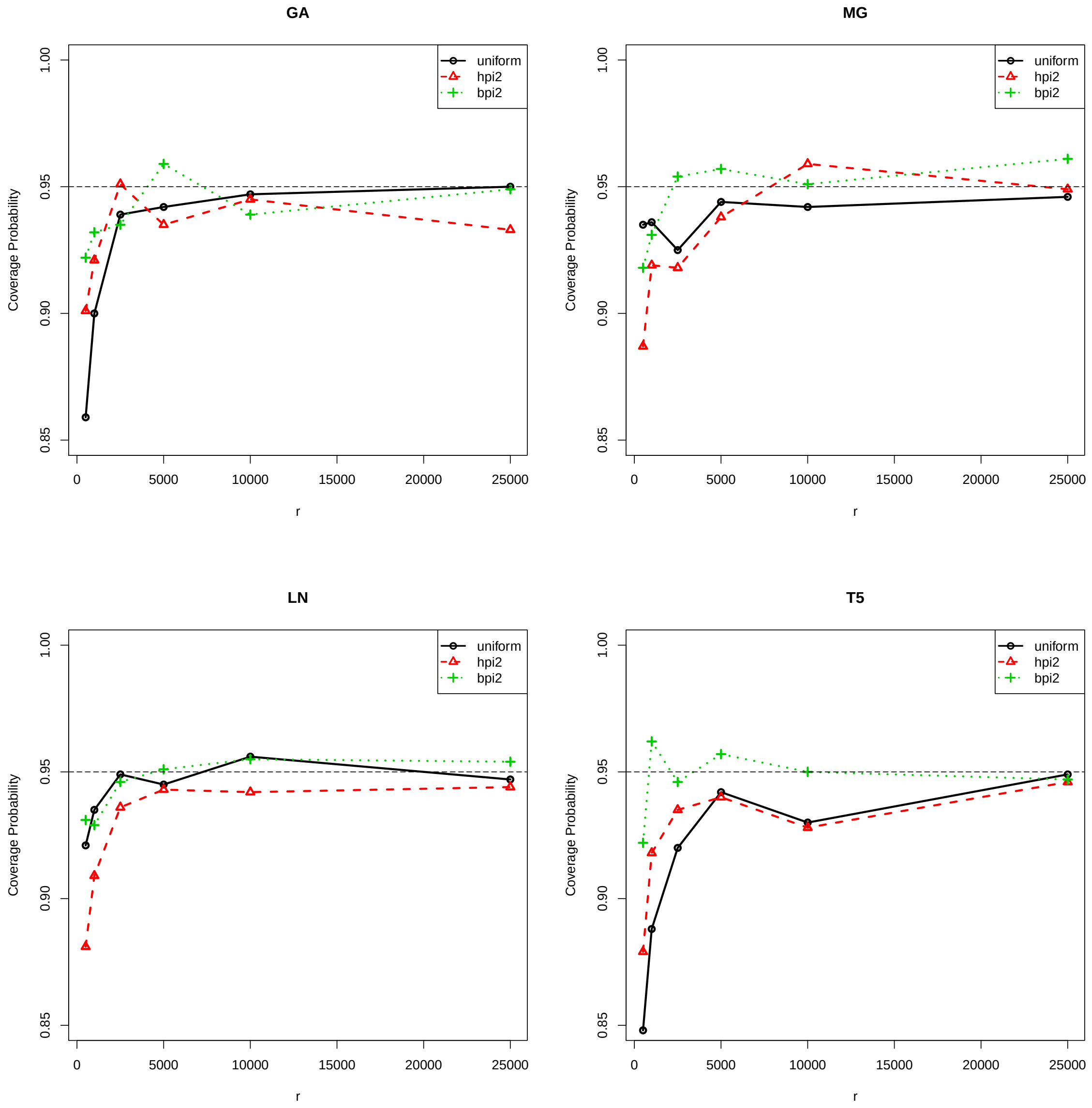


Table 2: The averages of the iterations in Newton's algorithm with Y_i generated from the Poisson model and $\mathbf{x}_i$ from the GA for $r_0 = 500$ and various r. The average of the iterations for the full data is 8.4.

	$\hat{\boldsymbol{\pi}}_2$		$\bar{\boldsymbol{\pi}}_2$		Uniform
r	Step1	Step2	Step1	Step2	
500	8.89	8.77	8.67	8.49	8.40
1000	8.75	8.56	8.56	8.23	8.80
1500	8.56	8.32	8.59	8.39	8.54
2000	8.55	8.01	8.58	8.53	8.34
2500	8.60	8.91	8.62	8.85	8.27

Table 3: The simulated ratios of the EMSE of the A-optimal subsampling estimator $\hat{\boldsymbol{\beta}}_r^*$ to those of the uniform subsampling estimator, calculated with Y_i generated from the Poisson and the full-sample estimator $\hat{\boldsymbol{\beta}}$ for $n = 50,000$ & $p = 50$.

r	500	1000	2500	5000	10000	25000
$r : n$	1%	2%	5%	10%	20%	50%
GA						
$\hat{\boldsymbol{\pi}}_2$	0.6533	0.5937	0.5627	0.5434	0.5343	0.5231
$\hat{\boldsymbol{\pi}}_1$	0.6554	0.6064	0.5613	0.5461	0.5322	0.5170
$\hat{\boldsymbol{\pi}}_0$	0.6494	0.6003	0.5672	0.5480	0.5346	0.5262
$\bar{\boldsymbol{\pi}}_2$	0.7715	0.7705	0.7898	0.7888	0.7972	0.7897
$\bar{\boldsymbol{\pi}}_1$	0.7665	0.7743	0.7960	0.7939	0.7975	0.8046
$\bar{\boldsymbol{\pi}}_0$	0.7592	0.7753	0.7794	0.8120	0.8020	0.7979
MG						
$\hat{\boldsymbol{\pi}}_2$	0.3678	0.3642	0.3629	0.3558	0.3467	0.3524
$\hat{\boldsymbol{\pi}}_1$	0.3502	0.3502	0.3492	0.3478	0.3504	0.3588
$\hat{\boldsymbol{\pi}}_0$	0.3529	0.3536	0.3489	0.3465	0.3565	0.3609
$\bar{\boldsymbol{\pi}}_2$	0.4230	0.4521	0.4856	0.5137	0.5268	0.5451
$\bar{\boldsymbol{\pi}}_1$	0.4186	0.4567	0.4905	0.5166	0.5251	0.5608
$\bar{\boldsymbol{\pi}}_0$	0.4098	0.4436	0.4877	0.5193	0.5393	0.5466
LN						
$\hat{\boldsymbol{\pi}}_2$	0.5328	0.5285	0.4992	0.4573	0.4823	0.4756
$\hat{\boldsymbol{\pi}}_1$	0.6002	0.5776	0.5177	0.4989	0.5560	0.5549
$\hat{\boldsymbol{\pi}}_0$	0.6267	0.5914	0.5418	0.5250	0.5200	0.5248
$\bar{\boldsymbol{\pi}}_2$	0.6602	0.6842	0.7031	0.7120	0.7010	0.7114
$\bar{\boldsymbol{\pi}}_1$	0.7049	0.7390	0.7586	0.7811	0.8152	0.8336
$\bar{\boldsymbol{\pi}}_0$	0.7348	0.7644	0.7840	0.7679	0.8163	0.7998
T5						
$\hat{\boldsymbol{\pi}}_2$	0.3587	0.3137	0.2867	0.2714	0.2760	0.2810
$\hat{\boldsymbol{\pi}}_1$	0.3469	0.2987	0.2709	0.2608	0.2678	0.2784
$\hat{\boldsymbol{\pi}}_0$	0.3318	0.2872	0.2598	0.2578	0.2657	0.2822
$\bar{\boldsymbol{\pi}}_2$	0.4013	0.3695	0.3596	0.3636	0.3861	0.4229
$\bar{\boldsymbol{\pi}}_1$	0.3807	0.3527	0.3426	0.3562	0.3812	0.4207
$\bar{\boldsymbol{\pi}}_0$	0.3622	0.3351	0.3445	0.3629	0.3867	0.4240

Table 4: The CPU times in seconds for computing the subsampling estimator $\hat{\boldsymbol{\beta}}_r^*$ with Y_i generated from the Poisson model and $\mathbf{x}_i$ from the GA model using the Scoring Algorithm 2 with pre-subsample size $r_0 = 500$, $n = 50,000$ & $p = 50$.

r	500	1000	1500	2000	2500	5000
$r : n$	1%	2%	3%	4%	5%	10%
$\hat{\boldsymbol{\pi}}_2$	4.191	4.205	4.226	4.241	4.567	4.632
$\bar{\boldsymbol{\pi}}_2$	2.313	2.334	2.356	2.395	3.025	3.564
	The CPU time for the full-data estimator $\hat{\boldsymbol{\beta}}$ is 5.872 seconds					

Table 5: The CPU times in seconds for computing $\hat{\boldsymbol{\beta}}_r^*$ with Y_i generated from the Poisson and $\mathbf{x}_i$ from the GA model using Newton's Algorithm for the full-sample sizes with pre-sample size $r_0 = 500$ & $r = 2,000$.

n	10^4	10^5	10^6	0.5×10^7
$\hat{\boldsymbol{\pi}}_2$	0.70	4.67	26.30	98.06
$\bar{\boldsymbol{\pi}}_2$	0.64	3.50	15.22	49.22
Full data	0.76	6.59	58.26	299.18

Table 6: Same as Table 3 except for truncation 10%.

r	500	1000	2500	5000	10000	25000
$r; n$	1%	2%	5%	10%	20%	50%
			GA			
$\hat{\boldsymbol{\pi}}_2$	0.6434	0.5718	0.5499	0.5325	0.5310	0.5159
$\hat{\boldsymbol{\pi}}_1$	0.6271	0.5811	0.5450	0.5389	0.5265	0.5199
$\hat{\boldsymbol{\pi}}_0$	0.6299	0.5823	0.5481	0.5410	0.5302	0.5163
$\bar{\boldsymbol{\pi}}_2$	0.7730	0.7662	0.7898	0.7965	0.7980	0.7960
$\bar{\boldsymbol{\pi}}_1$	0.7688	0.7658	0.8021	0.8032	0.7968	0.8079
$\bar{\boldsymbol{\pi}}_0$	0.7701	0.7726	0.7866	0.8121	0.8122	0.7935
			MG			
$\hat{\boldsymbol{\pi}}_2$	0.3671	0.3571	0.3534	0.3444	0.3483	0.3540
$\hat{\boldsymbol{\pi}}_1$	0.3525	0.3527	0.3404	0.3464	0.3534	0.3633
$\hat{\boldsymbol{\pi}}_0$	0.3488	0.3410	0.3441	0.3527	0.3524	0.3629
$\bar{\boldsymbol{\pi}}_2$	0.4236	0.4483	0.4925	0.5070	0.5306	0.5486
$\bar{\boldsymbol{\pi}}_1$	0.4201	0.4555	0.4938	0.5070	0.5340	0.5497
$\bar{\boldsymbol{\pi}}_0$	0.4111	0.4473	0.4915	0.5202	0.5354	0.5510
			LN			
$\hat{\boldsymbol{\pi}}_2$	0.5230	0.5275	0.4854	0.4466	0.4847	0.4764
$\hat{\boldsymbol{\pi}}_1$	0.5865	0.5411	0.5404	0.4924	0.5453	0.5439
$\hat{\boldsymbol{\pi}}_0$	0.5853	0.5853	0.5359	0.4973	0.5124	0.5395
$\bar{\boldsymbol{\pi}}_2$	0.6571	0.6894	0.6773	0.6833	0.7002	0.7404
$\bar{\boldsymbol{\pi}}_1$	0.6965	0.7325	0.7799	0.7791	0.8176	0.8318
$\bar{\boldsymbol{\pi}}_0$	0.7126	0.7565	0.8055	0.7710	0.8076	0.8029
			T5			
$\hat{\boldsymbol{\pi}}_2$	0.3538	0.3060	0.2815	0.2722	0.2753	0.2823
$\hat{\boldsymbol{\pi}}_1$	0.3394	0.2900	0.2678	0.2595	0.2650	0.2817
$\hat{\boldsymbol{\pi}}_0$	0.3233	0.2793	0.2604	0.2587	0.2659	0.2824
$\bar{\boldsymbol{\pi}}_2$	0.4081	0.3721	0.3595	0.3680	0.3872	0.4241
$\bar{\boldsymbol{\pi}}_1$	0.3844	0.3565	0.3451	0.3600	0.3812	0.4232
$\bar{\boldsymbol{\pi}}_0$	0.3613	0.3356	0.3453	0.3667	0.3885	0.4258

Table 7: Same as Table 3 except for truncation 30%.

r	500	1000	2500	5000	10000	25000
$r : n$	1%	2%	5%	10%	20%	50%
			GA			
$\hat{\boldsymbol{\pi}}_2$	0.6196	0.5769	0.5551	0.5372	0.5371	0.5317
$\hat{\boldsymbol{\pi}}_1$	0.6198	0.5752	0.5480	0.5435	0.5381	0.5373
$\hat{\boldsymbol{\pi}}_0$	0.6185	0.5723	0.5465	0.5486	0.5377	0.5345
$\bar{\boldsymbol{\pi}}_2$	0.7816	0.7805	0.8033	0.8041	0.8077	0.8126
$\bar{\boldsymbol{\pi}}_1$	0.7832	0.7811	0.8103	0.8055	0.8137	0.8168
$\bar{\boldsymbol{\pi}}_0$	0.7774	0.7823	0.7997	0.8125	0.8140	0.8043
			MG			
$\hat{\boldsymbol{\pi}}_2$	0.3667	0.3625	0.3568	0.3573	0.3536	0.3633
$\hat{\boldsymbol{\pi}}_1$	0.3515	0.3556	0.3491	0.3544	0.3660	0.3674
$\hat{\boldsymbol{\pi}}_0$	0.3502	0.3493	0.3488	0.3590	0.3515	0.3611
$\bar{\boldsymbol{\pi}}_2$	0.4309	0.4629	0.4868	0.5226	0.5351	0.5524
$\bar{\boldsymbol{\pi}}_1$	0.4250	0.4539	0.4985	0.5176	0.5354	0.5630
$\bar{\boldsymbol{\pi}}_0$	0.4102	0.4484	0.4977	0.5193	0.5393	0.5617
			LN			
$\hat{\boldsymbol{\pi}}_2$	0.5193	0.5118	0.4905	0.4791	0.4721	0.5021
$\hat{\boldsymbol{\pi}}_1$	0.5619	0.5496	0.5325	0.5132	0.5637	0.5466
$\hat{\boldsymbol{\pi}}_0$	0.5596	0.5675	0.5274	0.5120	0.5204	0.5371
$\bar{\boldsymbol{\pi}}_2$	0.6654	0.6930	0.7116	0.7232	0.7366	0.7309
$\bar{\boldsymbol{\pi}}_1$	0.6989	0.7329	0.7832	0.7546	0.8173	0.8252
$\bar{\boldsymbol{\pi}}_0$	0.7181	0.7604	0.7909	0.7819	0.8316	0.8255
			T5			
$\hat{\boldsymbol{\pi}}_2$	0.3608	0.3160	0.2893	0.2763	0.2826	0.2880
$\hat{\boldsymbol{\pi}}_1$	0.3460	0.2966	0.2785	0.2700	0.2724	0.2864
$\hat{\boldsymbol{\pi}}_0$	0.3295	0.2843	0.2629	0.2658	0.2763	0.2954
$\bar{\boldsymbol{\pi}}_2$	0.4121	0.3778	0.3704	0.3645	0.3875	0.4239
$\bar{\boldsymbol{\pi}}_1$	0.3882	0.3588	0.3493	0.3602	0.3860	0.4232
$\bar{\boldsymbol{\pi}}_0$	0.3676	0.3374	0.3424	0.3639	0.3911	0.4198

Table 8: Same as Table 3 except for using the Scoring Algorithm in Alg. 2 (instead of the full-sample $\hat{\boldsymbol{\beta}}$) with presample size $r_0 = 500$.

r	500	1000	2500	5000	10000	25000
$r : n$	1%	2%	5%	10%	20%	50%
GA						
$\hat{\pi}_2$	0.7778	0.7375	0.7749	0.8050	0.8276	0.8499
$\hat{\pi}_1$	0.7794	0.7594	0.7781	0.7898	0.8259	0.8778
$\hat{\pi}_0$	0.7792	0.7657	0.7750	0.8096	0.8413	0.8725
$\bar{\pi}_2$	0.7805	0.7879	0.8036	0.8237	0.8300	0.8205
$\bar{\pi}_1$	0.7930	0.7888	0.8188	0.8341	0.8271	0.8174
$\bar{\pi}_0$	0.7911	0.7967	0.8293	0.8419	0.8494	0.8416
MG						
$\hat{\pi}_2$	0.4192	0.4869	0.5671	0.6089	0.7003	0.7533
$\hat{\pi}_1$	0.4339	0.5021	0.5856	0.6567	0.7313	0.7869
$\hat{\pi}_0$	0.4486	0.5219	0.5941	0.6723	0.7247	0.7884
$\bar{\pi}_2$	0.4270	0.4712	0.4905	0.5279	0.5555	0.5557
$\bar{\pi}_1$	0.4195	0.4579	0.5144	0.5157	0.5618	0.5620
$\bar{\pi}_0$	0.4254	0.4603	0.4854	0.5371	0.5735	0.5805
LN						
$\hat{\pi}_2$	0.6271	0.6467	0.6639	0.6623	0.7056	0.7639
$\hat{\pi}_1$	0.6990	0.7057	0.6935	0.7226	0.8034	0.8218
$\hat{\pi}_0$	0.7114	0.7384	0.7262	0.7301	0.8335	0.8643
$\bar{\pi}_2$	0.6606	0.6884	0.7185	0.7238	0.7160	0.7500
$\bar{\pi}_1$	0.6960	0.7362	0.7549	0.7833	0.8286	0.8412
$\bar{\pi}_0$	0.7329	0.7824	0.8193	0.7992	0.8546	0.8145
T5						
$\hat{\pi}_2$	0.3184	0.3077	0.2828	0.2969	0.3139	0.3291
$\hat{\pi}_1$	0.3079	0.2933	0.2964	0.2957	0.3111	0.3295
$\hat{\pi}_0$	0.3260	0.3087	0.3022	0.3084	0.3240	0.3419
$\bar{\pi}_2$	0.3956	0.3808	0.3626	0.3719	0.3927	0.4156
$\bar{\pi}_1$	0.3744	0.3483	0.3500	0.3596	0.3853	0.4209
$\bar{\pi}_0$	0.3419	0.3425	0.3521	0.3628	0.3967	0.4285

Table 9: 23 Features In the Blog Feedback Data

Tc	Total number of comments before basetime
Cl24	Number of comments in the 24 hours right before the basetime
Ct1t2	Number of comments in the time period between $T1$ and $T2$, where $T1$ denotes the time 48 hours before basetime, $T2$ denotes the date time 24 hours before basetime,
Cf24	Number of comments in 24 hours immediately after publication of the post but before basetime
Tt	Total number of trackbacks before basetime,
Tl24	Number of trackbacks in the last 24 hours before the basettime
Tt1t2	Number of trackbacks between T1 and T2, where T1 is the time point 48 hours before basetime and T2 the time point 24 hours before basetime
Tf24	Number of trackbacks in 24 hours immediately after publication of the post but before basetime
Ltime	Length of time between the publication of the blog post and basetime
Lbp	Length of the blog post
Mbt, Tbt, Wbt THbt, Fbt, Sbt	Indicators (0 or 1) for whether Monday to whether Saturday of the basetime,
Mpb, Tpb, Wpb THpb, Fpb, Spb	Indicators (0 or 1) for whether Monday to whether Saturday of the blog publication date
Ppage	Number of parent pages.

4 The Blog Feedback Data

In this Section, we apply the Subsampling approach to analyzing the *Blog Feedback* data using the Poisson, the Quasipoisson and the Zero-Inflated regression models. The sampling distributions were calculated using the Zero-Inflated model in (3.5) with the estimates given in (4.1) and the discussion therein, the Poisson and the Quasipoisson models in (2.11).

The data is available from the UCI machine learning repository (URL: https://archive.ics.uci.edu/), and was collected and processed from raw htmls of the blog posts. The goal is to *predict the number of comments in the upcoming 24 hours relative to the base time*. The base time was chosen from the past, and the blog posts selected were published within 72 hours before the base time. The features were recorded at the base time based on the selected blog posts.

There are $52,397$ observations in the training data, and $7,624$ observations in the test data. We used the training data to build the model, and the test data to calculate the prediction errors. There are 23 features, see Table 9.

The Poisson model is not appropriate for this data because of the observed overdispersion and inflated number of zeros. The Quasipoisson model has the same parameter estimates as the Poisson model and does not accommodate zero-inflation, it is thus not a good choice either. The Zero-inflated Poisson model allows inflated zeros and is an appropriate choice.

As 64.05% of the values in the response variable are 0, we shall consider fitting the zero-inflated Poisson regression model in (3.5) for the data. The estimating equation of the model contains the parameter $0 \leq \rho \leq 1$, which accounts for the amount of positive structural zeros beyond the sampling zeros explained by the Poisson distribution f_{poi}. In the literature, ρ can be modeled as a function of the predictor variables, for example, via the logistic link. Here for simplifying the estimating process, we shall estimate ρ first. Specifically, based on the interpretation of ρ and noting that 64.05% is the proportion of zeros in the response variable while $\exp(\mu)$ is the probability of taking zero value in the Poisson distribution, we estimate ρ by

$$\hat{\rho} = 0.6405 - \exp(-\hat{\mu}), \tag{4.1}$$

where $\hat{\mu}$ is an estimator of μ. As Y follows the Zero-Inflated model (3.5), we have

$$\mathrm{P}(Y = 0) = \rho + (1 - \rho)\exp(-\mu).$$

On the other hand, $\mathrm{E}(Y) = (1 - \rho)\mu$. Thus $\mu = \mathrm{E}(Y)/(1 - \rho)$ and we get

$$p_0 = \mathrm{P}(Y = 0) = \rho + (1 - \rho)\exp(-\mathrm{E}(Y)/(1 - \rho)).$$

The empirical estimate of p_0 is

$$\hat{p}_0 = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[Y_i = 0] = \rho + (1 - \rho)\exp(-\bar{Y}/(1 - \rho)).$$

As $\bar{y} = 6.765$ and $\hat{p}_0 = 0.6405$, we get $\hat{\rho} \approx \hat{p}_0 = 0.6405$. Alternatively, we can use (4.1) to get $\hat{\rho}$ by plugging in $\hat{\mu} = 214.9628$, yielding the same value.

To compare the Poisson and the Quasipoisson models with the Zero-Inflated Poisson model, the full-sample estimates, the standard errors, the P-values for the three models are reported on Table 10. Observe that while many parameters in the Quasipoisson model were not significant, they were significant in the Zero-Inflated Poisson model.

To compare the $\hat{\boldsymbol{\pi}}_2$- subsampling with the uniform, the averages of the parameter estimates, the theoretical standard errors (Tse), the empirical standard error(Ese), and the P-values based on 1000 subsamples in the Zero-Inflated Poisson model are reported on Table 11. Observe that the averages of the $\hat{\boldsymbol{\pi}}_2$-subsampling estimates were closer to the full-sample estimates than those of the uniform subsampling estimates, and the standard errors of the former were smaller than those of the latter. In the presence of inflated zeros, the standard errors of the $\hat{\boldsymbol{\pi}}_2$-subsampling estimates were consistent with the theoretical standard errors, whereas those of the uniform subsampling estimates were not. Observe also that many P-values of the $\hat{\boldsymbol{\pi}}_2$ subsampling estimates were significant, but those of the uniform subsampling estimates were not. For example, the effects of Tc, Cl24, Cf24, Tt, Tl24, Tt1t2, Mbt, Fbt, Sbt, Mpb, Tpb, THpb, Fpb, Spb, and Ppage were detected by the $\hat{\boldsymbol{\pi}}_2$ subsampling but not by the uniform subsampling. This indicated that the optimal subsampling estimates led to more powerful tests.

Reported on Tables 12–13 are the simulated ratios of the lengths of the confidence intervals and the coverage probabilities. All the values in Table 12 were smaller than 1, indicating that the lengths of the 95% confidence intervals constructed by using $\hat{\boldsymbol{\pi}}_2$-subsampling estimates were significantly smaller than those by using the uniform subsampling estimates.

Table 14 lists the ratios of the EMSE of the $\hat{\boldsymbol{\pi}}_2$-subsampling estimates to the EMSE of the uniform subsampling estimates. The values were smaller than 0.1, indicating that the EMSE of the optimal subsampling estimates were less than 10% percent of the EMSE of the uniform subsampling estimates.

Table 15 reports the averages of the sum of squared predicted errors, and Fig. 4 graphically represents Table 15. Observes that when r was small, the uniform subsampling produced very large prediction errors. The prediction errors produced by the $\hat{\boldsymbol{\pi}}^{(2)}$ subsampling were significantly smaller than those by the uniform.

Table 10: The parameter estimates, standard errors (SE) and P-values in the Poisson, Quasipoisson and Zero-Inflated Poisson models using the full sample $n = 52,397$.

	Poisson	SE	P-value	Quasipoisson	SE	P-value	ZIPoisson	SE	P-value
Intcpt	2.70536	.01058	$< .0001$	2.70536	.08167	$< .0001$	3.42978	.01085	$< .0001$
Tc	.00371	.00004	$< .0001$	.00371	.00030	$< .0001$	.00312	.00004	$< .0001$
Cl24	.00282	.00004	$< .0001$	.00282	.00030	$< .0001$	.00276	.00004	$< .0001$
Ct1t2	.00013	.00005	.00373	.00013	.00036	.70717	.00025	.00005	$< .0001$
Cf24	-.00236	.00002	$< .0001$	-.00236	.00019	$< .0001$	-.00254	.00003	$< .0001$
Tt	.18007	.00482	$< .0001$	.18007	.03719	$< .0001$	.15279	.00471	$< .0001$
Tl24	-.09276	.00280	$< .0001$	-.09276	.02165	.00002	-.09377	.00267	$< .0001$
Tt1t2	-.03809	.00313	$< .0001$	-.03809	.02412	.11438	-.04378	.00298	$< .0001$
Tf24	-.06000	.00456	$< .0001$	-.06000	.03520	.08830	-.03660	.00445	$< .0001$
Ltime	-.06277	.00014	$< .0001$	-.06277	.00107	$< .0001$	-.05235	.00015	$< .0001$
Lbp	.00005	$< .0001$	$< .0001$	.00005	.00001	$< .0001$	.00004	.00001	$< .0001$
Mbt	.19249	.00912	$< .0001$	.19249	.07040	.00626	.09339	.00933	$< .0001$
Tbt	.07939	.01072	$< .0001$	.07939	.08276	.33744	-.06151	.01122	$< .0001$
Wbt	.02238	.01104	.04267	.02238	.08523	.79289	-.13030	.01155	$< .0001$
THbt	.05547	.01067	$< .0001$	.05547	.08238	.50077	-.09195	.01108	$< .0001$
Fbt	-.24868	.00977	$< .0001$	-.24868	.07542	.00098	-.31279	.01002	$< .0001$
Sbt	-.23916	.00794	$< .0001$	-.23916	.06128	.00010	-.22643	.00805	$< .0001$
Mpb	.18675	.00992	$< .0001$	.18675	.07658	.01474	.15946	.01051	$< .0001$
Tpb	.23210	.01107	$< .0001$	.23210	.08547	.00662	.22193	.01169	$< .0001$
Wpb	.05575	.01158	$< .0001$	.05575	.08935	.53271	.08395	.01204	$< .0001$
THpb	.36164	.01134	$< .0001$	.36164	.08755	.00004	.29686	.01174	$< .0001$
Fpb	.47488	.01037	$< .0001$	.47488	.08004	$< .0001$	.33577	.01060	$< .0001$
Spb	.19624	.00984	$< .0001$	.19624	.07599	.00982	.09328	.01011	$< .0001$
Ppage	-.17265	.00389	$< .0001$	-.17265	.03005	$< .0001$	-.11498	.00363	$< .0001$

Table 11: The averages of the parameter estimates, theoretical standard errors (Tse), empirical standard errors (Ese) and P-values based on 1000 subsamples in the Zero-Inflated Poisson model with $r_0 = 2500$ and $r = 5000$.

	Unif				$\hat{\boldsymbol{\pi}}_2$			
	Estimate	Tse	Ese	P-value	Estimate	Tse	Ese	P-value
Intercept	3.31604	.60943	.56943	$< .0001$	3.39144	.08391	.07782	$< .0001$
Tc	.00499	.00463	.00271	.28070	.00318	.00036	.00029	$< .0001$
Cl24	.00298	.00246	.00172	.22524	.00270	.00031	.00029	$< .0001$
Ct1t2	-.00006	.00275	.00212	.98193	.00024	.00037	.00034	.51017
Cf24	-.00407	.00332	.00266	.22057	-.00252	.00026	.00019	$< .0001$
Tt	.13274	.60876	.32564	.82739	.15511	.04384	.03117	.00040
Tl24	-.08212	.12111	.12792	.49776	-.09500	.01955	.02092	$< .0001$
Tt1t2	-.04429	.13443	.14660	.74182	-.04496	.02124	.02190	.03431
Tf24	-.02871	.62134	.32695	.96314	-.03759	.04335	.02892	.38585
Ltime	-.05948	.00787	.00744	$< .0001$	-.05443	.00168	.00164	$< .0001$
Lbp	.00003	.00001	.00001	.02618	.00004	.00001	.00001	$< .0001$
Mbt	.15348	.49478	.47033	.75641	.13700	.06758	.06715	.04264
Tbt	.01812	1.01089	.59248	.98570	-.07086	.09689	.10023	.46461
Wbt	-.15888	.94287	.61652	.86618	-.15383	.10749	.10257	.15239
THbt	-.11398	.85801	.59002	.89431	-.08562	.10579	.10892	.41830
Fbt	-.25691	.72860	.53337	.72438	-.25842	.09592	.11123	.00706
Sbt	-.25243	.62211	.43792	.68491	-.23805	.08079	.08066	.00321
Mpb	.18671	.73179	.49846	.79861	.22473	.07984	.10999	.00488
Tpb	.36845	.76743	.57054	.63115	.30397	.10215	.11611	.00292
Wpb	.23372	.71303	.59162	.74307	.12763	.10816	.11870	.23797
THpb	.33222	.64228	.61171	.60499	.29644	.10121	.12326	.00340
Fpb	.49398	.60383	.56036	.41331	.40595	.08629	.09056	$< .0001$
Spb	.24244	.57235	.50639	.67186	.14735	.07293	.07362	.04333
Ppage	-.13385	.10897	.12224	.21934	-.11631	.03527	.03816	.00097

Table 12: The length ratios of the 95% confidence intervals of the $\hat{\boldsymbol{\pi}}_2$-subsampling estimates to the uniform subsampling estimates in the Zero-Inflated Poisson model with the pre-subsample size $r_0 = 2500$.

r	1000	2500	5000	10000	25000	50000
Intercept	.1950	.1368	.1441	.1499	.1263	.1315
Tc	.0901	.0976	.1069	.0953	.0930	.1012
Cl24	.1452	.1532	.1792	.1583	.1571	.1568
Ct1t2	.1440	.1481	.1676	.1448	.1491	.1469
Cf24	.0599	.0633	.0720	.0767	.0736	.0837
Tt	.0848	.0761	.0915	.0863	.0792	.0810
Tl24	.1048	.1152	.1579	.1522	.1641	.1751
Tt1t2	.1055	.1129	.1522	.1498	.1555	.1645
Tf24	.0932	.0776	.0890	.0797	.0734	.0786
Ltime	.2348	.2432	.2191	.2124	.2308	.2093
Lbp	.3005	.2605	.2890	.3476	.2928	.2687
Mbt	.2015	.1284	.1478	.1625	.1280	.1324
Tbt	.1950	.1338	.1526	.1611	.1092	.1218
Wbt	.1829	.1543	.1669	.1694	.1312	.1289
THbt	.2104	.1747	.1678	.1635	.1480	.1399
Fbt	.2024	.1512	.1632	.1571	.1486	.1556
Sbt	.2066	.1591	.1737	.1632	.1438	.1602
Mpb	.1602	.1746	.1666	.1742	.1548	.1285
Tpb	.1756	.1622	.1721	.1910	.1795	.1407
Wpb	.1782	.1923	.1725	.1979	.1810	.1415
THpb	.1922	.1623	.1647	.1770	.1549	.1568
Fpb	.1597	.1447	.1662	.1782	.1426	.1423
Spb	.1664	.1336	.1333	.1430	.1242	.1210
Ppage	.2388	.2534	.3325	.2951	.2876	.3433

Table 13: The simulated percentages of the 95% confidence intervals catching the full-sample MLE in the Zero-Inflated Poisson model with the pre-subsample size $r_0 = 2500$.

r	1000	2500	5000	10000	25000	50000
Intercept	.9989	.9989	.9919	.9955	.9924	.9917
Tc	.9905	.9956	.9902	.9979	.9915	.9999
Cl24	.9998	.9916	.9947	.9941	.9995	.9938
Ct1t2	.9965	.9986	.9989	.9940	.9923	.9928
Cf24	.9981	.9973	.9977	.9977	.9971	.9958
Tt	.9959	.9942	.9933	.9991	.9941	.9922
Tl24	.9901	.9936	.9951	.9947	.9998	.9979
Tt1t2	.9916	.9998	.9950	.9916	.9928	.9961
Tf24	.9994	.9976	.9949	.9986	.9920	.9918
Ltime	.9907	.9944	.9922	.9944	.9917	.9984
Lbp	.9903	.9998	.9997	.9936	.9934	.9948
Mbt	.9940	.9903	.9971	.9932	.9908	.9948
Tbt	.9992	.9998	.9972	.9922	.9989	.9970
Wbt	.9952	.9916	.9938	.9927	.9926	.9979
THbt	.9931	.9918	.9905	.9914	.9947	.9930
Fbt	.9983	.9987	.9949	.9962	.9934	.9955
Sbt	.9990	.9978	.9932	.9949	.9914	.9995
Mpb	.9978	.9986	.9936	1.0000	.9999	.9911
Tpb	.9918	.9961	.9944	.9987	.9906	.9990
Wpb	.9950	.9900	.9919	.9922	.9974	.9951
THpb	.9917	.9958	.9945	.9945	.9963	.9984
Fpb	.9979	.9986	.9932	.9966	.9957	.9998
Spb	.9988	.9929	.9925	.9994	.9996	.9905
Ppage	.9953	.9959	.9917	.9907	.9980	.9958

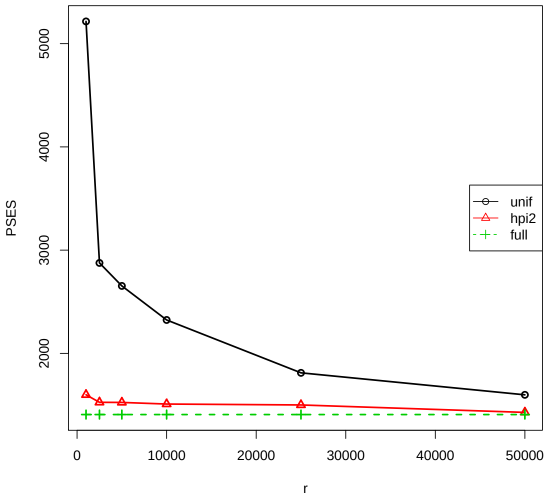
Table 14: The ratios of the EMSE of the $\hat{\boldsymbol{\pi}}_2$-subsampling estimate to that of the uniform subsampling estimate in the Zero-Inflated Poisson model with the pre-subsample size $r_0 = 2500$.

r	1000	2500	5000	10000	25000	50000
$\hat{\boldsymbol{\pi}}_2$	0.0287	0.0360	0.0373	0.0396	0.0580	0.0823

Table 15: The averages of the sum of squared predicted errors (SSPE) in the Zero-Inflated Poisson model with the pre-subsample size $r_0 = 2500$. The full-sample SSPE is $1,407.4712$.

r	1000	2500	5000	10000	25000	50000
uniform	5215.3313	2876.1691	2653.2441	2323.7320	1811.6740	1598.1760
$\hat{\boldsymbol{\pi}}_2$	1599.7506	1525.9297	1524.9536	1509.2128	1500.5681	1428.4280

Figure 4: The plots of the averages of the sum of the predicted squared errors in the Zero-Inflated Poisson model against the subsample size r with $r_0 = 2500$ for three samplings.



5 Supplementary Tables

Figure 5: The boxplots of log(probability) of the A-optimal distributions with Y_i generated from the Negative Binomial and the full-sample estimator $\hat{\boldsymbol{\beta}}$ with $n = 50,000$ & $p = 50$.

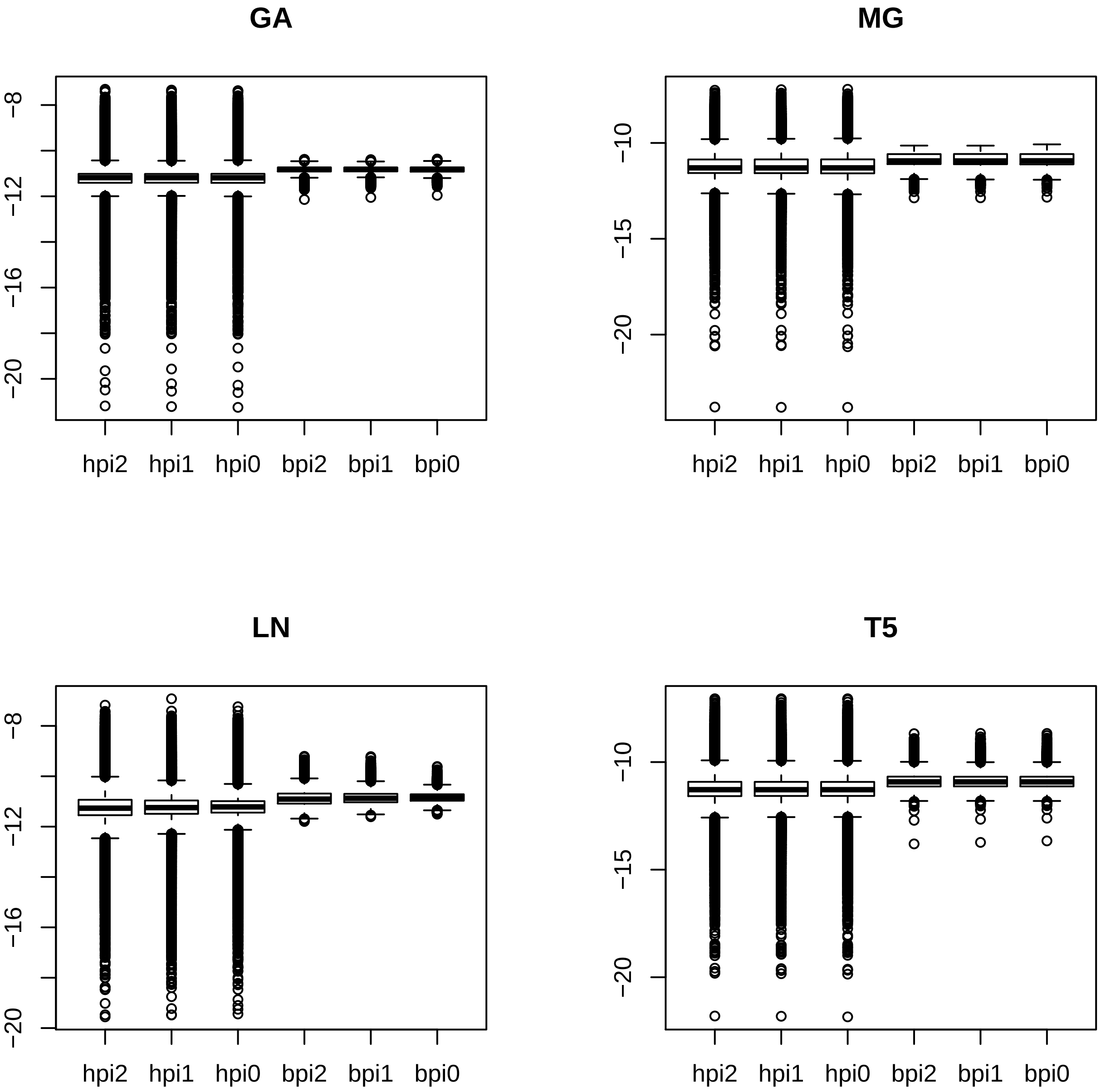


Figure 6: The plots of log (EMSE) of the subsampling estimate $\hat{\boldsymbol{\beta}}_r^*$ under different samplings against the subsample size r with Y_i generated from the Negative Binomial and the full-sample estimate $\hat{\boldsymbol{\beta}}$ with $n = 50,000$ & $p = 50$.

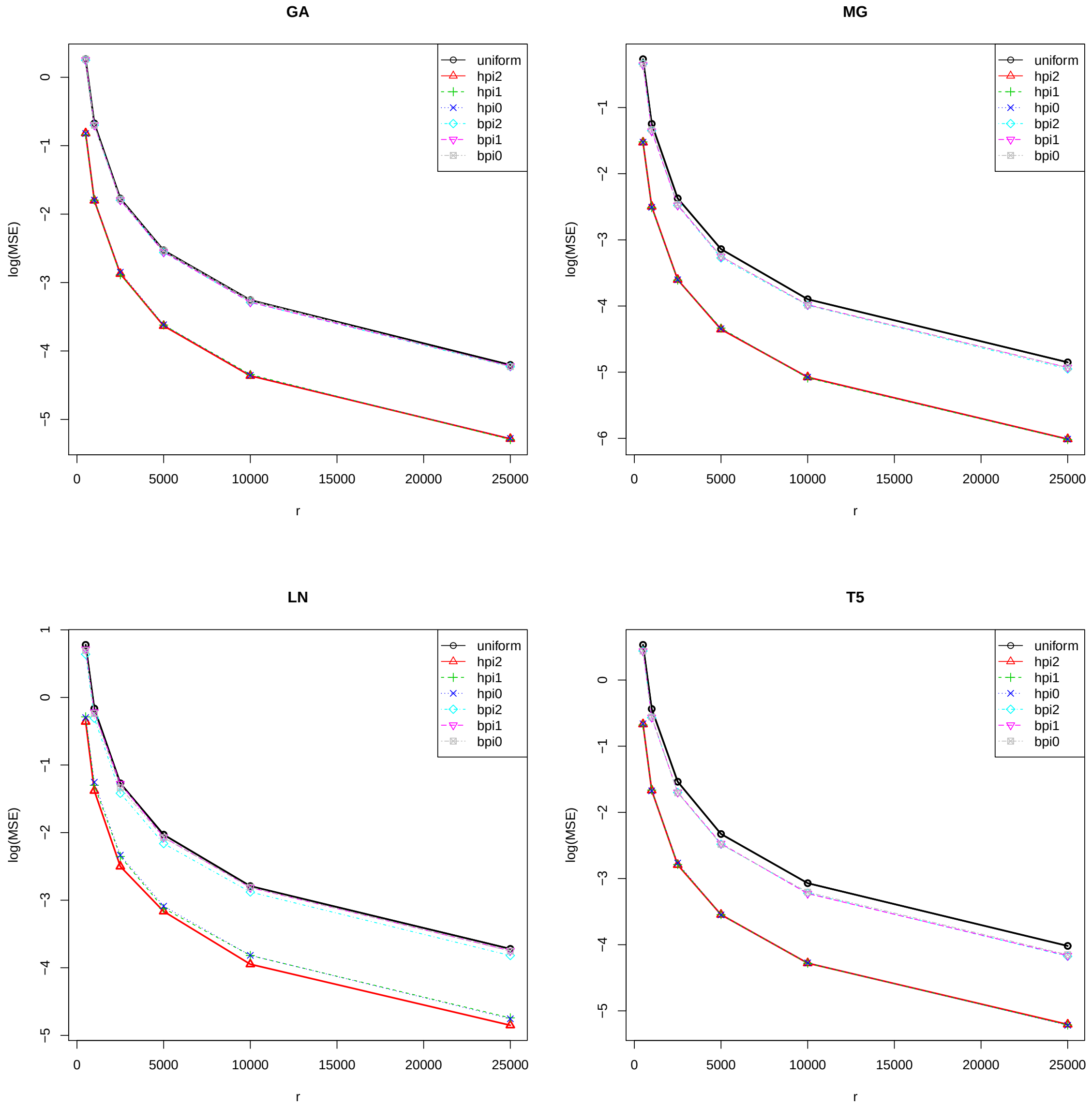


Figure 7: The plots of the simulated percentages of the 95% confidence intervals catching the true coefficient β_2 under different samplings against the subsample size r with Y_i generated from the Negative Binomial with pre-subsample size $r_0 = 500$, $n = 50,000$ & $p = 50$.

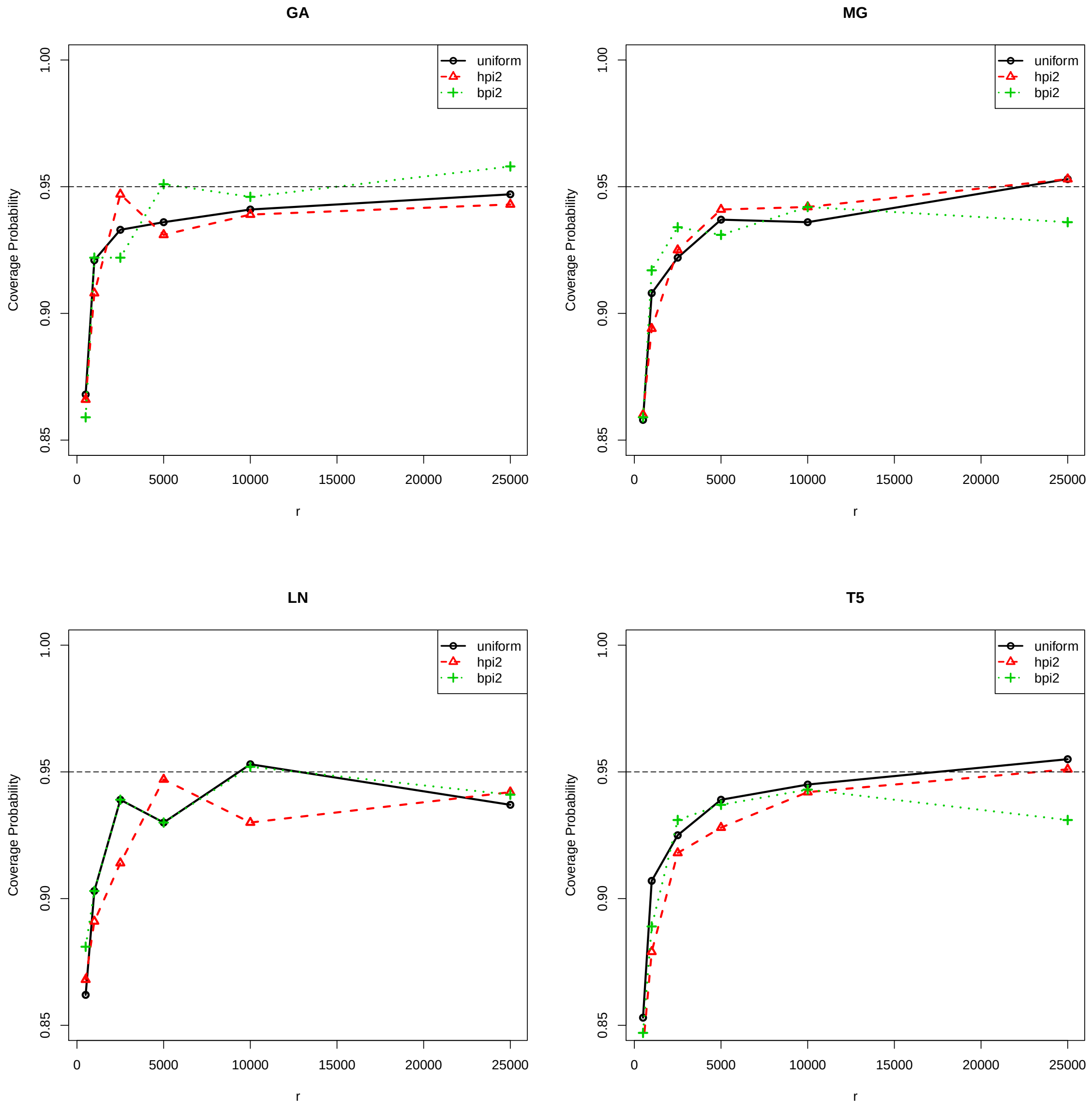


Table 16: The simulated ratios of the EMSE of the A-optimal subsampling estimator $\hat{\boldsymbol{\beta}}_r^*$ to uniform subsampling estimator, calculated with Y_i generated from the Negative Binomial model and the full-sample estimator $\hat{\boldsymbol{\beta}}$ for $n = 50,000$ & $p = 50$.

r	500	1000	2500	5000	10000	25000
$r : n$	1%	2%	5%	10%	20%	50%
GA						
$\hat{\boldsymbol{\pi}}_2$	0.3390	0.3243	0.3328	0.3319	0.3310	0.3398
$\hat{\boldsymbol{\pi}}_1$	0.3315	0.3259	0.3281	0.3359	0.3372	0.3358
$\hat{\boldsymbol{\pi}}_0$	0.3374	0.3265	0.3408	0.3374	0.3333	0.3416
$\bar{\boldsymbol{\pi}}_2$	0.9850	0.9747	0.9765	0.9699	0.9668	0.9738
$\bar{\boldsymbol{\pi}}_1$	0.9881	0.9752	0.9775	0.9747	0.9695	0.9845
$\bar{\boldsymbol{\pi}}_0$	0.9992	0.9739	0.9913	0.9955	0.9978	0.9757
MG						
$\hat{\boldsymbol{\pi}}_2$	0.2843	0.2863	0.2924	0.2974	0.3078	0.3132
$\hat{\boldsymbol{\pi}}_1$	0.2863	0.2819	0.2908	0.3030	0.3040	0.3107
$\hat{\boldsymbol{\pi}}_0$	0.2854	0.2831	0.2922	0.3004	0.3076	0.3129
$\bar{\boldsymbol{\pi}}_2$	0.9295	0.9020	0.9000	0.8748	0.9118	0.9040
$\bar{\boldsymbol{\pi}}_1$	0.9164	0.8945	0.9006	0.8936	0.9203	0.9243
$\bar{\boldsymbol{\pi}}_0$	0.9347	0.9163	0.9142	0.8970	0.9152	0.9229
LN						
$\hat{\boldsymbol{\pi}}_2$	0.3208	0.2963	0.2923	0.3214	0.3148	0.3229
$\hat{\boldsymbol{\pi}}_1$	0.3447	0.3214	0.3389	0.3364	0.3584	0.3603
$\hat{\boldsymbol{\pi}}_0$	0.3409	0.3361	0.3454	0.3474	0.3590	0.3554
$\bar{\boldsymbol{\pi}}_2$	0.8698	0.8666	0.8634	0.8762	0.9167	0.9062
$\bar{\boldsymbol{\pi}}_1$	0.9364	0.9482	0.9942	0.9643	0.9789	0.9733
$\bar{\boldsymbol{\pi}}_0$	0.9197	0.9289	0.9370	0.9564	0.9849	0.9673
T5						
$\hat{\boldsymbol{\pi}}_2$	0.3013	0.2923	0.2844	0.2955	0.2986	0.3053
$\hat{\boldsymbol{\pi}}_1$	0.2979	0.2933	0.2863	0.2956	0.2983	0.3027
$\hat{\boldsymbol{\pi}}_0$	0.3034	0.2898	0.2924	0.2944	0.2998	0.3014
$\bar{\boldsymbol{\pi}}_2$	0.9115	0.8764	0.8493	0.8565	0.8599	0.8543
$\bar{\boldsymbol{\pi}}_1$	0.9087	0.8787	0.8516	0.8658	0.8545	0.8632
$\bar{\boldsymbol{\pi}}_0$	0.9107	0.8861	0.8461	0.8546	0.8730	0.8752

Table 17: Same as Table 16 except for truncation 10%.

r	500	1000	2500	5000	10000	25000
$r : n$	1%	2%	5%	10%	20%	50%
GA						
$\hat{\boldsymbol{\pi}}_2$	0.3158	0.3146	0.3273	0.3301	0.3375	0.3390
$\hat{\boldsymbol{\pi}}_1$	0.3158	0.3184	0.3284	0.3253	0.3370	0.3366
$\hat{\boldsymbol{\pi}}_0$	0.3171	0.3162	0.3269	0.3308	0.3362	0.3391
$\bar{\boldsymbol{\pi}}_2$	0.9836	0.9777	0.9828	0.9807	0.9761	0.9685
$\bar{\boldsymbol{\pi}}_1$	0.9797	0.9771	0.9901	0.9670	0.9732	0.9783
$\bar{\boldsymbol{\pi}}_0$	0.9693	0.9804	0.9931	0.9763	0.9792	0.9720
MG						
$\hat{\boldsymbol{\pi}}_2$	0.2793	0.2801	0.2901	0.2987	0.3014	0.3108
$\hat{\boldsymbol{\pi}}_1$	0.2756	0.2722	0.2888	0.3009	0.3048	0.3099
$\hat{\boldsymbol{\pi}}_0$	0.2793	0.2762	0.2959	0.2989	0.3078	0.3116
$\bar{\boldsymbol{\pi}}_2$	0.9420	0.9153	0.9214	0.9208	0.8974	0.9136
$\bar{\boldsymbol{\pi}}_1$	0.9524	0.9160	0.9236	0.9062	0.8968	0.9199
$\bar{\boldsymbol{\pi}}_0$	0.9404	0.9213	0.9039	0.9301	0.9096	0.9147
LN						
$\hat{\boldsymbol{\pi}}_2$	0.2936	0.2887	0.2768	0.3003	0.3024	0.3175
$\hat{\boldsymbol{\pi}}_1$	0.3125	0.3169	0.3067	0.3230	0.3348	0.3719
$\hat{\boldsymbol{\pi}}_0$	0.3233	0.3062	0.3069	0.3233	0.3294	0.3652
$\bar{\boldsymbol{\pi}}_2$	0.8520	0.8418	0.8104	0.8698	0.8743	0.8878
$\bar{\boldsymbol{\pi}}_1$	0.8721	0.9179	0.8642	0.9182	0.9409	0.9457
$\bar{\boldsymbol{\pi}}_0$	0.9088	0.9586	0.8802	0.8937	0.9499	0.9804
T5						
$\hat{\boldsymbol{\pi}}_2$	0.2855	0.2843	0.2843	0.2881	0.2902	0.3015
$\hat{\boldsymbol{\pi}}_1$	0.2871	0.2817	0.2812	0.2910	0.2969	0.3014
$\hat{\boldsymbol{\pi}}_0$	0.2875	0.2819	0.2842	0.2903	0.2991	0.2960
$\bar{\boldsymbol{\pi}}_2$	0.8808	0.8615	0.8441	0.8464	0.8579	0.8484
$\bar{\boldsymbol{\pi}}_1$	0.8945	0.8723	0.8583	0.8475	0.8497	0.8476
$\bar{\boldsymbol{\pi}}_0$	0.8965	0.8792	0.8621	0.8601	0.8516	0.8470

Table 18: Same as Table 16 except for truncation 30%.

r	500	1000	2500	5000	10000	25000
$r:n$	1%	2%	5%	10%	20%	50%
			GA			
$\hat{\pi}_2$	0.3163	0.3154	0.3307	0.3333	0.3416	0.3471
$\hat{\pi}_1$	0.3091	0.3200	0.3286	0.3334	0.3417	0.3435
$\hat{\pi}_0$	0.3163	0.3199	0.3363	0.3349	0.3400	0.3476
$\bar{\pi}_2$	0.9869	0.9854	0.9928	0.9835	0.9831	0.9716
$\bar{\pi}_1$	0.9797	0.9859	0.9910	0.9656	0.9860	0.9828
$\bar{\pi}_0$	0.9671	0.9795	0.9934	0.9739	0.9882	0.9734
			MG			
$\hat{\pi}_2$	0.2735	0.2780	0.2944	0.3023	0.3077	0.3155
$\hat{\pi}_1$	0.2715	0.2762	0.2930	0.3069	0.3116	0.3143
$\hat{\pi}_0$	0.2796	0.2809	0.2962	0.3068	0.3150	0.3187
$\bar{\pi}_2$	0.9551	0.9141	0.9206	0.9370	0.9004	0.9148
$\bar{\pi}_1$	0.9483	0.9256	0.9297	0.9127	0.9088	0.9279
$\bar{\pi}_0$	0.9340	0.9295	0.9061	0.9305	0.9129	0.9191
			LN			
$\hat{\pi}_2$	0.2909	0.2874	0.2925	0.3050	0.2907	0.3126
$\hat{\pi}_1$	0.3255	0.3129	0.3126	0.3418	0.3258	0.3435
$\hat{\pi}_0$	0.3119	0.3235	0.3249	0.3390	0.3134	0.3446
$\bar{\pi}_2$	0.8524	0.8349	0.8412	0.8808	0.8313	0.8860
$\bar{\pi}_1$	0.8938	0.8668	0.9118	0.9429	0.8637	0.9391
$\bar{\pi}_0$	0.8918	0.8835	0.9237	0.9518	0.9301	0.9241
			T5			
$\hat{\pi}_2$	0.2921	0.2842	0.2888	0.2932	0.2981	0.3083
$\hat{\pi}_1$	0.2880	0.2847	0.2876	0.2957	0.3039	0.3047
$\hat{\pi}_0$	0.2867	0.2885	0.2919	0.2911	0.2998	0.3048
$\bar{\pi}_2$	0.8796	0.8935	0.8612	0.8459	0.8582	0.8555
$\bar{\pi}_1$	0.8767	0.8819	0.8668	0.8484	0.8623	0.8532
$\bar{\pi}_0$	0.8964	0.8898	0.8797	0.8615	0.8484	0.8537

Table 19: Same as Table 8 except for the Negative Binomial model.

r	500	1000	2500	5000	10000	25000
$r:n$	1%	2%	5%	10%	20%	50%
			GA			
$\hat{\pi}_2$	0.3814	0.3811	0.3822	0.3847	0.3840	0.3920
$\hat{\pi}_1$	0.3837	0.3790	0.3835	0.3858	0.3867	0.4050
$\hat{\pi}_0$	0.3788	0.3841	0.3851	0.3821	0.3868	0.3954
$\bar{\pi}_2$	1.0045	0.9018	0.9891	0.9718	0.9738	0.9896
$\bar{\pi}_1$	0.9895	0.9757	0.9905	0.9849	0.9701	0.9860
$\bar{\pi}_0$	0.9831	0.9543	0.9872	0.9744	0.9925	0.9855
			MG			
$\hat{\pi}_2$	0.3140	0.3386	0.3478	0.3578	0.3852	0.3800
$\hat{\pi}_1$	0.3232	0.3385	0.3438	0.3672	0.3859	0.3866
$\hat{\pi}_0$	0.3300	0.3405	0.3480	0.3670	0.3803	0.3801
$\bar{\pi}_2$	0.9098	0.9207	0.8999	0.9189	0.9346	0.8895
$\bar{\pi}_1$	0.9286	0.9233	0.9198	0.9253	0.9341	0.9117
$\bar{\pi}_0$	0.9521	0.9209	0.9021	0.9141	0.9454	0.9161
			LN			
$\hat{\pi}_2$	0.3759	0.3577	0.3380	0.3625	0.3892	0.3796
$\hat{\pi}_1$	0.4049	0.3750	0.3696	0.3977	0.4197	0.4378
$\hat{\pi}_0$	0.3976	0.3793	0.3573	0.3858	0.4499	0.4148
$\bar{\pi}_2$	0.8391	0.8651	0.8383	0.8846	0.9278	0.9738
$\bar{\pi}_1$	0.9403	0.9732	0.8511	0.9292	0.9367	0.9631
$\bar{\pi}_0$	0.9426	0.9851	0.9166	0.9415	0.9132	0.9970
			T5			
$\hat{\pi}_2$	0.3473	0.3404	0.3480	0.3473	0.3576	0.3747
$\hat{\pi}_1$	0.3521	0.3383	0.3498	0.3526	0.3601	0.3620
$\hat{\pi}_0$	0.3462	0.3426	0.3473	0.3573	0.3622	0.3672
$\bar{\pi}_2$	0.8952	0.8512	0.8679	0.8400	0.8387	0.8497
$\bar{\pi}_1$	0.8704	0.8551	0.8690	0.8528	0.8337	0.8557
$\bar{\pi}_0$	0.9097	0.8591	0.8697	0.8583	0.8465	0.8518

References

[1] CAMERON, C. AND TRIVEDI, P. (1998). *Regression analysis of count data*. Cambridge University Press, UK.

[2] DRINEAS P., KANNAN R. AND MAHONEY M.W. (2006a). Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM Journal on Computing*, **36**: 132-157.

[3] DRINEAS P., MAHONEY M.W. AND MUTHUKRISHNAN S. (2006b). Sampling algorithms for l2 regression and applications. *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, Pages: 1127-1136.

[4] FAN, J., HAN, F. AND LIU, H. (2014). Challenges of Big Data Analysis. *Natl Sci Rev.* **1**(2): 293-314.

[5] KLEINER, A., TALWALKAR, A., SARKAR, P. AND JORDAN, M. I. (2014). A scalable bootstrap for massive data. *J. R. Stat. Soc. Series B Stat. Methodol.* **76**(4), 795-816.

[6] SMITHSON, H., LI, F. AND PENG, H. (2024). An A-Optimal Subsampling Approach To Big Data Penalized Spline Single Index Models. *Preprint*, Available at `https://math.iupui.edu/~hanxpeng/subs-sim-0124.pdf`

[7] LIANG, F., CHENG, Y., SONG, Q., PARK, J., AND YANG, P. (2013). Stochastic approximation method for analysis of large geostatistical data. *J. Amer. Statist. Assoc.* **108**(501): 325-339.

[8] MA, P. AND SUN, X. (2014). Leveraging for big data regression. *Computational Statistics*, **7**(1): 70-76.

[9] MA, P. , MAHONEY, M.W, AND YU, B. (2015). A statistical perspective on algorithmic leveraging. *Journal of Machine Learning Research*, **16** (April): 861-911.

[10] MA, P., ZHANG, X., XING, X., MA, J., MAHONEY, W. M. (2022). Asymptotic Analysis of Sampling Estimators for Randomized Numerical Linear Algebra Algorithms. *Journal of Machine Learning Research*, **23**: 1-45.

[11] MAHONEY, M. W. (2011). Randomized algorithms for matrices and data. *arXiv:1104.5557v3* [cs.DS]

[12] MCCULLAGH, P. AND NELDER, J. (1984). *Generalized Linear Models*. Springer-Verlag, New York, NY.

[13] WANG, H.Y. (2019). More efficient estimation for logistic regression with optimal subsamples. *Journal of Machine Learning Research* **20**(132): 1-59.

[14] WANG, H.Y. AND KIM, J.K. (2022). Maximum sampled conditional likelihood for informative subsampling. *Journal of Machine Learning Research* **23**(1): 14937–14986.

[15] WANG, J., WANG, H. and XIONG, S. (2022). Unweighted estimation based on optimal sample under measurement constraints. *Canadian Journal of Statistics* **52** (2): 291-309.

[16] WANG, H., YANG, M. and STUFKEN, J. (2019). Information-Based Optimal Subdata Selection for Big Data Linear Regression. *J. Amer. Statist. Assoc.* **114** (525): 393-405.

[17] Wang, H., Zhu, R. and Ma, P. (2018). Optimal subsampling for large sample logistic regression. *J. Amer. Statist. Assoc.* **113**(522): 829-844.

[18] WANG, H.Y., ZHANG, A. AND WANG, C. (2021). Nonuniform negative sampling and log odds correction with rare events data. In Proceedings of The 35 Conference on Neural Information Processing Systems, *Proceedings of Machine Learning Research*.

[19] WANG, J., ZOU, J., AND WANG, H.Y. (2022). Sampling with replacement vs poisson sampling: A comparative study in optimal subsampling. *IEEE Transactions on Information Theory* **68**(10): 6605-6630.

[20] XU, P., YANG, J., ROOSTA-KHORASANI, F., RÉ, C. AND MAHONEY, M.W. (2016). Subsampled Newton Methods with Non-uniform Sampling. *arXiv:1607.00559.v2* [math.OC].

[21] Zhang, S., Tan, F. and Peng, H. (2023). Sample Size Determination for Multidimensional Parameters and A-Optimal Subsampling in a Big Data Linear Regression Model. To appear in the *Journal of Statistical Computation and Simulation*. *Preprint*, Available at `https://math.iupui.edu/~hanxpeng/SSD_23_4.pdf`