OPTIMAL SUBSAMPLING FOR MASSIVE PENALIZED SPINE SINGLE INDEX MODELS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Haixia Smithson

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Fang Li, Cochair

School of Science

Dr. Hanxiang Peng, Cochair

School of Science

Dr. Peijun Li

College of Science, Department of Mathematics, Purdue University

Dr. Zuofeng Shang

School of Science

Dr. Wanzhu Tu

School of Medicine, Department of Biostatistics, Indiana University

Approved by:

Dr. Plamen Stefanov

Head of the Departmental Graduate Programs

This thesis is dedicated to my family.

ACKNOWLEDGMENTS

First I would like to express my sincere gratitude to my advisors Professor Fang Li and Professor Hanxiang Peng, for their patient guidance, encouragement and advices throughout my PhD study. Professor Li is the most considerate and caring advisors I have ever met. Professor Peng's dedication to research and rigorous work styles have greatly influenced me. From them, I learned to be a better researcher and most importantly, a better person.

I would like to thank Professor Zhongmin Shen especially. He encouraged me to pursue PhD study at first place. He is the most positive and unselfish person I have ever known. He was and remains my best role model for a scientist, teacher and friend.

My sincere thank you also goes to Professor Evgeny Mukhin, who has always been caring and supportive. Thank you to Professor Joytimore Sarkar, it was a great pleasure to take his reliability class and I really enjoyed the time to do research with him.

Thank you to my thesis committee: professor Hanxiang Peng, professor Fang Li, professor Peijun Li, professor Zuofeng Shang, and professor Wanzhu Tu for their time and valuable comments on my work.

I really appreciate the chance to study and work at the Department of Mathematical Sciences of IUPUI, where I not only gained invaluable knowledge and skills, but also met lots of thoughtfully supportive faculties and staffs, and the smartest mathematician and statistician colleagues.

The most thank you to my family, especially my parents, for their unconditional love. The last special thank you goes to my lovely husband Kyle Smithson for his consistent understanding, support and always being beside me wherever I am.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

Table Page

Table Page

Table Page

LIST OF FIGURES

SYMBOLS

n	sample size or number of observations
p	number of predictor variables
r	subsample size for obtaining the subsample estimator
r_0	subsample size for calculating the sampling probability
$\mathbf{X}$	design matrix with columns $\mathbf{x}_i$, $i = 1, ..., n$ being predictor variables
$\mathbf{y}$	response variable with each observation y_i, $i = 1, ..., n$
$\boldsymbol{\beta}$	coefficient parameter of $\mathbf{X}$, called index parameter in single index models
Q	objective function, to be optimized in modeling
$\mathbf{H}$	hessian matrix
$\boldsymbol{\theta}$	unknown parameter vectors
$\hat{\boldsymbol{\theta}}$	parameter estimation on full sample
$\hat{\boldsymbol{\theta}}^*$	parameter estimation on subsample
$\boldsymbol{\pi}$	sampling probability $\{\pi_i\}, i = 1, 2, ..., n$
$\lambda_{min}\{\cdot\}$	minimum eigenvalue of matrix $\cdot$
$\lambda_{max}\{\cdot\}$	maximum eigenvalue of matrix $\cdot$
$tr\{\cdot\}$	trace of matrix $\cdot$
$\mathbf{1}[\cdot]$	indicator function: $\mathbf{1}[s] = s$ if $s > 0$, otherwise 0

ABBREVIATIONS

MSE	mean squared error
OLS	ordinary lease sqaure
SIM	single index model
AMSE	asymptotic mean squared error
tr	trace
TPF	truncated power function

ABSTRACT

Smithson, Haixia Ph.D., Purdue University, August 2019. OPTIMAL SUBSAMPLING FOR MASSIVE PENALIZED SPINE SINGLE INDEX MODELS. Major Professor: Fang Li, Hanxiang Peng.

The semiparametric single index model is well known as a compromise between parametric and nonparametric regression models, with its response mean dependent on a linear combination of covariates through an unknown univariate function. It has been widely studied due to its simplicity and flexibility, yet the challenge of its application exists especially for large datasets. This thesis focuses on the subsampling approach to fit a semiparametric single index models on large datasets, which can be computationally difficult due to the long calculating time and its high requirements on storage memory. By subsampling, the estimation on subsample, called the subsampling estimator, is used to approximate the estimation on the full sample, called the full sample estimator. To obtain an optimal sampling probability for subsampling, i.e., the optimal subsampling method, we first study the asymptotic properties of the subsampling estimator in a general semiparametric single index model with a general subsampling method, then we derive the formula of the optimal sampling probability by minimizing the asymptotic MSE of the subsampling estimator. We consider specific models in simulation studies and real data applications to investigate the numerical performance of the optimal subsampling method.

1. INTRODUCTION

1.1 Subsampling

Big data has been a successful product under rapid development of technologies in modern time. The unprecedented size and complexity of big data provide us great opportunities to explore as much information as possible, which is in demand for precise decision-making and knowledge discoveries in many industries, such as health care contributions, public sector services, industrialized and natural resources, banking sectors, etc.. The high expectation on big data analytic leads us to a reality that the "big" in either sample size or number of predictors makes the computing difficult, in terms of both the long calculating time and insufficient storage memory.

There have been many methods to deal with the time consuming calculations in big data model fitting, such as the improvement on computing facilities, for example, using supercomputers, the divide and conquer method, and the subsampling method. In consideration of user accessibility limitation and economic cost, we consider the subsampling method as our priority choice to deal with the penalized spline single index model on large datasets in this paper, see Drineas et al. (2006b) as an example of the subsampling method in practice.

As we know that the simplest thus often used sampling method is the uniform sampling method. Uniform sampling randomly assigns equal probability to each observation, hence saves us the effort to calculate the sampling probability. However, this also means that the chance of selecting each important observation is same as that of selecting each trivial observation. Hence, to keep as more information as possible in the sampling process, we expect an sampling method that extracts a small portion of data that best represents the full data. This is the initial thought of this paper's work. There have been successful researches on the optimal sampling method in regression analysis on large datasets, for example, Drineas et al. (2011) proposed to take randomized Hadamard transformation on datasets then take a subsample uniformly to approximate the OLS estimator in linear regressions. Zhu et al (2015) proposed to use normalized leverage score of the covariate matrix of predictor variables as the non-uniform subsampling probabilities in linear regression. Wang et al. (2017) developed an optimal subsampling method in logistic regression by minimizing the asymptotic mean squared error (MSE) of the subsample estimator. Peng and Tan (2018) studied the optimal subsampling method on linear regression both theoretically and computationally. These have been a great motivation and resource for this paper's work on the optimal sampling method for nonlinear regression models, from which, we consider one semiparametric single index model— penalized spline single index models. The computation cost of penalized spline single index models comes from two aspects. One is the high dimensionality of the objective function optimization caused by the extra parameters in the regression function, the other one is the grid search to find the optimal penalization to the spline coefficients. The detailed structure of penalized single index models can be seen in section 2.1 and section 4.

1.2 Single Index Models(SIM)

In a single index model, the response y_i and p covriate $\mathbf{x}_i$ satisfy

$$y_i = m(\boldsymbol{\beta}^T \mathbf{x_i}) + \epsilon_i, \quad i = 1, 2, ..., n, \tag{1.1}$$

where (i) $\boldsymbol{\beta} \in \mathbf{R}^p$ is an unknown parameter, referred as the index parameter, which is assumed to satisfy $\|\boldsymbol{\beta}\| = 1$ with its first element $\beta_1 > 0$, for identifiability;

(ii) $m : \mathbf{R} \to \mathbf{R}$ is an unknown univariate function;

(iii) $\epsilon_i' s$ are i.i.d random errors with mean 0 and constant variance $var(\epsilon_i) = \sigma_0^2$.

Let us write $(\mathbf{X}, \mathbf{y})$ for the dataset, where $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x_n})^T$ and $\mathbf{y} = (y_1, y_2, ..., y_n)^T$.

The single index model generalizes the linear regression model by introducing an unknown univariate link function m on the linear term $\boldsymbol{\beta}^T \mathbf{x_i}$ (called the index). This not only relaxes the restriction of parametric regression models' assumption on the fixed data pattern, such as linear regression and generalized linear regression etc., but also models the interactions between the covariates. On the other hand, the index reduces the multivariate predictors to the univariate term, thus avoids the "curse of dimensionality" problem in the fully nonparametric setting.

In literature, there have been many studies on the estimation methods of single index models. One is to obtain the nonparametric regression estimator of the model function m using such as kernel smoothing method(Ichimura (1993) and Hardle et al (1993)), penalized splines (Yu and Ruppert (2002))), regression splines (Antoniadis et al. (2004)) and B-splines (Antoniadis et al. (2004)), then minimize the proposed objective function to obtain the estimation of the index parameter β. Another one is to directly estimate β without estimating η, for example, the average derivative method (Stoker 1986), local linear estimation (Hristache et al. (2001)), methods involving the conditional variance of Y in Xia et al. (2002) and Xia (2006) etc.. This paper focuses on the penalized spline estimation of single index models.

Penalized splines gained popularity since 1990s as a flexible smoothing method in semi-parametric regression models. The idea of penalization was originally from O'Sullivan who in 1986 proposed the integrated squared derivative of the fitted curve as the penalty (O'Sullivan (1986)). Then in 1992, Eiler and Marx derived the difference penalty, which is purely discrete, thus much simpler as it is trivial to calculate the difference of any order (Eiler and Marx (1992)). Later in 1996, they proposed a benchmark method of curve fitting by combining regressions with the B(Basic)-spline basis and their difference penalty. Subsequently Ruppert and Carrol (1997) proposed to use the truncated power function basis as components of penalized splines with smoothness from a ridge penalty on the coefficients of parameters. Later Ruppert & Carroll (2000) and Yu & Ruppert (2002) used truncated power functions in the basis with equally spaced quantiles as knots plus a partial ridge penalty on the model function, they named their approach as P-spline, with truncated power function (TPF). Their work has greatly supported the study of penalized splines. Since then, penalized splines become more and more popular and are extended to regression models focusing on different purposes. The most recent works are the penalized spline estimation for generalized partially linear single-index models by Yu, Wu and Zhang (2017), variable selections for single index models with diverging number of index parameters by G Wang and L Wang (2015), multivariate single index models on longitudinal data by Wu and Tu (2016) etc..

In Chapter 2 we will provide methodologies and theories for the subsampling method in the penalized spline single index models. Based on Chapter 2, two specific penalized spline single index models are investigated in Chapter 3. One is the single index model with B-spline plus the integrated second order penalty (Eilers & Marx (1996)), another one is the single index model with the TPF as spline basis plus a ridge penalty (Yu & Ruppert (2002)).

2. SUBSAMPLING METHOD IN PENALIZED SPLINE SINGLE INDEX MODELS

2.1 Introduction

The penalized spline single index model is one type of semi-parametric regression models. While parametric regression model such as linear regression, generalized linear regression, and nonlinear regression models have known model functions that describe the relationship between the response variable and explanatory variables, they are not flexible enough to capture data patterns correctly. Fully nonparametric regression has its reputation of flexibility as it has no predetermined relationship between the response and the explanatory variables, but it suffers from the curse of dimensionality which requires the sample size to increase exponentially with the number of explanatory variables. These give the purpose of semiparametric model as semiparametric model function is assumed to be known but with unknown parameters, this is where "semi" comes from.

Given a dataset $(\mathbf{X}, \mathbf{y})$ described in Section 1.1, let $\mathbf{B} = \mathbf{B}(\nu)$ be a continuous spline basis function of dimension d, i.e., $\mathbf{B} \in \mathbf{R}^d$ with the index vector $\nu = \boldsymbol{\beta}^T\mathbf{x}$. Then the mean function in (1.1) for the penalized spline single index model is estimated by

$$m(\nu) = \boldsymbol{\delta}^T\mathbf{B}(\nu), \tag{2.1}$$

where $\boldsymbol{\delta} \in \mathbf{R}^d$ is an unknown vector of control points. As mentioned in section 1.2, B spline (Eilers and Marx 1996) and TPF (Ruppert et al. 2003) are the two splines advocated in literature. We will introduce the detailed structure of these two splines in section 4.

Denote the unknown parameter by $\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\delta} \end{pmatrix} \in \mathbf{R}^{p+d}$.

The estimation of $\boldsymbol{\theta}$ is obtained by minimizing the residual sum of squares plus a penalty term, i.e.

$$Q(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{\delta}^T\mathbf{B}(\boldsymbol{\beta}^T\mathbf{x}_i))^2 + \lambda P_\lambda(\boldsymbol{\theta}), \tag{2.2}$$

subject to the constraints

$$\|\boldsymbol{\beta}\| = 1 \quad \text{and} \quad \boldsymbol{\beta}_1 > 0.$$

Here λ is the penalty (or tuning) parameter, and $P_\lambda(\boldsymbol{\theta})$ is the penalization on the parameter estimation. As $\lambda \to 0$, it ends up with no penalty hence the curve fitting will be very noisy to fit closely to every current data point, but not necessarily close to the new data, which is called ”overfitting”. As $\lambda \to \infty$, the penalty term dominates, and the solution converges to the OLS line as its second derivative is always zero, this leads to ”underfitting”.

In literature, the popular penalties are the partial ridge penalty on $\boldsymbol{\delta}$ (Yu and Ruppert (2002)), the penalty on the integrated second derivative of fitted curve (Osullivan 1986&1988), the difference penalty (Eilers & Marx (1996)), the SCAD penalty for variable selection (Fan and Li (2001)) etc.. The way to choose the penalty λ is by grid searching using criterion such as minimizing cross-validation (CV) score, generalized cross-validation (GCV) score, or Akaike's information criterion (AIC) etc.. In this paper, from a grid values of λ, for example, as Yu and Ruppert (2002) suggested, from 30-points grid where $\log_{10}(\lambda)$ are equally spaced quantiles in the interval $[-6, 7]$, we choose one that minimizes the GCV score

$$GCV(\lambda) = \frac{n^{-1}\sum_{i=1}^{n}\{y_i - \boldsymbol{\delta}^T\boldsymbol{B}(\boldsymbol{\beta}^T\mathbf{x}_i)\}^2}{\{1 - n^{-1}tr\mathbf{A}(\lambda)\}^2}$$

Here $\mathbf{A}(\lambda)$ is the hat matrix of the penalized p-splined single-index model, so that the fitted values are

$$\hat{\mathbf{y}} = \mathbf{A}(\lambda)\mathbf{y}. \tag{2.3}$$

Notice that we applied the cyclic permutations of the trace function in above, which reduces the dimension of matrix thus helps relax the memory issue on storing big matrices.

For the estimation of $\boldsymbol{\beta}$, to apply the constraints on $\boldsymbol{\beta}$, we can reparametrize $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta}(\boldsymbol{\phi}) = \frac{(1, \boldsymbol{\phi}^T)^T}{\sqrt{1 + \|\boldsymbol{\phi}\|^2}} \tag{2.4}$$

with $\boldsymbol{\phi} \in \mathbf{R}^{p-1}$. Then the parameter to be estimated becomes $\boldsymbol{\theta}_{\boldsymbol{\phi}} = \begin{pmatrix} \boldsymbol{\phi} \\ \boldsymbol{\delta} \end{pmatrix} \in \boldsymbol{R}^{p+d-1}$. Thus the original parameter vector $\boldsymbol{\theta}$ loses one dimension after it is reparameterized to $\boldsymbol{\theta}_{\boldsymbol{\phi}}$. The Jacobian matrix of transforming $\boldsymbol{\theta}$ to $\boldsymbol{\theta}_{\boldsymbol{\phi}}$ is

$$\mathbf{J} = \begin{bmatrix} -\frac{\boldsymbol{\phi}^T}{(1+\|\boldsymbol{\phi}\|^2)^{3/2}} & \mathbf{0}_{1\times d} \\ \frac{\mathbf{I}_{(p-1)\times(p-1)}}{\sqrt{1+\|\boldsymbol{\phi}\|^2}} - \frac{\boldsymbol{\phi}\boldsymbol{\phi}^T}{(1+\|\boldsymbol{\phi}\|^2)^{3/2}} & \mathbf{0}_{(p-1)\times d} \\ \mathbf{0}_{d\times(p-1)} & \mathbf{I}_{d\times d} \end{bmatrix}. \tag{2.5}$$

Note that the inverse of the transformation (2.4) is

$$\boldsymbol{\phi} = \frac{(\beta_2, ..., \beta_p)^T}{\beta_1}. \tag{2.6}$$

The objective function (1.1) becomes

$$Q_n(\boldsymbol{\theta}_{\boldsymbol{\phi}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{\delta}^T \boldsymbol{B}(\boldsymbol{\beta(\phi)}^T \mathbf{x}_i))^2 + \lambda P_\lambda(\boldsymbol{\theta}_{\boldsymbol{\phi}}). \tag{2.7}$$

Denote the estimate of $\boldsymbol{\theta}_{\boldsymbol{\phi}}$ by $\hat{\boldsymbol{\theta}}_{\boldsymbol{\phi}} = \begin{pmatrix} \hat{\boldsymbol{\phi}} \\ \hat{\boldsymbol{\delta}} \end{pmatrix}$. Then

$$\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}(\hat{\boldsymbol{\phi}}).$$

The algorithm for the estimation procedure is as follows:

Algorithm 1

Step 1. Initialization. Calculate $\hat{\boldsymbol{\beta}}_0$ as the least squares estimate in the linear regression based on the dataset $(\mathbf{X}; \mathbf{y})$. By the constraint on $\boldsymbol{\beta}$, set

$$\hat{\boldsymbol{\beta}}_0 = \text{sign}(\hat{\beta}_{01})\frac{\hat{\boldsymbol{\beta}}_0}{\|\hat{\boldsymbol{\beta}}_0\|},$$

where $\hat{\beta}_{01}$ is the first element of $\hat{\beta}_0$. Then obtain $\hat{\boldsymbol{\phi}}_0$ from (2.6), and $\hat{\boldsymbol{\delta}}_0$ by minimizing the function (2.7) after plugging in $\boldsymbol{\phi}_0 = \hat{\boldsymbol{\phi}}_0$.

Step 2. Optimization. Use "optim" package in R to minimize $Q_n(\boldsymbol{\theta})$ in (2.7) with the initial value $\hat{\boldsymbol{\theta}}_{\boldsymbol{\phi}_0} = \begin{pmatrix} \hat{\boldsymbol{\phi}}_0 \\ \hat{\boldsymbol{\delta}}_0 \end{pmatrix}$ to obtain $\hat{\boldsymbol{\theta}}_{\boldsymbol{\phi}} = \begin{pmatrix} \hat{\boldsymbol{\phi}} \\ \hat{\boldsymbol{\delta}} \end{pmatrix}$

Step 3. Reparametrization. Use (2.4) to obtain $\hat{\boldsymbol{\beta}}$, hence $\hat{\boldsymbol{\theta}} = \begin{pmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{\delta}} \end{pmatrix}$.

Step 4. Cross validation. Apply above three steps to each λ in the grid, and choose one that has the minimal GCV score value, its corresponding parameter estimates will be the final estimates.

Till now, we can observe the enormous computational work in the penalized spline single index models. The optimization of (2.2) through newton or quasi-newton method takes $O(n^2(p + d))$ running time in each iteration, plus the cross validation process, the computing can be extremely challenging especially when n is huge. A subsampling method can downsize the data. Next, we are going to introduce the methodologies of the general subsampling method in penalized single index models in section 2.2, and the asymptotic theories of the resultant subsample estimator in section 2.3.

2.2 The Methodology

In this section, we will apply the general subsampling method to penalized spline single index models to obtain the subsample estimator.

We shall the subsampling procedure described in Peng and Tan (2018), Zhao (2018). First, take a small sample of size $r(r << n)$ from a given full sample $(\mathbf{X}; \mathbf{y}) = \{(\mathbf{x}_1; y_1), (\mathbf{x}_2; y_2), ..., (\mathbf{x}_n; y_n)\}$ with replacement and using the sampling probability distribution $\boldsymbol{\pi} = (\pi_1, \pi_2, ..., \pi_n)$ (assumed known for now). The obtained subsample is denoted as $(\mathbf{X}^*; \mathbf{y}^*) = \{(\mathbf{x}_1^*; y_1^*), (\mathbf{x}_2^*; y_2^*), ..., (\mathbf{x}_r^*; y_r^*)\}$ with the corresponding sampling probability vector $\boldsymbol{\pi}^* = (\pi_1^*, \pi_2^*, ..., \pi_r^*)$. Then, based on (2.7), the subsampling estimator, which is denoted by $\hat{\boldsymbol{\theta}}_{\boldsymbol{\phi}}^* = \begin{pmatrix} \hat{\boldsymbol{\phi}}^* \\ \hat{\boldsymbol{\delta}}^* \end{pmatrix}$, is obtained by minimizing

$$Q^*(\boldsymbol{\theta_\phi}) =: \frac{1}{nr}\sum_{i=1}^{r}\frac{(y_i^* - \boldsymbol{\delta}^T\boldsymbol{B}(\boldsymbol{\beta}^T\mathbf{x}_i^*))^2}{\pi_i^*} + \lambda P_\lambda(\boldsymbol{\theta_\phi}). \tag{2.8}$$

By (2.4), we obtain

$$\hat{\boldsymbol{\theta}}^* = \begin{pmatrix} \boldsymbol{\beta}(\hat{\boldsymbol{\phi}}^*) \\ \hat{\boldsymbol{\delta}}^* \end{pmatrix}.$$

We can see that (2.8) is actually the objective function (2.7) based on the weighted subsample $(\frac{\mathbf{X}^*}{\sqrt{r\boldsymbol{\pi}^*}}; \frac{\mathbf{y}^*}{\sqrt{r\boldsymbol{\pi}^*}})$. If we denote the weight as $\mathbf{w} = (w_1, w_2, ..., w_n)^T$, where $w_i = \frac{k_i}{r\pi_i}$, $k_i = 0$(i-th observation is not selected) or $k_i = 1$(i-th observation is selected), then $\mathbf{w}$ has the scaled multinomial distribution, i.e.

$$P(w_1 = \frac{k_1}{r\pi_1},\ ...,\ w_n = \frac{k_n}{r\pi_n}) = \frac{r!}{\Pi_{i=1}^n k!}\Pi_{i=1}^n \pi_i^{k_i}, \quad \sum_{i=1}^{n} k_i = r. \tag{2.9}$$

For more details, see Peng and Tan (2018).

It is clear that $Q^*(\boldsymbol{\theta_\phi})$ is continuous on the parameter space and is the Hansen-Hurwitz estimator of $Q(\boldsymbol{\theta_\phi})$, that is, $Q^*(\boldsymbol{\theta_\phi})$ is an unbiased estimate of $Q(\boldsymbol{\theta_\phi})$,

$$E^*Q^*(\boldsymbol{\theta_\phi}) = Q_n(\boldsymbol{\theta_\phi}).$$

where E^* denotes the expectation calculated under the subsampling distribution $\boldsymbol{\pi}$.

Below is an algorithm for calculating the subsampling estimator $\hat{\boldsymbol{\theta}}^*$ under a sampling probability distribution $\boldsymbol{\pi}$.

Algorithm 2

Step 1. Take a subsample $(\mathbf{X}^*; \mathbf{y}^*) = \{(\mathbf{x}_1^*; y_1^*), (\mathbf{x}_2^*; y_2^*), ..., (\mathbf{x}_r^*; y_r^*)\}$ of size $r << n$ from the full sample $(\mathbf{X}; \mathbf{y}) = \{(\mathbf{x}_1; y_1), (\mathbf{x}_2; y_2), ..., (\mathbf{x}_n; y_n)\}$ with the corresponding sampling probabilities $\boldsymbol{\pi}^* = (\pi_1^*, \pi_2^*, ..., \pi_r^*)$;

Step 2. Apply Algorithm 1 on the weighted subsample

$$(\frac{\mathbf{X}^*}{\sqrt{r\pi^*}}; \frac{\mathbf{y}^*}{\sqrt{r\pi^*}}) = \{(\frac{\mathbf{x}_1^*}{\sqrt{r\pi_1^*}}; \frac{y_1^*}{\sqrt{r\pi_1^*}}), (\frac{\mathbf{x}_2^*}{\sqrt{r\pi_2^*}}; \frac{y_2^*}{\sqrt{r\pi_2^*}}), ..., (\frac{\mathbf{x}_r^*}{\sqrt{r\pi_r^*}}; \frac{y_r^*}{\sqrt{r\pi_r^*}})\}$$

to obtain the subsampling estimator.

2.3 Asymptotic Theories

2.3.1 Basic definitions from probability theory

This section recalls definitions and important results in asymptotic theory in probability. Note that in this section, the notations are general notations.

Let $(\Omega, \mathcal{F}, \mathcal{P})$ be a probability space, where Ω denotes a sample space, $\mathcal{F}$ is a suitable $\sigma-$algebra on Ω, and $\mathcal{P}$ is a probability measure. A random variable Z is a measurable function $Z : \Omega \mapsto R$. Z could also be a vector valued random variable, i.e., $\mathbf{Z} : \Omega \mapsto R^k$. The expectation (mean) of a random variable Z is

$$E[Z] = \int_{\Omega} Z(\omega) dP(\omega)$$

The variance is

$$Var(Z) = E[(Z - E(Z))^2]$$

Given random vector variables $\mathbf{X}$, $\mathbf{W}$, the variance and variance-covariance matrix are defined by

$$Var[\mathbf{Z}] = E[(\mathbf{Z} - E[\mathbf{Z}])(\mathbf{Z} - E[\mathbf{Z}])^T]$$

$$Cov[\mathbf{Z}, \mathbf{W}] = E[(\mathbf{Z} - E[\mathbf{Z}])(\mathbf{W} - E[\mathbf{W}])^T]$$

The distribution function of X is the function $F_{\mathbf{Z}} : \mathcal{R} \mapsto [0, 1] : F_Z(z) := P(Z \leq z)$ For a random k dimensional vector variable $\mathbf{X}$ we similarly define $F_{\mathbf{Z}} : \mathcal{R}^k \mapsto [0, 1]$ by

$$F_{\mathbf{Z}}(\mathbf{z}) := P(\mathbf{Z} \leq \mathbf{z}).$$

where $\mathbf{Z} \leq \mathbf{z}$ is understood component-wise.

Next are the definitions of convergence and its notations.

Definition 2.3.1 *Let $\{Z_n\}$ be a sequence of random variables defined on $(\Omega, \mathcal{F}, \mathcal{P})$. Z_n convergences in probability to zero, written as $Z_n = o_P(1)$ or $Z_n \overset{P}{\to} 0$, if for arbitrary $\epsilon > 0$,*

$$P(|Z_n| > \epsilon) \to 0, \quad as\ n \to \infty.$$

Or more specifically, $Z_n = o_P(1)$ *if for arbitrary* $\epsilon, \epsilon_1 > 0$*, there exists* $N > 0$*, such as if* $n > N$*,*

$$P(|Z_n| > \epsilon) < \epsilon_1.$$

Definition 2.3.2 *Let* $\{Z_n\}$ *be a sequence of random variables defined on* $(\Omega, \mathcal{F}, \mathcal{P})$*.* Z_n *is bounded in probability, if for any* $\epsilon > 0$*, there exists* $M_\epsilon > 0$ *such that*

$$P(|Z_n| > M_\epsilon) < \epsilon, \quad for\ all\ n.$$

An alternative way to understand this boundedness in probability is that:

Z_n *is bounded in probability if for every* $\epsilon > 0$*, there is a set* $F \in \mathcal{F}$ *and a number* M_ϵ *such that for all* $\omega \in F$*,*

$$|Z_n(\omega)| \le M_\epsilon, \quad for\ all\ n,$$

and

$$P(F^c) < \epsilon.$$

Definition 2.3.3 *Let* $\{Z_n\}$ *be a sequence of random variables defined on* $(\Omega, \mathcal{F}, \mathcal{P})$ *and let* $\{c_n\}$ *be a sequence of strictly positive real numbers.*

1. Z_n *converges in probability to* Z *if and only if* $Z_n - Z = o_P(1)$*.*
2. $Z_n = o_P(c_n)$ *if and only if* $c_n^{-1} Z_n = o_P(1)$*.*
3. $Z_n = O_P(c_n)$ *if and only if* $c_n^{-1} Z_n = O_P(1)$*.*

The following propositions shows properties of o_P and O_P (the little o and big O notation).

Proposition 2.3.4 *Let* $\{Z_n\}$ *and* $\{W_n\}$ *be two sequences of random variables on* $(\Omega, \mathcal{F}, P)$ *and* $\{s_n\}$ *and* $\{t_n\}$ *are two sequences of strictly positive numbers.*

1. *Suppose* $Z_n = o_P(s_n)$ *and* $W_n = o_P(t_n)$*, then*
   (a) $Z_n W_n = o_P(s_n t_n)$*.*
   (b) $Z_n + W_n = o_P(\max(s_n, t_n))$*.*
   (c) $|Z_n|^r = o_P(s_n^r)$ *for* $r > 0$*.*

2. If $Z_n = o_P(s_n)$ and $W_n = O_p(t_n)$, then $Z_nW_n = o_P(s_nt_n)$.

3. If $Z_n \overset{P}{\to} Z$ and $g : \mathcal{R} \mapsto \mathcal{R}$ is a continuous mapping, then $g(Z_n) \overset{P}{\to} g(Z)$.

4. If $Z_n - W_n \overset{P}{\to} 0$ and $W_n \overset{P}{\to} W$, where W is a random variable on $(\Omega, \mathcal{F}, P)$, then $Z_n \overset{P}{\to} W$ also.

The above delevelopments can be extended to the random vector variables and matrix variables element wise, which we will not repeat again. For the k-dimensional vector $\mathbf{Z}$, denote its j-th component by Z^j.

Definition 2.3.5 *Let $\{\mathbf{Z}_n\}$ be a sequence of k dimensional random vectors on $(\Omega, \mathcal{F}, \mathcal{P})$, let $\{c_n\}$ be a sequence of strictly positive numbers. Assume k is fixed.*

1. $\mathbf{Z}_n = o_P(c_n)$ if and only if its j-th component $Z_{n,j} = o_P(c_n)$ for $j = 1, ..., k$.

2. $\mathbf{Z}_n = O_P(c_n)$ if and only if $Z_{n,j} = O_P(c_n)$ for $j = 1, ..., k$.

3. $\mathbf{Z}_n$ convergences in probability to $\mathbf{Z}$, written as $\mathbf{Z}_n \overset{P}{\to} \mathbf{Z}$ if and only if $\mathbf{Z}_n - \mathbf{Z} = o_P(1)$.

We introduce some of the notation we use throughout. We write $\|\boldsymbol{A}\|$ for the euclidean norm and $|\boldsymbol{A}|_o$ for the operator (or spectral) norm of a matrix $\boldsymbol{A}$ which are defined by

$$\|\boldsymbol{A}\|^2 = \text{trace}(\boldsymbol{A}^T\boldsymbol{A}) = \sum_{i,j} A_{ij}^2 \quad |\boldsymbol{A}|_o = \sup_{\|\boldsymbol{u}\|=1} |\boldsymbol{A}\boldsymbol{u}| = \sup_{\|\boldsymbol{u}\|=1} (\boldsymbol{u}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{u})^{1/2}.$$

In other words, the squared euclidean norm $\|\boldsymbol{A}\|^2$ equals the sum of the eigen values of $\boldsymbol{A}^T\boldsymbol{A}$, while the squared operator norm $|\boldsymbol{A}|_o^2$ equals the largest eigen value of $\boldsymbol{A}^T\boldsymbol{A}$. Consequently, the inequality $|\boldsymbol{A}|_o \leq \|\boldsymbol{A}\|$ holds. We should point out that

$$|\boldsymbol{A}|_o = \sup_{\|\boldsymbol{u}\|=1} \sup_{\|\boldsymbol{v}\|=1} \boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{v}$$

and that this simplifies to

$$|\boldsymbol{A}|_o = \sup_{\|\boldsymbol{u}\|=1} \boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{u}$$

if $\boldsymbol{A}$ is a nonnegative definite symmetric matrix.

Suppose $k = k_n$ is a sequence of positive integers possibly diverging to infinity. We have the following

Proposition 2.3.6 *Let $\{\mathbf{Z}_n\}$ be a sequence of $k = k_n$ dimensional random vectors on $(\Omega, \mathcal{F}, \mathcal{P})$, and $\mathbf{Z}$ a $k = k_n$ dimensional random vector on the same probability space. Then, $\mathbf{Z}_n - \mathbf{Z} = o_P(1)$ if and only if $\|\mathbf{Z}_n - \mathbf{Z}\| = o_P(1)$.*

For above convergence in probability, the sequence of random variables are defined on the same probability space. For convergence in distribution, this is not necessary. Below is the definition of convergence in distribution, which can also be extended to vector variables.

Definition 2.3.7 *Let $\{Z_n\}$ be a sequence of random variables, with distribution functions $\{F_{Z_n}\}$. Z_n convergences in distribution to Z if*

$$\lim_{n\to\infty} F_{Z_n}(z) = F_Z(z)$$

for all $z \in C(F_Z)$, where $C(F_Z)$ denotes the set of continuity points of F_Z. Write $Z_n \overset{D}{\to} Z$ for the convergence in distribution.

Proposition 2.3.8 *Suppose $Z_n \overset{D}{\to} Z$, then $Z_n = O_P(1)$. If $Z_n = o_P(1)$, $Z_n = O_P(1)$ also holds.*

Note that in the following section on asymptotic theory, we use P to denote the probability measure, where we use o_P and O_P notation, and use P^* to denote the probability measure for the subsampling distribution, where we use o_{P^*} and O_{P^*} notation.

2.3.2 Dimension asymptotics

Now we proceed to the asymptotic theory of the subsample estimator under the general subsampling method. The theory includes both cases where the number of covariates p fixed and where p increases with growing sample size n, which is a popular approach in the high dimensionality problem.

For the convenience of formulation in the main theorem coming next, we would like to introduce a few notations.

In what follows, unless otherwise specified, we write $\boldsymbol{\theta}_{\boldsymbol{\phi}} = \boldsymbol{\theta}$. Let

$$e_i(\boldsymbol{\theta}) = y_i - \boldsymbol{\delta}^T B(\mathbf{x}_i^T \boldsymbol{\beta}(\boldsymbol{\phi})), \quad i = 1, 2, ..., n.$$

Then $e_i(\hat{\boldsymbol{\theta}})$, $i = 1, 2, ..., n$ are the residuals of the SIM fitting on the full sample. Let $f(\boldsymbol{\theta}) = \boldsymbol{\delta}^T B(\mathbf{X}^T \boldsymbol{\beta}(\boldsymbol{\phi}))$, so $f_i(\boldsymbol{\theta}) = \boldsymbol{\delta}^T B(\mathbf{x}_i^T \boldsymbol{\beta}(\boldsymbol{\phi}))$ is the i-th component of $f(\boldsymbol{\theta})$ for $i = 1, 2, ..., n$. Introduce

$$\boldsymbol{\Psi}_n(\boldsymbol{\theta}) := n\dot{Q}(\boldsymbol{\theta}), \quad \boldsymbol{\Psi}_r^*(\boldsymbol{\theta}) := n\dot{Q}^*(\boldsymbol{\theta}).$$

Using these, one calculates

$$\boldsymbol{\Psi}_n(\boldsymbol{\theta}) = \sum_{i=1}^n \boldsymbol{\Phi}_i(\boldsymbol{\theta}), \quad \boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbf{R}^{p+d-1},$$

where

$$\boldsymbol{\Phi}_i(\boldsymbol{\theta}) = -2e_i(\boldsymbol{\theta})\dot{f}_i(\boldsymbol{\theta}) + n\lambda \dot{P}_\lambda(\boldsymbol{\theta}), \quad i = 1, 2, ..., n. \tag{2.10}$$

The hessian matrix is then given by

$$\mathbf{H}_n(\boldsymbol{\theta}) := \frac{\partial^2 nQ_n(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} = \sum_{i=1}^n \dot{\boldsymbol{\Phi}}_i(\boldsymbol{\theta}). \tag{2.11}$$

Note

$$\boldsymbol{\Psi}_n(\hat{\boldsymbol{\theta}}) = \sum_{i=1}^n \boldsymbol{\Phi}_i(\hat{\boldsymbol{\theta}}) = 0. \tag{2.12}$$

This and (2.10) imply

$$\sum_{i=1}^n 2e_i(\hat{\boldsymbol{\theta}})\dot{f}_i(\hat{\boldsymbol{\theta}}) = n^2 \lambda \dot{P}_\lambda(\hat{\boldsymbol{\theta}}), \tag{2.13}$$

For $\boldsymbol{\theta} \in \Theta$, let

$$\mathbf{M}_n(\boldsymbol{\theta}) = \sum_{i=1}^n \frac{\boldsymbol{\Phi}_i(\boldsymbol{\theta})\boldsymbol{\Phi}_i^T(\boldsymbol{\theta})}{\pi_i}, \quad \hat{\mathbf{M}}_n = \mathbf{M}_n(\hat{\boldsymbol{\theta}}).$$

Thus, for future use, by (2.10) we derive

$$\begin{aligned} \hat{\mathbf{M}}_n &= \sum_{i=1}^n \frac{-2e_i(\hat{\boldsymbol{\theta}})\dot{f}_i(\hat{\boldsymbol{\theta}})\boldsymbol{\Phi}_i^T(\hat{\boldsymbol{\theta}})}{\pi_i} + n\lambda \dot{P}_\lambda(\hat{\boldsymbol{\theta}}) \sum_{i=1}^n \boldsymbol{\Phi}_i^T(\hat{\boldsymbol{\theta}}) \\ &= \sum_{i=1}^n \frac{4e_i^2(\hat{\boldsymbol{\theta}})\dot{f}_i(\hat{\boldsymbol{\theta}})\dot{f}_i^T(\hat{\boldsymbol{\theta}})}{\pi_i} - n^2\lambda^2 \dot{P}_\lambda(\hat{\boldsymbol{\theta}})\dot{P}_\lambda^T(\hat{\boldsymbol{\theta}}), \end{aligned} \tag{2.14}$$

In Section 3, we shall use (2.14) to derive the A-optimal sampling probability distribution $\boldsymbol{\pi}$.

Let $\lambda_{\min}\{\boldsymbol{M}\}$ ($\lambda_{\max}\{\boldsymbol{M}\}$) be the minimum (maximum) eigenvalue of matrix M. For $\boldsymbol{\theta} \in \Theta$, let

$$\tilde{\mathbf{M}}_n(\boldsymbol{\theta}) = \sum_{i=1}^{n} \boldsymbol{\Phi}_i(\boldsymbol{\theta})\boldsymbol{\Phi}_i^T(\boldsymbol{\theta}), \quad \tilde{\mathbf{M}}_n = \tilde{\mathbf{M}}_n(\hat{\boldsymbol{\theta}}).$$

Let

$$a_n = \lambda_{\max}^{1/2}(\hat{\boldsymbol{M}}_n), \quad \tilde{a}_n = \lambda_{\max}^{1/2}(\tilde{\boldsymbol{M}}_n).$$

Let σ_n^2 be a sequence of positive numbers. Typically, $\sigma_n^2 = 1/\max_i \pi_i$. We need the following conditions for Theorem 2.3.9.

A1 There exists a constant b_0 such that it holds in probability that

$$a_n \to \infty, \quad \tilde{a}_n \to \infty, \quad \tilde{a}_n^{-2}\lambda_{\min}(\mathbf{H}_n(\hat{\boldsymbol{\theta}})) \geq b_0 > 0.$$

A2

$$\sum_{i=1}^{n} \pi_i^{-1}\|\dot{\Phi}_i(\hat{\boldsymbol{\theta}})\|^2 = o_P(p^{-1}r\tilde{a}_n^4).$$

A3 There is a neighborhood N_0 of $\boldsymbol{\theta}_0$ and Lipschitz constant η_i such that

$$|\dot{\Phi}_i(\boldsymbol{\theta}) - \dot{\Phi}_i(\boldsymbol{\theta}_0)|_o \leq \eta_i\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|, \quad \boldsymbol{\theta} \in N_0, \quad i = 1, \dots, n,$$

where η_i's satisfy

$$\sum_{i=1}^{n} \pi_i^{-1}\eta_i^2 = o_P(p^{-2}\sigma_n^{-2}r^2\tilde{a}_n^6).$$

A4

$$\lambda_{\max}(\tilde{\boldsymbol{M}}_n)/\lambda_{\min}(\tilde{\boldsymbol{M}}_n) = O_P(1).$$

A5 For arbitrary $\mathbf{u}$ with $\|\mathbf{u}\| = 1$, the double array $\mathbf{z}_{nj}^* = s_n^{-1}\boldsymbol{u}^T\mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}})\Phi_i(\hat{\boldsymbol{\theta}})/\pi_j^*$ $j = 1, \dots, r$, $r \geq 1$ satisfies the Lindeberg condition: for any $\epsilon > 0$, as $r \to \infty$,

$$\sum_{i=1}^{n} \pi_i\|\mathbf{z}_{ni}\|^2\mathbf{1}[\|\mathbf{z}_{ni}\| \geq \sqrt{r}\epsilon] = o_P(1),$$

where $s_n^2 = \mathbf{u}^T\mathbf{H}_n^{-1}\mathbf{M}_n\mathbf{H}_n^{-T}|_{\hat{\boldsymbol{\theta}}}\mathbf{u}$.

Theorem 2.3.9 *Assume A1-A3 hold. Suppose* $\hat{\boldsymbol{\theta}}$ *is a sequence of to (2.12) such that* $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_0 + o_P(1)$. *Assume*

$$\sum_{i=1}^{n} \pi_i^{-1} \|\boldsymbol{\Phi}_i(\hat{\boldsymbol{\theta}})\|^2 = O_P(p\sigma_n^2 \tilde{a}_n^2). \tag{2.15}$$

Then it holds in probability that there exists a sequence of the subsampling estimators $\hat{\boldsymbol{\theta}}^*$ *that minimize* (2.8) *such that if* $p^{1/2} r^{-1/2} \sigma_n \tilde{a}_n^{-1} = o_P(1)$ *then*

$$p^{-1/2} \sigma_n^{-1} \tilde{a}_n r^{1/2} (\hat{\boldsymbol{\theta}}^* - \hat{\boldsymbol{\theta}}) = O_{P^*}(1), \tag{2.16}$$

$$p^{-1/2} \sigma_n^{-1} \tilde{a}_n^{-1} \mathbf{H}_n(\hat{\boldsymbol{\theta}}) r^{1/2} (\hat{\boldsymbol{\theta}}^* - \hat{\boldsymbol{\theta}}) = -p^{-1/2} \sigma_n^{-1} \tilde{a}_n^{-1} \frac{1}{\sqrt{r}} \sum_{j=1}^{r} \frac{\Phi_j^*(\hat{\boldsymbol{\theta}})}{\pi_j^*} + o_{P^*}(1). \tag{2.17}$$

If, further, $(A4) - (A5)$ *hold, then it holds in probability that as* $r \to \infty$,

$$\mathbf{s}_n^{-1} \sqrt{r} \boldsymbol{u}^T (\hat{\boldsymbol{\theta}}^* - \hat{\boldsymbol{\theta}}) \Rightarrow N(0,1), \tag{2.18}$$

where $\mathbf{u}$ *is an arbitrary unit vector.*

Remark 2.3.10 Assumption A1 implies that $\mathbf{H}_n(\hat{\boldsymbol{\theta}})$ is positive definitely in probability uniformly in n, and satisfies

$$|\mathbf{H}_n(\hat{\boldsymbol{\theta}})|_o = O_P(1). \tag{2.19}$$

Remark 2.3.11 Assumptions A1 and A2 guarantee that the variance-covariance matrix of the subsample estimator $\hat{\boldsymbol{\theta}}^*$ is finite and positive definite. These two assumptions, together with A3, are required to prove that certain remainder terms are negligible as the subsample size r increases.

Remark 2.3.12 The commonly used cubic spline is Lipschitz continuous, and hence satisfies the Lipchitz condition in Assumption A2.

Remark 2.3.13 Consider the linear regression which has been fully studied in Peng and Tan (2018). The least square estimate is

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2$$

for the linear regression model

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon,$$

the above theorem also applies. In this case, there is no penalty, that is, $\lambda = 0$. So

$$\Phi_i(\boldsymbol{\theta}) = -2e_i(\boldsymbol{\theta})\mathbf{x}_i. \tag{2.20}$$

For simplicity, we check the case where subsampling is uniform sampling. (2.15) in Theorem 2.3.9 implies

$$\frac{1}{n}\sum_{i=1}^{n} e_i^2 \mathbf{x}_i \mathbf{x}_i^T = O_P(1),$$

which holds if $E(\mathbf{X}\mathbf{X}^T) < \infty$ and it is positive definite. The hessian matrix

$$\mathbf{H}_n = \frac{2}{n}\mathbf{X}^T\mathbf{X}, \tag{2.21}$$

A1 then implies that the minimum eigenvalue of $\frac{1}{n}\mathbf{X}^T\mathbf{X}$ is bounded away from 0, which also guarantees the positive definiteness of the hessian matrix. Since

$$\frac{\partial \mathbf{\Phi}_i(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} = 2\mathbf{x}_i\mathbf{x}_i^T$$

is irrelevant of θ, so the Lipschitz condition in A3 is trivially valid as the left side of the equation equals 0.

Remark 2.3.14 Theorem 2.3.9 shows the subsample estimator approximates the full sample estimator, and is asymptotically normal.

Remark 2.3.15 *By (2.17), we derive the main term of the variance-covariance matrix* $Var^*(\hat{\boldsymbol{\theta}}^*)$ *of the subsampling estimate* $\hat{\boldsymbol{\theta}}^*$ *as follows:*

$$\mathbf{V}_n = \mathbf{H}_n^{-1} \sum_{i=1}^{n} \frac{\Phi_i \Phi_i^T}{\pi_i} \mathbf{H}_n^{-T}\Big|_{\hat{\boldsymbol{\theta}}}$$

Proof of Theorem 2.3.9

Before we prove Theorem 2.3.9, we need Lemma 2.3.16, which is Theorem 6.3.4 of Ortega and Rheinboldt (1970). For a given set C, its closure and boundary are denoted by $\bar{C}$ and ∂C, respectively.

Lemma 2.3.16 *Let C be an open, bounded set in $\mathbf{R}^n$ and assume that $\mathbf{F} : \bar{C} \subset \mathbf{R}^n \to \mathbf{R}^n$ is continuous and satisfies $(\mathbf{x} - \mathbf{x}_0)^T \mathbf{F}(\mathbf{x}) \geq 0$ for some $\mathbf{x}_0 \in C$ and all $\mathbf{x} \in \partial C$. Then $F(\mathbf{x}) = 0$ has a solution in $\bar{C}$.*

Proof of Theorem 2.3.9: For $\mathbf{t} \in \mathbf{R}^{\mathbf{p+d-1}}$, let $\mathbf{t}_n = p^{1/2} r^{-1/2} \sigma_n \tilde{a}_n^{-1} \mathbf{t}$, and let

$$\mathbf{T}^*(\mathbf{t}) = p^{-1/2} r^{1/2} \sigma_n^{-1} \tilde{a}_n^{-1} [\boldsymbol{\Psi}^*(\hat{\boldsymbol{\theta}} + \mathbf{t}_n) - \boldsymbol{\Psi}^*(\hat{\boldsymbol{\theta}})] - \tilde{a}_n^{-2} \mathbf{H}_n(\hat{\boldsymbol{\theta}}) \mathbf{t}. \tag{2.22}$$

For an arbitrary constant $c > 0$, fix $\|\boldsymbol{t}\| \leq c$. By Assumption, $\mathbf{t}_n = o_P(1)$. Hence by the first equality in Assumption A3, for large r and with large probability,

$$\|\mathbf{T}^*(\mathbf{t})\|^2 \leq 2c^2 \tilde{a}_n^{-4} \big(\|\bar{\mathbf{H}}^*(\hat{\boldsymbol{\theta}})\|^2 + c^2 p r^{-1} \sigma_n^2 \tilde{a}_n^{-2} | \sum_{i=1}^n w_i \eta_i |^2 \big), \tag{2.23}$$

where $\bar{\mathbf{G}}^* = \sum_{i=1}^n \bar{w}_i \mathbf{G}_i$ for $\mathbf{G}^* = \sum_{i=1}^n w_i \mathbf{G}_i$ with $\bar{w}_i = w_i - E^*(w_i) = w_i - 1$. It is easy to calculate

$$rE^*(\|\bar{\mathbf{H}}^*(\hat{\boldsymbol{\theta}})\|^2) \leq \sum_{i=1}^n \pi_i^{-1} \|\dot{\Phi}_i(\hat{\boldsymbol{\theta}})\|^2 =: A_n, \tag{2.24}$$

$$rE^*(|\sum_{i=1}^n w_i \eta_i|^2) \leq rE^*(\sum_{i=1}^n w_i |\eta_i|^2) \leq \sum_{i=1}^n \pi_i^{-1} \eta_i^2 =: B_n.$$

It then follows from Assumptions A2-A3 that

$$E^*(\sup_{\|\boldsymbol{t}\| \leq c} \|\mathbf{T}^*(\mathbf{t})\|^2) \leq 2c^2 r^{-1} \tilde{a}_n^{-4} (A_n + c^2 p r^{-1} \sigma_n^2 \tilde{a}_n^{-2} B_n) = o_P(p^{-1}). \tag{2.25}$$

Note (2.22) that

$$\begin{aligned} \ell^*(c) &= \inf_{\|\mathbf{t}\| = c} \Big\{ p^{-1/2} r^{1/2} \sigma_n^{-1} \tilde{a}_n^{-1} \mathbf{t}^T \Psi^*(\hat{\boldsymbol{\theta}} + \mathbf{t}_n) \Big\} \\ &\geq c^2 \tilde{a}_n^{-2} \lambda_{\min}(\mathbf{H}_n(\hat{\boldsymbol{\theta}})) - c \sup_{\|\mathbf{t}\| = c} \|\mathbf{T}^*(\mathbf{t})\| - c p^{-1/2} r^{1/2} \sigma_n^{-1} \tilde{a}_n^{-1} \|\boldsymbol{\Psi}^*(\hat{\boldsymbol{\theta}})\|. \end{aligned} \tag{2.26}$$

By Assumption A1, $\tilde{a}_n^{-2} \lambda_{\min}(\mathbf{H}_n(\hat{\boldsymbol{\theta}})) \geq b_0 > 0$ with large probability and for large n. For $K > 0$, using $\boldsymbol{\Psi}_n(\hat{\boldsymbol{\theta}}) = 0$, we have

$$\begin{aligned} & P^*(p^{-1/2} r^{1/2} \sigma_n^{-1} \tilde{a}_n^{-1} \|\boldsymbol{\Psi}^*(\hat{\boldsymbol{\theta}})\| > K) \\ &= P^*(p^{-1/2} r^{1/2} \sigma_n^{-1} \tilde{a}_n^{-1} \|\bar{\boldsymbol{\Psi}}^*(\hat{\boldsymbol{\theta}})\| > K) \\ &\leq K^{-2} p^{-1} \sigma_n^{-2} \tilde{a}_n^{-2} \sum_i \pi_i^{-1} \|\Phi_i(\hat{\boldsymbol{\theta}})\|^2 \\ &= K^{-2} O_P(1) = o_P(1), \end{aligned}$$

where (2.15) is used for the second equality. This and (2.25)-(2.26) imply for large c,

$$\begin{aligned} P^*(\ell^*(c) > 0) &\leq 1 - P^*(\sup_{\|\mathbf{t}\|=c} \|\mathbf{T}^*(\mathbf{t})\| > b_0 c) - P^*(p^{-1/2} r^{1/2} \sigma_n^{-1} \tilde{a}_n^{-1} \|\boldsymbol{\Psi}^*(\hat{\boldsymbol{\theta}})\| > b_0 c) \\ &= 1 - o_P(1). \end{aligned}$$

By the continuity of $\boldsymbol{\Psi}^*(\boldsymbol{\theta})$ on Θ and Lemma 2.3.16, there exists some $\mathbf{t}^*$ with $\|\mathbf{t}^*\| < c$ such that

$$\boldsymbol{\Psi}^*(\hat{\boldsymbol{\theta}} + p^{1/2} r^{-1/2} \sigma_n \tilde{a}_n^{-1} \mathbf{t}^*) = 0.$$

Let $\hat{\boldsymbol{\theta}}^* = \hat{\boldsymbol{\theta}} + p^{1/2} r^{-1/2} \sigma_n \tilde{a}_n^{-1} \mathbf{t}^*$. Then $\hat{\boldsymbol{\theta}}^*$ minimizes (2.8) and satisfies

$$P^*(\|p^{-1/2} \sigma_n^{-1} \tilde{a}_n r^{1/2} (\hat{\boldsymbol{\theta}}^* - \hat{\boldsymbol{\theta}})\| \leq c) \geq 1 - o_P(1).$$

This shows (2.16). By (2.25),

$$\mathbf{T}^*(\mathbf{t}^*) = o_P(p^{-1/2}). \tag{2.27}$$

This, in view of (2.22), shows (2.17)

The asymptotic normality (2.18) follows from the established relation (2.17) and the Lindeberg-Feller theorem (e.g. Theorem 7.2.1 of Chung, 2001). More specifically, the Lindeberg condition (A5) implies that the main term on the left side of (2.17) has an asymptotic standard normal in conditional probability given the data, while the remainder term is negligible,

$$s_n^{-1} \tilde{a}_n \sigma_n \mathbf{u}^T \mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}}) p^{1/2} \boldsymbol{\alpha}_n^* = o_{P^*}(1),$$

where $p^{1/2} \boldsymbol{\alpha}_n^* = o_{P^*}(1)$ by (2.27). This follows from

$$\frac{\tilde{a}_n \sigma_n}{s_n} \leq \frac{\lambda_{\max}^{1/2}(\tilde{\mathbf{M}}_n) \sigma_n}{\lambda_{\min}^{1/2}(\tilde{\mathbf{M}}_n)) \sigma_n \|\mathbf{u}^\top \mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}})\|} \leq \frac{B}{\|\mathbf{u}^T \mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}})\|},$$

where B is a constant by Assumption A4. The proof is now complete.

3. OPTIMAL SUBSAMPLING FORMULATION AND IMPLEMENTATION

3.1 Formulation

To obtain the subsample estimator $\boldsymbol{\theta}$ as referred in Theorem 2.3.9, the sampling probability distribution π needs to be specified. As we illustrated in the introduction, a simple choice is the uniform sampling, which may not perform well due to its inefficiency. Alternatively, a non-uniform subsampling probability is derived in this section and is shown to have better performance than the uniform subsampling method in section 4 and section 5.

A-optimality aims to minimize the trace of the variance-covariance matrix of the subsampling estimator, to obtain the sampling probability. It has been used in the subsampling method for linear regression (Zhu and Ma et al. 2015), logistic regression(Wang et al. 2017). In our case, since the estimations of $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ are interconnected in the sense that if parameter $\boldsymbol{\beta}$ can be estimated accurately and efficiently, one can plug the estimate into the single index model function to obtain a good estimation for the link function, hence the estimation of parameter $\boldsymbol{\beta}$ is more important than the estimation of $\boldsymbol{\delta}$. Based on this, we only consider the subsampling method for improving the efficiency of estimating $\boldsymbol{\beta}$, which is equivalent to that of $\boldsymbol{\phi}$, we have the following result:

Theorem 3.1.1 *There exists a sampling probability $\boldsymbol{\pi} = (\pi_1, \pi_2, ..., \pi_n)$ that minimizes asympotittically the trace of variance covariance matrix of the subsampling estimator $\hat{\phi}^*$, which is given by*

$$\pi_i = \frac{\|\boldsymbol{\Lambda}_1 \mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}}) \dot{f}_i(\hat{\boldsymbol{\theta}})\| |e_i(\hat{\boldsymbol{\theta}})|}{\sum_{i=1}^{n} \|\boldsymbol{\Lambda}_1 \mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}}) \dot{f}_i(\hat{\boldsymbol{\theta}})\| |e_i(\hat{\boldsymbol{\theta}})|}, \quad i = 1, 2, ..., n, \tag{3.1}$$

where $\boldsymbol{\Lambda}_1$ is a diagonal matrix with first $p-1$ diagonal elements being 1 and the rest equals 0.

Note that $\boldsymbol{\Lambda}_1$ extracts the variance-covariance matrix of the subsampling estimator $\hat{\boldsymbol{\phi}}^*$.

Proof of Theorem 3.1.1:

Let $tr(M)$ denote the trace of a square matrix M, that is, $tr(M)$ is the sum of the diagonal entries.

By Theorem (2.3.15), we calculate that the trace as follows:

$$\begin{aligned}
& tr(Var^*(\hat{\boldsymbol{\phi}}^*)) = tr(\boldsymbol{\Lambda}_1\mathbf{V}_n\boldsymbol{\Lambda}_1) \\
= \quad & tr(\boldsymbol{\Lambda}_1\mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}})\sum_{i=1}^n \frac{\Phi_i(\hat{\boldsymbol{\theta}}))\Phi_i(\hat{\boldsymbol{\theta}}))^T}{\pi_i}\mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}})\boldsymbol{\Lambda}_1) \\
= \quad & tr(\boldsymbol{\Lambda}_1\mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}})\sum_{i=1}^n \frac{4e_i^2(\hat{\boldsymbol{\theta}})\dot{f}_i(\hat{\boldsymbol{\theta}})\dot{f}_i^T(\hat{\boldsymbol{\theta}})}{\pi_i}\mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}})\boldsymbol{\Lambda}_1) \\
& \quad -n^2 tr(\boldsymbol{\Lambda}_1\mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}})\lambda^2\dot{P}(\hat{\boldsymbol{\theta}})\dot{P}^T(\hat{\boldsymbol{\theta}})\mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}})\boldsymbol{\Lambda}_1) \\
= \quad & \sum_{i=1}^n \frac{\|2e_i(\hat{\boldsymbol{\theta}})\boldsymbol{\Lambda}_1\mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}})\dot{f}_i(\hat{\boldsymbol{\theta}})\|^2}{\pi_i} - n^2\|\boldsymbol{\Lambda}_1\mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}})\lambda\dot{P}(\hat{\boldsymbol{\theta}})\|^2.
\end{aligned}$$

where the second last equality used (2.14) and the last equality used

$$tr(\mathbf{a}\mathbf{a}^T) = \|\mathbf{a}\|^2$$

for any vector $\mathbf{a}$.

The goal is to obtain $\boldsymbol{\pi}$ by minimizing (3.2) under the constraint

$$\sum_{i=1}^n \pi_i = 1.$$

Since the second term of (3.2) is irrelevant to $\boldsymbol{\pi}$, only the first term is used to be minimized as a function of $\boldsymbol{\pi}$. By Lagrange multiplier, write the Lagrange function as

$$L(\boldsymbol{\pi};\tau) = \frac{4}{n^2}\sum_{i=1}^n \frac{\|e_i(\hat{\boldsymbol{\theta}})\boldsymbol{\Lambda}_1\mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}})\dot{f}_i(\hat{\boldsymbol{\theta}})\|^2}{\pi_i} + \tau(\sum_{i=1}^n \pi_i - 1)$$

Let

$$\frac{\partial L(\boldsymbol{\pi};\tau)}{\partial \pi_i} = -\frac{4}{n^2}\frac{\|e_i(\hat{\boldsymbol{\theta}})\boldsymbol{\Lambda}_1\mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}})\dot{f}_i(\hat{\boldsymbol{\theta}})\|^2}{\pi_i^2} + \tau = 0,$$

we obtain

$$\pi_i = \frac{2\|e_i(\hat{\boldsymbol{\theta}})\boldsymbol{\Lambda}_1\mathbf{H}_n^{-1}(\hat{\boldsymbol{\theta}})\dot{f}_i(\hat{\boldsymbol{\theta}})\|}{n\sqrt{\tau}},$$

together with $\sum_{i=1}^n \pi_i = 1$, we have (3.1).

Remark 3.1.2 As we discussed in Remark 2.3.13, the asymptotic theorems can be applied to the linear regression case $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, which was studied in Peng and Tan (2018).. When there is no penalty and no constraint on the parameters, the optimal sampling probability in (3.1) becomes

$$\pi_i^{lm} = \frac{|e_i(\hat{\boldsymbol{\theta}})|\|(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i\|}{\sum_{i=1}^n |e_i(\hat{\boldsymbol{\theta}})|\|(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i\|}, i = 1, 2, ..., n \tag{3.2}$$

we can see that the optimal sampling probability depends on the residuals and the covariates. The sampling probability is larger if the responses end up with bigger residuals, which are harder to observe as they are in the tail region of the response distribution. This means the optimal subsampling probability tends to include more off-track observations. For the covariates, the observations with larger $\|(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i\| = \mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-2}\mathbf{x}_i, i = 1, 2, .., n$ are more likely to be selected. Note that for linear regression, $\mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i, i = 1, 2, .., n$ are defined as statistical leverage scores. There has been studies for subsampling method using normalized leverage scores as sampling probability, see Ma and Sun 2014 and Ma et al. 2015.

3.2 The A-optimal Scoring Algorithm

This section solves practical issues when implementing the optimal subsampling method. We follow the A-optimal scoring algorithm proposed by Peng and Tan (2018).

Firstly, notice that computing $\boldsymbol{\pi}$ requires computing $\mathbf{H} =: \mathbf{H}_n(\hat{\boldsymbol{\theta}})$. As the Hessian matrix $\mathbf{H}(\hat{\boldsymbol{\theta}})$ is expensive to calculate and invert, a popular method to solve this issue is by using the simple diagonal approximation to the hessian (le Cun 1987). We utilize this method to approximate $\mathbf{H}$, that is, we calculate

$$\begin{pmatrix} \mathbf{H}_{(p-1)\times(p-1)} & 0 \\ 0 & \mathbf{H}_{d\times d} \end{pmatrix},$$

to replace $\mathbf{H}$, i.e. the diagonal blocks are kept, where information from other entries are skipped. Note that $\mathbf{H}_{(p-1)\times(p-1)} = \frac{\partial^2 Q}{\partial\phi\partial\phi}$. Denote the first $(p-1)$ elements of $\dot{f}_i$ in (3.1) as $\dot{f}_{i(p-1)}$. Then we can obtain an easier-to-compute version of $\boldsymbol{\pi}$ in (3.1) as

$$\tilde{\pi}_i = \frac{\|e_i(\hat{\boldsymbol{\theta}})\mathbf{H}^{-1}_{(p-1)\times(p-1)}(\hat{\boldsymbol{\theta}})\dot{f}_{i(p-1)}(\hat{\boldsymbol{\theta}})\|}{\sum_{i=1}^{n}\|e_i(\hat{\boldsymbol{\theta}})\mathbf{H}^{-1}_{(p-1)\times(p-1)}(\hat{\boldsymbol{\theta}})\dot{f}_{i(p-1)}(\hat{\boldsymbol{\theta}})\|}, \quad i = 1, 2, ..., n. \tag{3.3}$$

In Chapter 4, we report the simulation results about investigating the performance of the approximation $\tilde{\boldsymbol{\pi}}$ in (3.3) to $\boldsymbol{\pi}$ in (3.1).

Note that for calculating the sampling probability for small and moderate size of datasets or datasets with large dimension p, using $\tilde{\boldsymbol{\pi}}$ in (3.3) doesn't save much time. However, in big data analysis where it is very difficult to calculate the hessian matrix and its inverse, $\tilde{\boldsymbol{\pi}}$ is very effective in time efficiency. In Chapter 4, we will compare the performance of $\tilde{\boldsymbol{\pi}}$ and $\boldsymbol{\pi}$ in terms of efficiency of the subsampling estimation.

Secondly, we can see that the sampling probability π relys on the full sample estimator $\hat{\boldsymbol{\theta}}$, which is not practical since the full sample estimation is unknown in reality, the main goal of doing subsampling is to reduce the sample size so that the full sample estimator can be approximated by the subsample estimator instead. One way to handle this issue is to go through the two steps algorithm—Algorithm 3, this method can be referred to the A-optimal Scoring Algorithm proposed by Peng and Tan 2018.

The A-optimal Scoring Algorithm 3

Step 1. Take a subsample of size $r_0 << n$ uniformly from the full sample, then obtain the initial estimator $\hat{\boldsymbol{\theta}}_{r0}$ based on this subsample, replace $\hat{\boldsymbol{\theta}}$ with $\hat{\boldsymbol{\theta}}_{r0}$ in (3.3) to calculate $\boldsymbol{\pi}$;

Step 2. Take a subsample of size $r << n$ from the full sample using the sampling probability $\boldsymbol{\pi}$ calculated in the first step, then base on this subsample to calculate the subsample estimator $\hat{\boldsymbol{\theta}}^*$, see Algorithm 2.

Remark 3.2.1 Observe that the sampling probability (3.3) relys on derivative of the parameter $\boldsymbol{\phi}$ only, which we can relate to the case where $\boldsymbol{\delta}$ is treated as fixed. As a matter of fact, if we fix $\boldsymbol{\delta}$ so that the only unknown parameter is $\boldsymbol{\phi}$, then we use the same way, i.e. by deriving the variance covariance matrix of the subsample estimator, which is $\hat{\boldsymbol{\phi}}^*$ in this case, then minimizing the trace of variance covariance matrix, we can obtain the exactly same formula of the sampling probability as in (3.3).

Remark 3.2.2 From formula (3.1), the optimal sampling probability π_i for the i-th observation is proportional to its residual value $e_i(\hat{\boldsymbol{\theta}})$ on the full data, the hessian matrix $\mathbf{H}(\hat{\boldsymbol{\theta}})$, and the gradient vector $\dot{f}_i(\hat{\boldsymbol{\theta}})$ of the objective function at the full sample estimation $\hat{\boldsymbol{\theta}}$. The inverse of the optimal sampling probabilities for each observation are used as weights for calculating the optimal subsample estimator, because of this, the sampling probability should be bounded away from zero. Theoretically, the sampling probabilities that are nearly zero violate the assumptions of boundedness listed in section 2.3. Practically, since we calculate the sampling probability using the two steps algorithm(Algorithm 3), i.e., the sampling probability takes values on a subsample instead of the full data, the hessian matrix $\mathbf{H}$ could be poor conditioned, which leads to the small values in the sampling probability. In this case, observations that could have been selected in real life won't be selected due to the calculated zero sampling probability. To alleviate this issue, we can treat all the observations with the calculated zero sampling probabilities equally, i.e., we select them uniformly instead of skipping all of them, that means a truncation on the sampling distribution can be applied. In details, first we summarize the sampling probability $\pi = (\pi_1, \pi_2, ..., \pi_n)$, check if there are zeros or nearly zero values. If they exist, we calculate the total number of them and calculate their percentage p_0, then truncate $\boldsymbol{\pi}$ as

$$\pi_i^{trunc} = \pi_i \mathbf{1}[\pi_i > p_0] + p_0 \mathbf{1}[\pi_i \leq p_0], \quad i = 1, 2, ..., n.$$

In our application of the sampling probability truncation in simulation studies in section 4 and real data applications in section 5, we truncated at most 25% of the sampling probability in few cases, as normally zero values happens before the 25% quantile of the sampling probability. For the cases where our optimal sampling probability doesn't work, i.e., doesn't perform better than the uniform sampling, truncation will be a remarkable improvement. However, for cases where our optimal sampling probability already works, truncation weakens the performance of the optimal sampling probability hence is not necessary.

4. SIMULATION STUDIES

In this chapter, we shall apply the optimal subsampling method to specific penalized spline single index models on simulated datasets Dataset 1-Dateset 4 shown below. Note that we calculate the sampling probability distribution using $\tilde{\pi}$ in (3.3) and the A-optimal Scoring Algorithm in this chapter and chapter 5, however we also give the simulated MSE results using π in (3.1) in the last section of this chapter for a comparison.

Dataset 1. Consider the following single index model,

$$y_i = (\mathbf{x}_i^T \boldsymbol{\beta}_0)^2 e^{\mathbf{x}_i^T \boldsymbol{\beta}_0} + \sigma_0 \epsilon_i, \quad i = 1, 2, ..., n$$

where $\mathbf{x}_i = (x_{i1}, x_{i2}, \cdots, x_{ip})$ was generated from $N(0, \mathbf{\Sigma})$ with $\mathbf{\Sigma} = (\Sigma_{jk}) = (0.5^{|j-k|})$, and ϵ_i's are independent and identically distributed as the standard normal $N(0, 1)$. We chose the sample size $n = 100,000$, the covariate dimension $p = 12$, the true parameter $\boldsymbol{\beta}_0$ equals to a normalized vector of (1,0.001,0.001) repeated $p/3$ times, and $\sigma_0 = 1$.

Dataset 2. Same as Dataset 1 except that $\mathbf{x_i}$'s were generated from the normal mixture $0.8N(0, \mathbf{\Sigma}) + 0.2N(0, 10\mathbf{\Sigma})$.

Dataset 3. Same as Dataset 1, except that $\mathbf{x_i}$'s were generated from t- distribution with 8 degrees of freedom.

Dataset 4. Consider the following model

$$y_i = (\mathbf{x}_i^T \boldsymbol{\beta}_0)^3 + 5 \sin(\mathbf{x}_i^T \boldsymbol{\beta}_0) + \sigma_0 \epsilon_i, \quad i = 1, 2, ..., n,$$

where $\sigma_0 = 0.1$, $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{ip})$ with $x_{i1} \sim binomial(0.4)$, $x_{i2} \sim binomial(0.6)$, $x_{i3} \sim binomial(0.8)$, $(x_{i4}, \cdots, x_{ip}) \sim N(0, \mathbf{\Sigma}_{(p-3)\times(p-3)})$ with $\Sigma_{jk} = 0.5^{|j-k|}$, $j, k = 4, 5, ..., p-3$, while $\boldsymbol{\beta}_0$, n and p were the same as in Dataset 1.

We variate $\mathbf{X}$ in Datasets 1, 2 and 3, as the theories have no specific assumption on the distribution of $\mathbf{X}$. Dataset 4 has different true univariate marginal distributions, and includes discrete binary predictor variables.

Next, we shall apply the subsampling method on the specific penalized spline single index models.

4.1 The Penalized B-spline SIM

Given κ knots, i.e., a nondecreasing sequence $\mathbf{t} := \{t_i\}_{i=0}^{\kappa+1}$ such that

$$t_0 \leq t_1 \leq \cdots \leq t_{\kappa+1},$$

i.e., there are κ interior knots. The augumented knots set $\{t_i\}_{i=1-m}^{\kappa+m}$ is defined by

$$t_{-(m-1)} = \cdots = t_{-1} = t_0 \leq t_1 \leq \cdots \leq t_\kappa \leq t_{\kappa+1} = \cdots = t_{\kappa+m}.$$

Reset the index of knots to obtain $\{t_i\}_{i=0}^{\kappa+2m-1}$, for the the B-spline of order m (of degree $m-1$), $\{B_{ij}\}_{j=0,1,\cdots,m-1}$ is defined by recurrence:

$$B_{i0}(t) := \begin{cases} 1, & \text{if } t_i \leq t < t_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

$$B_{i,j+1} := \omega_{ij+1} B_{i,j} + (1 - \omega_{i+1,j+1}) B_{i+1,j} \tag{4.2}$$

with

$$\omega_{ij}(t) := \begin{cases} \frac{t-t_i}{t_{i+j}-t_i}, & \text{if } t_i \neq t_{i+j} \\ 0, & \text{otherwise} \end{cases}$$

Note that these functions are right-continuous.

Example 4.1.1 *Fourth order B spline(cubic spline) with three interior knots Suppose there are $\kappa = 3$ interior knots uniformly spaced between 0 and 1, i.e. $(0.25, 0.5, 0.75)$, and the boundary knots are (0, 1). The degree of the spline is $N = 3$, hence the order is $m = 4$. The sequence of knots needed to construct the B spline is*

$$(0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1),$$

the number of basis functions is $\kappa + m = 7$. The seven cubic spline basis functions are denoted by $B_{03}, ..., B_{63}$, which are calculated iteratively in the following order (left to right and top to below)

$$B_{00}, B_{10}, B_{20}, B_{30}, B_{40}, B_{50}, B_{60}, B_{70}, B_{80}, B_{90}$$

$$B_{01}, B_{11}, B_{21}, B_{31}, B_{41}, B_{51}, B_{61}, B_{71}, B_{81}$$

$$B_{02}, B_{12}, B_{22}, B_{32}, B_{42}, B_{52}, B_{62}, B_{72}$$

$$B_{03}, B_{13}, B_{23}, B_{33}, B_{43}, B_{53}, B_{63}$$

Figure 4.1 shows the graphs for functions $B_{03} = B_{03}(t), B_{13} = B_{13}(t), B_{23} = B_{23}(t), B_{33} = B_{33}(t), B_{43} = B_{43}(t), B_{53} = B_{53}(t), B_{63} = B_{63}(t)$, respectively.

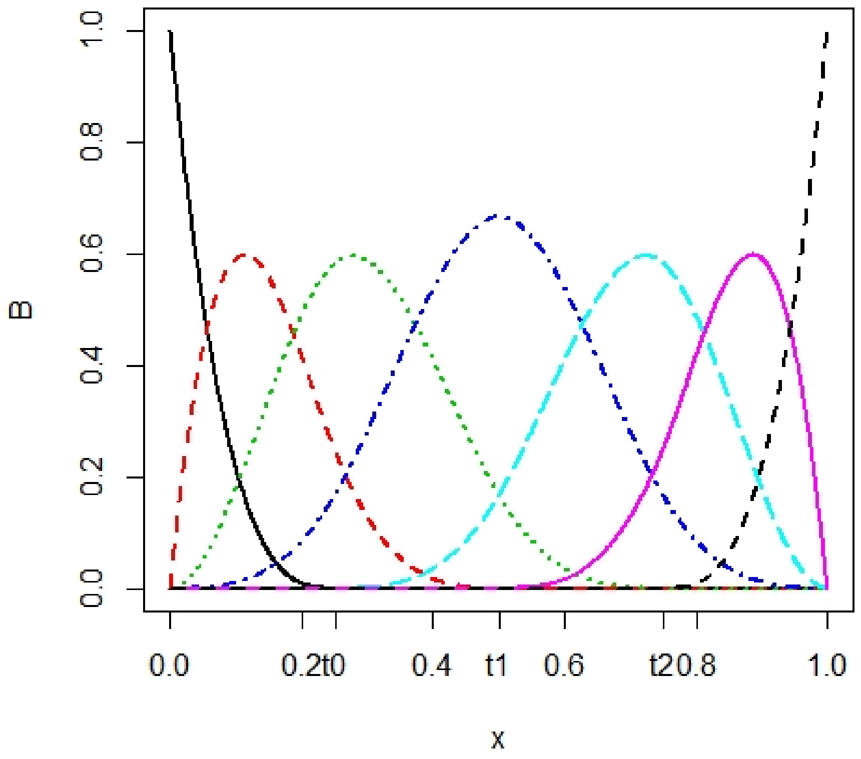

Fig. 4.1.: Plot of cubic B spline functions in Example 4.1.1

By definition, a B-spline of order m (degree $N = m-1$) with knots $\mathbf{t}$ (with length $\kappa+2$, i.e., κ interior knots), is a linear combination of the B-splines B_{iN}, as described in Section 2.1, $\boldsymbol{\delta}^T\mathbf{B}(t) = \sum_{i=0}^{\kappa+N}\delta_i B_{iN}(t)$.

In literature, knots are selected as equally spaced quantiles of indices, and the larger the number of knots, the more flexible the curve fitting is to different data sets. To avoid overfitting, O'Sullivan(1986, 1988) proposed the roughness penalty

$$P_\lambda(\boldsymbol{\theta}) = \int_{t_{\min}}^{t_{\max}} \{\sum_{i=0}^{\kappa+N}\delta_i B''_{iN} w(t)\}^2\, dt. \tag{4.3}$$

It has been proven that it is equivalent to the second order difference penalty in Eiler and Marx (1996). In this section, we apply the subsampling method on the B spline SIM with this penalty. Clearly the Lipschitz condition in Assumption A2 is met. With the penalized B spline, the objective function (2.7) becomes

$$\begin{aligned} Q(\boldsymbol{\theta}_{\boldsymbol{\phi}}) &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{\delta}^T\mathbf{B}(\boldsymbol{\beta}^T\mathbf{x}_i))^2 + \lambda\int_{(\boldsymbol{\beta}^T\mathbf{x}_i^*)_{\min}}^{(\boldsymbol{\beta}^T\mathbf{x}_i^*)_{\max}}\{\sum_{i=0}^{\kappa+N}\delta_i\mathbf{B}''_{iN}w(t)\}^2\, dt \\ &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{\delta}^T\mathbf{B}(\boldsymbol{\beta}^T\mathbf{x}_i))^2 + \lambda\sum_{i=1}^{\kappa+N}\sum_{j=1}^{\kappa+N}\delta_i\delta_j\int_{(\boldsymbol{\beta}^T\mathbf{x}_i^*)_{\min}}^{(\boldsymbol{\beta}^T\mathbf{x}_i^*)_{\max}}\mathbf{B}''_{iN}(t)\mathbf{B}''_{jN}(t) \\ &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{\delta}^T\mathbf{B}(\boldsymbol{\beta}^T\mathbf{x}_i))^2 + \lambda\boldsymbol{\delta}^T\boldsymbol{\Omega}\boldsymbol{\delta}, \end{aligned} \tag{4.4}$$

where $\Omega = \Omega_{(\kappa+N)\times(\kappa+N)}$ with $\Omega_{ij} = \int_{(\boldsymbol{\beta}^T\mathbf{x}_i^*)_{\min}}^{(\boldsymbol{\beta}^T\mathbf{x}_i^*)_{\max}} B''_{iN}(t)B''_{jN}(t)$.

The minimizer $\hat{\boldsymbol{\theta}} = \begin{pmatrix}\hat{\boldsymbol{\beta}}\\ \hat{\boldsymbol{\delta}}\end{pmatrix}$ satisfies $\dot{Q}_n(\hat{\boldsymbol{\theta}}) = 0$ with

$$\dot{Q}_n(\boldsymbol{\theta}) = \begin{pmatrix}\frac{\partial Q(\boldsymbol{\theta})}{\partial\phi}\\ \frac{\partial Q(\boldsymbol{\theta})}{\partial\delta}\end{pmatrix}.$$

Let

$$\frac{\partial Q(\boldsymbol{\theta})}{\partial\delta} = 0,$$

then

$$\boldsymbol{\delta} = (\mathbf{B}^T\mathbf{B} + n\lambda\Omega)^{-1}\mathbf{B}\mathbf{y},$$

where $\mathbf{B} = (B^T(\mathbf{x}_1^T\boldsymbol{\beta}(\boldsymbol{\phi}), \cdots, B^T(\mathbf{x}_n^T\boldsymbol{\beta}(\boldsymbol{\phi})))^T$. Hence the dimension of the optimization is $p-1$.

The Hessian matrix is usually unavailable as it is too computationally expensive to calculate, while quasi-newton method is a popular optimization method for approximating the hessian matrix with only gradient information. We use the quasi-newton method instead of the traditional newton's method. This is implemented in the BFGS algorithm in the "optim" package in R. See Dennis and Schnabel (1983) for discussion of the properties of BFGS.

We compare our proposed A-optimal subsampling method with the uniform subsampling in the penalized B spline single index model on the four simulated datasets. We use the cubic spline, i.e., $N = 4$, and the number of interior knots was chosen to be 10. Cubic splines are commonly used for its simplicity and smoothness (Lipschitz continuous second order derivative) properties.

Before we proceed, we want to have an idea about choices or possible values of the first step subsample size r_0 in the A-optimal Scoring Algorithm 3. To find out, we take a range values of r_0 to see how it changes the performance of the subsampling method.

Practically, r_0 should be as small as possible compared to n, while keeping the subsample estimation on the subsample of size r_0 not too losing much efficiency. From this point of view, we can take r_0 to be one of the values

$$100(.001n),\ 300(.003n),\ 500(.005n), 000(.01n),\ 5000(.05n).$$

For each r_0, given a subsample size r, repeat the subsampling estimation $B = 500$ times to obtain the mean squared error MSE of the subsample estimator, given by

$$\text{MSE} = \frac{1}{B}\sum_{i=1}^{B} \|\hat{\boldsymbol{\beta}}^{*i} - \hat{\boldsymbol{\beta}}\|^2 \tag{4.5}$$

where $\hat{\boldsymbol{\beta}}^{*i}$ is the subsampling estimator in the i-th repetition. We use MSE_{optim} to denote MSE values of the subsampling estimator using our optimal subsampling method, and MSE_{unif} for the uniform subsampling method.

Reported in Table 4.1-4.6 are the results of the B-spline single index model on Dataset 1. We can see that 100(.1%n) to 500(.5%n) are good choices for r_0, which usually have the smallest MSE values, otherwise, the MSE values are quite consistent for all r_0. In the following simulations, we choose $r_0 = 500(.5\%n)$.

Table 4.1.: Simulated MSE for $r = 100(.1\%n)$

r_0	100(.1n%)	300(.3n%)	500(.5n%)	1000(1n%)	3000(3n%)	5000(5n%)
MSE_{opt}	0.7726	1.5404	1.1943	2.8854	2.3551	2.1133
MSE_{unif}	1.4195	1.4337	1.3802	1.4273	1.4238	1.5157

Table 4.2.: Simulated MSE for $r = .3\%n$

r_0	100(.1%n)	300(.3%n)	500(.5%n)	1000(1%n)	3000(3%n)	5000(5%n)
MSE_{opt}	0.000287	0.000301	3e-04	0.000305	0.001998	0.036039
MSE_{unif}	0.003712	0.006765	0.009715	0.009005	0.012306	0.003329

Table 4.3.: Simulated MSE for $r = .5\%n$

r_0	100(.1%n)	300(.3%n)	500(.5%n)	1000(1%n)	3000(3%n)	5000(5%n)
MSE_{opt}	1.5298	0.3516	0.9006	1.1283	2.7568	1.3495
MSE_{unif}	1.0338	0.9987	0.9061	1.0376	0.9829	1.0378

Table 4.4.: Simulated MSE for $r = 1\%n$

r_0	100(.1%n)	300(.3%n)	500(.5%n)	1000(1%n)	3000(3%n)	5000(5%n)
MSE_{opt}	0.0002837	0.0002976	0.0002971	0.0002403	0.000294	0.0002829
MSE_{unif}	0.0022425	0.0022611	0.0021868	0.00223	0.0022668	0.0023012

Table 4.5.: Simulated MSE for $r = 3\%n$

r_0	100(.1%n)	300(.3%n)	500(.5%n)	1000(1%n)	3000(3%n)	5000(5%n)
MSE_{opt}	0.0002564	0.0002716	0.0002638	0.0002608	0.001021	0.0002607
MSE_{unif}	0.0027495	0.0026211	0.0025745	0.0024038	0.0025238	0.0025354

Table 4.6.: Simulated MSE for $r = 5\%n$

r_0	100(.1%n)	300(.3%n)	500(.5%n)	1000(1%n)	3000(3%n)	5000(5%n)
MSE_{opt}	0.000187	0.000264	0.000261	0.000262	0.000259	0.000262
MSE_{unif}	0.002432	0.002722	0.002682	0.002567	0.002524	0.002529

We then calculate the subsampling estimator based on the subsample from the uniform and the A-optimal subsampling method, respectively, and repeat $B = 500$ times to check the performance of the subsampling method by comparing the MSE in (4.5). That is, calculate the ratio

$$MSE_{ratio} = \frac{MSE\_opt}{MSE\_unif}.$$

If MSE_{ratio} is less than 1, then the A-optimal subsampling performs more efficiently than the uniform, hence the optimization works. According to The A-optimal Scoring Algorithm 3 in Section 3, choose the first step size $r_0 = 500(.5\%n)$ to calculate the subsampling probability distribution, then go to the second step for the subsampling estimator variating from 100(.1n%) to 5000(5n%).

The simulated MSE for Dataset 1 is reported Table 4.7 and Figure 4.2. We can observe that MSE decreases as r increases until r reaches 3000(3%n), it becomes stable. The MSE ratios are consistently and significantly less than 1. From Table 4.8, we can see that the subsampling method saved significant amount of time compared to the full sample estimation for the subsample sizes considered while kept desirable accuracy in terms of the MSE. The optimal subsampling estimate takes a bit less time than the uniform subsampling estimate but needs extra time to calculate the sampling probability distribution, which is 25.67 seconds, it is significantly small compared to the time 1174.33 seconds used in the full sample estimation.

The simulation results for Dataset 2 are reported in Tables 4.9, 4.10 and Figure 4.3. They are similar to Dataset 1. Specifically, the MSE decreases as r increases until r reaches $3\%n$ where the MSE becomes stable. The MSE ratios are consistently less than 1. Similar conclusions can be drawn for Tables 4.11, 4.12 and Figure 4.4 for Dataset 3.

The simulation results for Dataset 4 are listed in Table 4.13, Figure 4.5 and Table 4.14. Similarly, the MSE decrease as r increases. The MSE ratios are smaller than 1 except that MSE=452.36 for $r = 100(.1\%n)$, which is much larger than 1. We can relate this to the correspondingly large bias ratio 33.01. That is, when subsample size is too small, the optimal subsampling doesn't include enough information to beat the uniform subsampling.

Table 4.7.: Simulated MSE of the optimal subsampling estimator and the uniform subsample estimator and their ratios under different subsample sizes for B-spline SIM for Dataset 1 with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

r	MSE_opt	MSE_unif	MSE_ratio	Bias_opt	Bias_unif	Bias_ratio
100(.1%n)	0.007611	0.246294	0.030902	0.000361	0.020007	0.018049
300(.3%n)	0.000301	0.010877	0.027708	0.000277	0.000419	0.660328
500(.5%n)	0.000298	0.002183	0.136721	0.000279	0.000363	0.769482
1000(1%n)	0.000290	0.002175	0.133564	0.000275	0.000407	0.676333
3000(3%n)	0.000267	0.002454	0.108674	0.000257	0.000310	0.828419
5000(5%n)	0.000262	0.002598	0.100932	0.000255	0.000277	0.919580

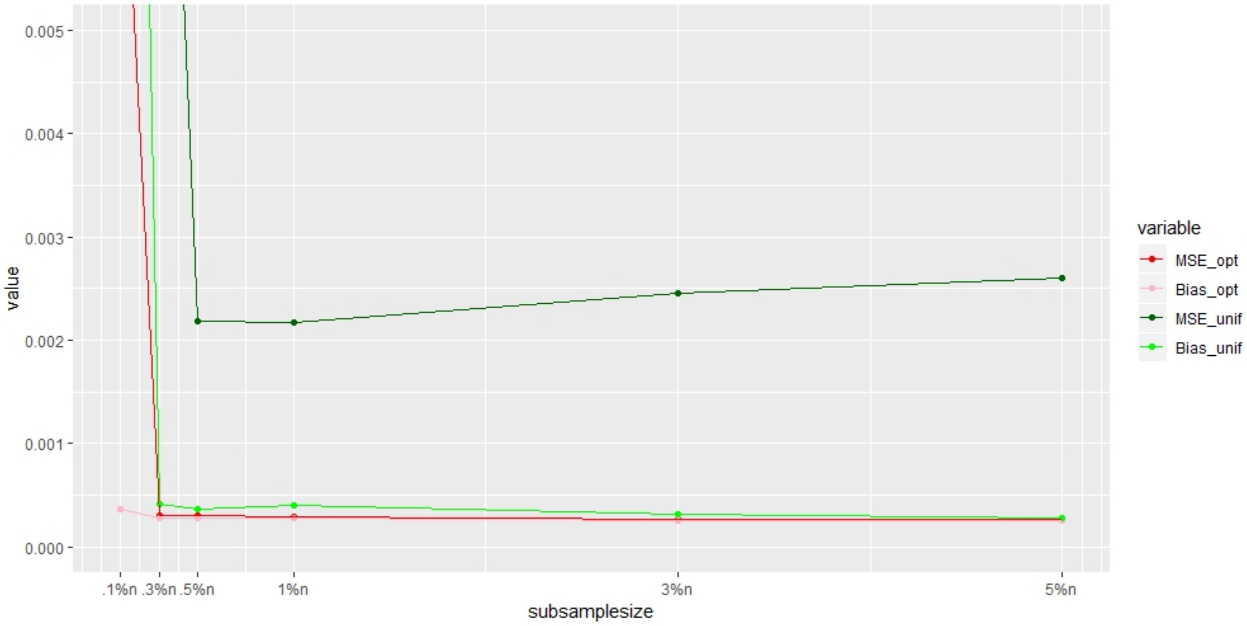

Fig. 4.2.: Plot of MSE and bias values from Table 4.7

Table 4.8.: The average time (in seconds) taken to calculate the subsampling estimator under each subsample size for B-spline SIM on Dataset 1(the full sample estimation takes 1174.33s, calculating $\tilde{\boldsymbol{\pi}}$ (first step) takes 25.67s)

r	Time_opt	Time_unif
100(.1%n)	31.4029	50.2206
300(.3%n)	5.9084	7.5738
500(.5%n)	6.1495	7.4472
1000(1%n)	8.3305	10.0831
3000(3%n)	10.5931	15.9996
5000(5%n)	9.0264	15.1435

Table 4.9.: Simulated MSE of the optimal subsampling estimator and the uniform subsample estimator and their ratios under different subsample sizes for B-spline SIM for Dataset 2 with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

r	MSE_opt	MSE_unif	MSE_ratio	Bias_opt	Bias_unif	Bias_ratio
100(.1%n)	0.7437	0.8841	0.8412	0.2171	0.2273	0.9552
300(.3%n)	0.3003	0.4929	0.6092	0.0389	0.0847	0.4598
500(.5%n)	0.1363	0.3776	0.3611	0.0099	0.0615	0.1616
1000(1%n)	0.0555	0.2511	0.2212	0.0061	0.0460	0.1315
3000(3%n)	0.0231	0.1333	0.1730	0.0061	0.0386	0.1571
5000(5%n)	0.0315	0.1221	0.2577	0.0070	0.0364	0.1938

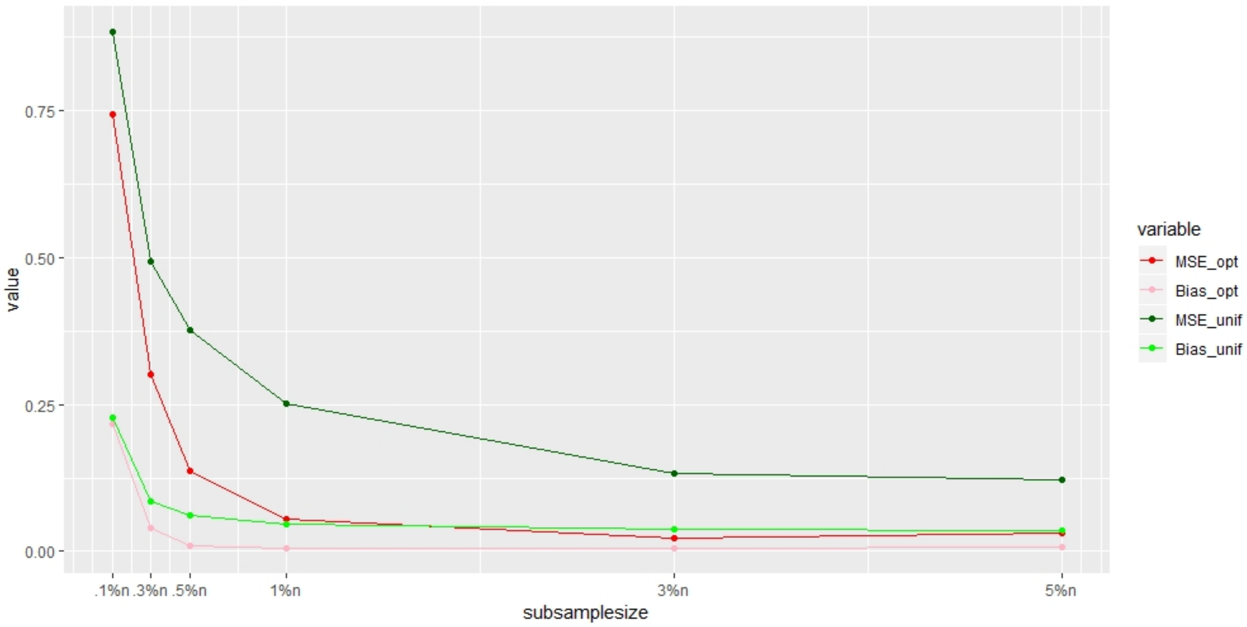

Fig. 4.3.: Plot of MSE and bias values from Table 4.9

Table 4.10.: The average time (in seconds) taken to calculate the subsampling estimator under each subsample size for B-spline SIM for Dataset 2 with $n = 100,000$ (the full sample estimation takes 1577.28s, calculating $\tilde{\boldsymbol{\pi}}$ (first step) takes 45.43s)

r	Time_opt	Time_unif
100(.1%n)	7.2712	7.9470
300(.3%n)	8.4250	8.3083
500(.5%n)	9.0337	8.6701
1000(1%n)	10.6185	11.3170
3000(3%n)	17.1332	21.2636
5000(5%n)	27.9052	31.9889

Table 4.11.: Simulated MSE of the optimal subsampling estimator and the uniform subsampling estimator and their ratios under different subsample sizes for B-spline SIM for Dataset 3 with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

r	MSE_opt	MSE_unif	MSE_ratio	Bias_opt	Bias_unif	Bias_ratio
100(.1%n)	1.1369	1.1717	0.9703	0.4036	0.7062	0.5715
300(.3%n)	0.7980	1.0016	0.7967	0.2128	0.7016	0.3033
500(.5%n)	0.5631	1.0015	0.5622	0.1541	0.7050	0.2186
1000(1%n)	0.4542	0.9873	0.4600	0.1413	0.7172	0.1970
3000(3%n)	0.3491	0.9230	0.3782	0.1367	0.7496	0.1824
5000(5%n)	0.3373	0.8728	0.3865	0.1217	0.6829	0.1781

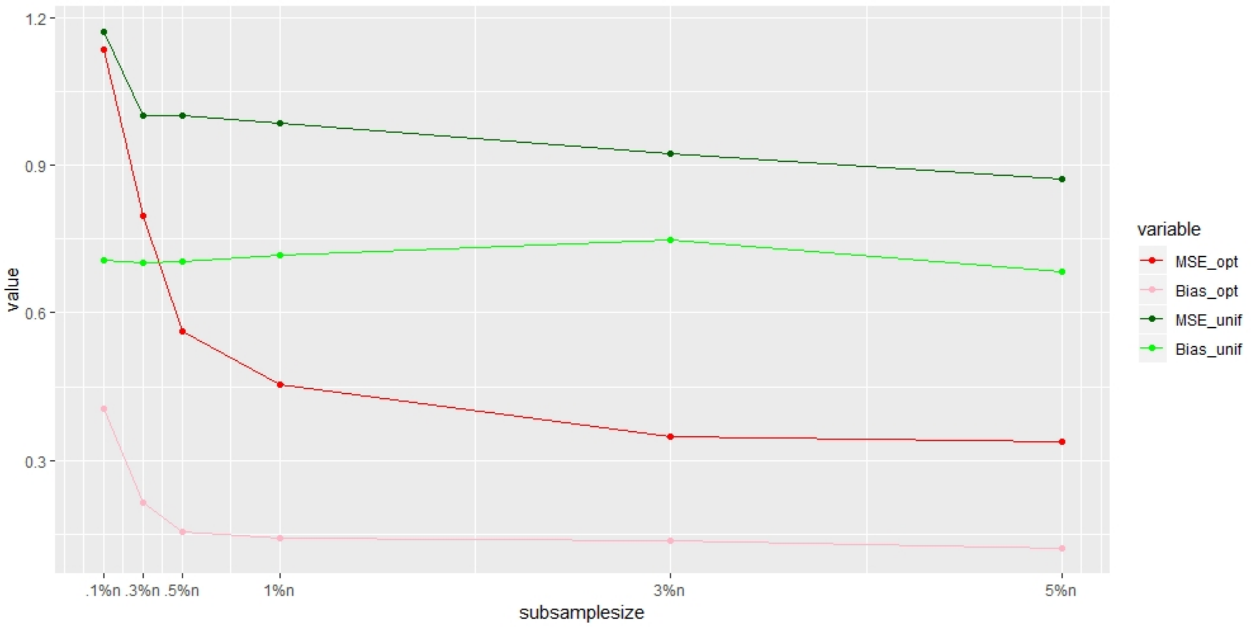

Fig. 4.4.: Plot of MSE and bias values from Table 4.11

Table 4.12.: The average time (in seconds) taken to calculate the subsampling estimator under each subsample size for B-spline SIM for Dataset 3 with $n = 100,000$ (the full sample estimation takes 1782.5s, calculating $\tilde{\boldsymbol{\pi}}$ (first step) takes 46.03s)

r	Time_opt	Time_unif
100(.1%n)	11.1171	6.9309
300(.3%n)	7.5731	7.8669
500(.5%n)	6.1142	9.2347
1000(1%n)	9.2911	14.0700
3000(3%n)	26.8115	45.8354
5000(5%n)	20.9743	34.5437

Table 4.13.: Simulated MSE of the optimal subsampling estimator and the uniform subsampling estimator and their ratios under different subsample sizes for B-spline SIM for Dataset 4 with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

r	MSE_opt	MSE_unif	MSE_ratio	Bias_opt	Bias_unif	Bias_ratio
100(.1%n)	0.0006863387	0.0000015172	452.3601813703	0.0000016776	0.0000000508	33.0193771030
300(.3%n)	0.0000008014	0.0000008333	0.9617051584	0.0000000153	0.0000000411	0.3714458355
500(.5%n)	0.0000004995	0.0000006737	0.7413186253	0.0000000204	0.0000000352	0.5797040200
1000(1%n)	0.0000002735	0.0000005465	0.5003661550	0.0000000206	0.0000000391	0.5270716127
3000(3%n)	0.0000001060	0.0000004159	0.2548913425	0.0000000195	0.0000000311	0.6288872545
5000(5%n)	0.0000000704	0.0000003252	0.2166383296	0.0000000180	0.0000000235	0.7681715531

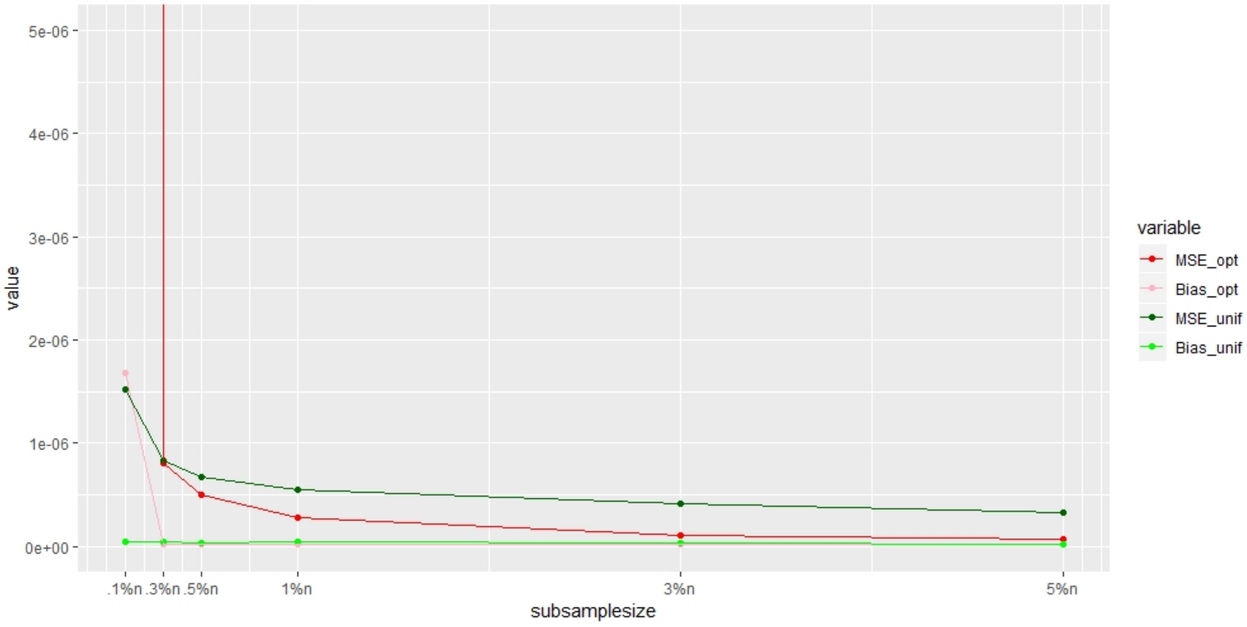

Fig. 4.5.: Plot of MSE and bias values from Table 4.13

Table 4.14.: The average time (in seconds) taken to calculate the subsampling estimator under each subsample size for B-spline SIM for Dataset 4 (the full sample estimation takes 545.98s, calculating $\tilde{\boldsymbol{\pi}}$ (first step) takes 22.67s)

r	Time_opt	Time_unif
100(.1%n)	6.13298	5.71258
300(.3%n)	5.40642	3.73712
500(.5%n)	5.65354	3.66320
1000(1%n)	6.63902	4.31884
3000(3%n)	10.58992	9.05540
5000(5%n)	14.77560	15.01004

4.2 The Penalized P-spline SIM

This model proposed by Yu and Ruppert 2002 estimates the univariate function η by a P-spline,

$$\eta(u) = \boldsymbol{\delta}^T \mathbf{B}(u),$$

where $\mu = \boldsymbol{\beta}^T \mathbf{x}$ is the index, $\boldsymbol{\delta} = (\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, ..., \boldsymbol{\delta}_{q+\kappa})^T$ is the spline coefficient vector, and the spline basis is

$$B(u) = (1, u, ..., u^q, (u - \kappa_1)_+^q, ..., (u - \kappa_K)_+^q)^T, \tag{4.6}$$

which is a truncated power basis. q is the order of spline basis, and K is the number of knots. The knots $\kappa_1, \kappa_2, ..., \kappa_K$ are selected to be the equally spaced sample quantiles of the index $\boldsymbol{\beta}^T \mathbf{x}$. Note that $q > 2$ is needed to ensure the second order differentiability of the spline basis function. When $q = 3$, the spline is called the cubic spline, which has Lipschitz continuous second order derivatives. As for the choice of number of knots K, Ruppert (2002) suggested that 5 to 10 knots are quite adequate for smooth and either monotonic or unimodal regression function.

They proposed the residual sum of squares plus the partial ridge penalty as the objective function, i.e.,

$$Q_n(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \boldsymbol{\delta}^T B(\boldsymbol{\beta}(\boldsymbol{\phi})^T \mathbf{x}_i))^2 + \lambda \boldsymbol{\delta}^T \mathbf{D} \boldsymbol{\delta}, \tag{4.7}$$

where $\mathbf{D}$ is a diagonal matrix with the last K diagonal entries being 1 and the rest being 0. This together with the definition of the spline basis (4.6) imply that the penalty parameter λ works to avoid overfitting by penalizing the last K elements of model parameter $\boldsymbol{\delta}$, which forces the fitted curve to bend toward the data points closely through the knots. The penalty function clearly satisfy the desired smoothness assmptions.

We apply the subsampling method on the P-spline SIM for Dataset 1 to Dataset 4, to evaluate the performance of the optimal subsampling method in the penalized P-spline SIM. We choose cubic spline, i.e., q=3, and the number of knots $\kappa = 10$.

For Dataset 1, the MSE values and running time results are reported in Table 4.15, Figure 4.6 and Table 4.16. We can see that the MSE's for the A-optimal and uniform sampling both decrease, and start from $r = 1000(1n\%)$, the MSE of the A-optimal subsampling estimate decreases faster so that the MSE ratios are around .2, which is much less than 1.

For Dataset 2, see Table 4.17, Figure 4.7 and Table 4.18. Start from $r = 300(.3n\%)$, the MSE_{ratio} values are consistently less than 1.

The results for Dataset 3 are in given in Table 4.19, Figure 4.8 and Table 4.20. The MSE and $bias$ values are quite stable for both the A-optimal and uniform subsampling methods. The MSE_{ratio} values are less than 1 for all r's.

For Dataset 4, the results are in Table 4.21, Figure 4.9 and Table 4.22. We can see that MSE ratios are less than 1 from $r = 500(.5\%n)$. When $r = 100(.1\%n)$, the MSE ratio is relatively larger than other values. This could be caused by the corresponding large bias ratio. Specifically, when $r = 100(.1\%n)$, the optimal subsampling estimate has much larger bias than uniform sampling estimate and it can not be compensated by minimizing the variance.

For all datasets, the amounts of time saving are very significant as far as the subsample sizes considered.

Table 4.15.: Simulated MSE of the optimal subsampling estimator and the uniform subsampling estimator and their ratios under different subsample sizes for P-spline SIM for Dataset 1 with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

r	MSE_opt	MSE_unif	MSE_ratio	Bias_opt	Bias_unif	Bias_ratio
100(.1%n)	0.5601926	0.2587949	2.1646196	0.1046895	0.0220413	4.7496934
300(.3%n)	0.1042635	0.0120707	8.6377527	0.0084708	0.0011116	7.6204553
500(.5%n)	0.0137123	0.0038783	3.5356218	0.0012988	0.0010875	1.1943317
1000(1%n)	0.0010950	0.0039514	0.2771201	0.0010885	0.0011457	0.9501409
3000(3%n)	0.0011112	0.0041497	0.2677879	0.0011090	0.0008217	1.3495989
5000(5%n)	0.0010887	0.0038210	0.2849246	0.0010871	0.0006551	1.6594412

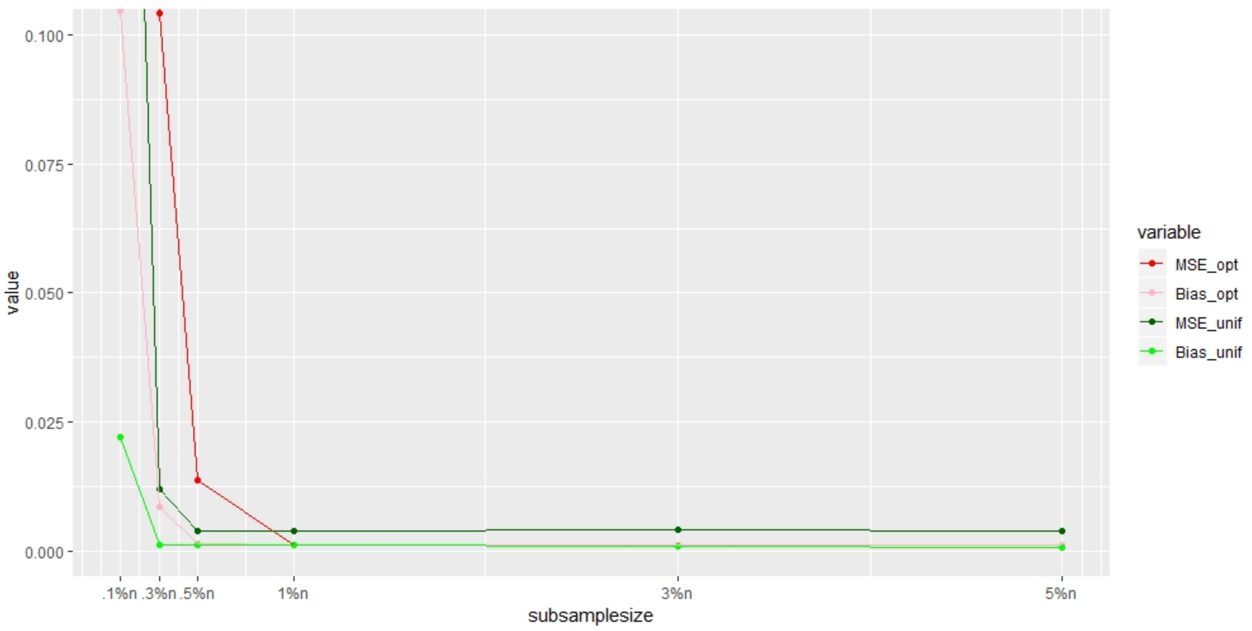

Fig. 4.6.: Plot of MSE and bias values from Table 4.15

Table 4.16.: The average time (in seconds) taken to calculate the subsampling estimator under each subsample size for P-spline SIM for Dataset 1 (the full sample estimation takes 3357.25s, calculating $\tilde{\boldsymbol{\pi}}$ (first step) takes 46.51s)

r	Time_opt	Time_unif
100(.1%n)	4.16990	3.24740
300(.3%n)	5.34612	3.37662
500(.5%n)	4.65214	3.40784
1000(1%n)	2.64888	2.30686
3000(3%n)	5.66108	8.02662
5000(5%n)	8.76270	14.72636

Table 4.17.: Simulated MSE of the optimal subsampling estimator and the uniform subsampling estimator and their ratios under different subsample sizes for P-spline SIM for Dataset 2 with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

r	MSE_opt	MSE_unif	MSE_ratio	Bias_opt	Bias_unif	Bias_ratio
100(.1%n)	1.28833	1.28740	1.00072	0.44064	0.43423	1.01477
300(.3%n)	0.77724	0.96087	0.80889	0.20310	0.27307	0.74374
500(.5%n)	0.66962	0.79219	0.84528	0.22648	0.20932	1.08198
1000(1%n)	0.49564	0.62578	0.79203	0.20184	0.17232	1.17131
3000(3%n)	0.22898	0.43788	0.52293	0.07403	0.11969	0.61848
5000(5%n)	0.15672	0.38733	0.40462	0.05543	0.10607	0.52257

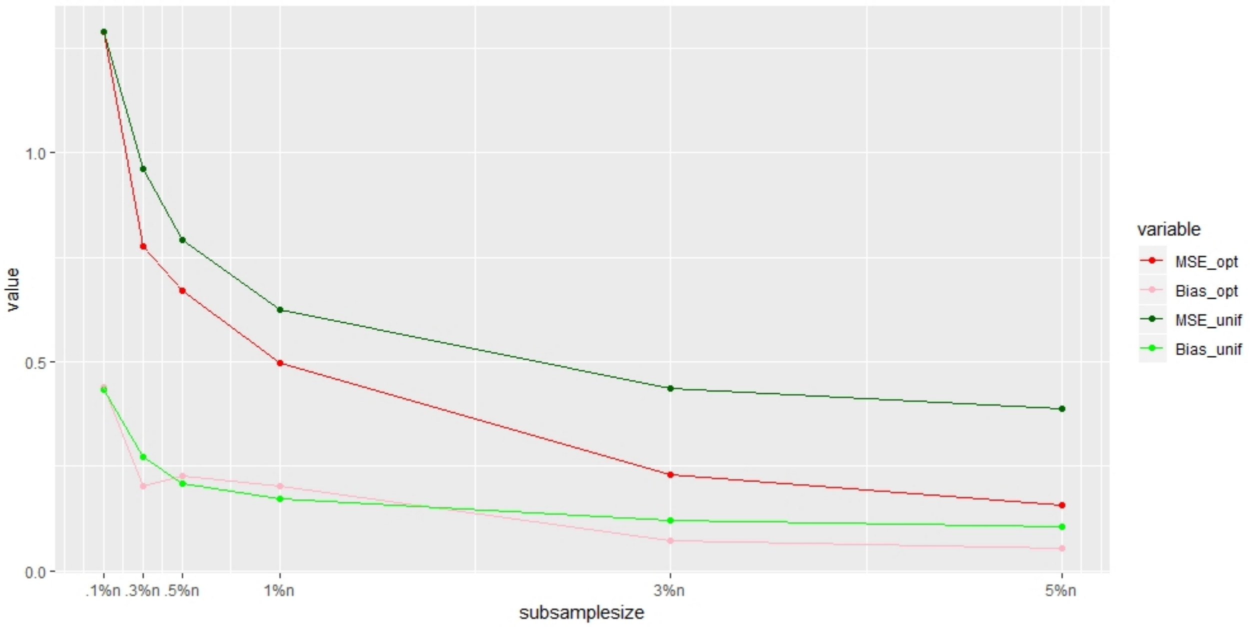

Fig. 4.7.: Plot of MSE and bias values from Table 4.17

Table 4.18.: The average time (in seconds) taken to calculate the subsampling estimator under each subsample size for P-spline SIM for Dataset 2 (the full sample estimation takes 3477.2s, calculating $\tilde{\boldsymbol{\pi}}$ (first step) takes 17.71s)

r	Time_opt	Time_unif
100(.1%n)	2.35708	1.99488
300(.3%n)	3.66070	3.38116
500(.5%n)	5.17740	4.71838
1000(1%n)	10.57100	8.18670
3000(3%n)	57.00426	35.08972
5000(5%n)	73.81938	43.59866

Table 4.19.: Simulated MSE of the optimal subsampling estimator and the uniform subsampling estimator and their ratios under different subsample sizes for P-spline SIM for Dataset 3 with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

r	MSE_opt	MSE_unif	MSE_ratio	Bias_opt	Bias_unif	Bias_ratio
100(.1%n)	0.93231	1.80099	0.51767	0.42049	0.87001	0.48332
300(.3%n)	0.64008	1.82378	0.35096	0.41384	0.88681	0.46666
500(.5%n)	0.62072	1.82251	0.34058	0.42452	0.88969	0.47715
1000(1%n)	0.56786	1.87309	0.30317	0.43791	0.94684	0.46250
3000(3%n)	0.54606	1.85944	0.29367	0.41962	0.93361	0.44946
5000(5%n)	0.53593	1.94236	0.27591	0.41756	1.01757	0.41035

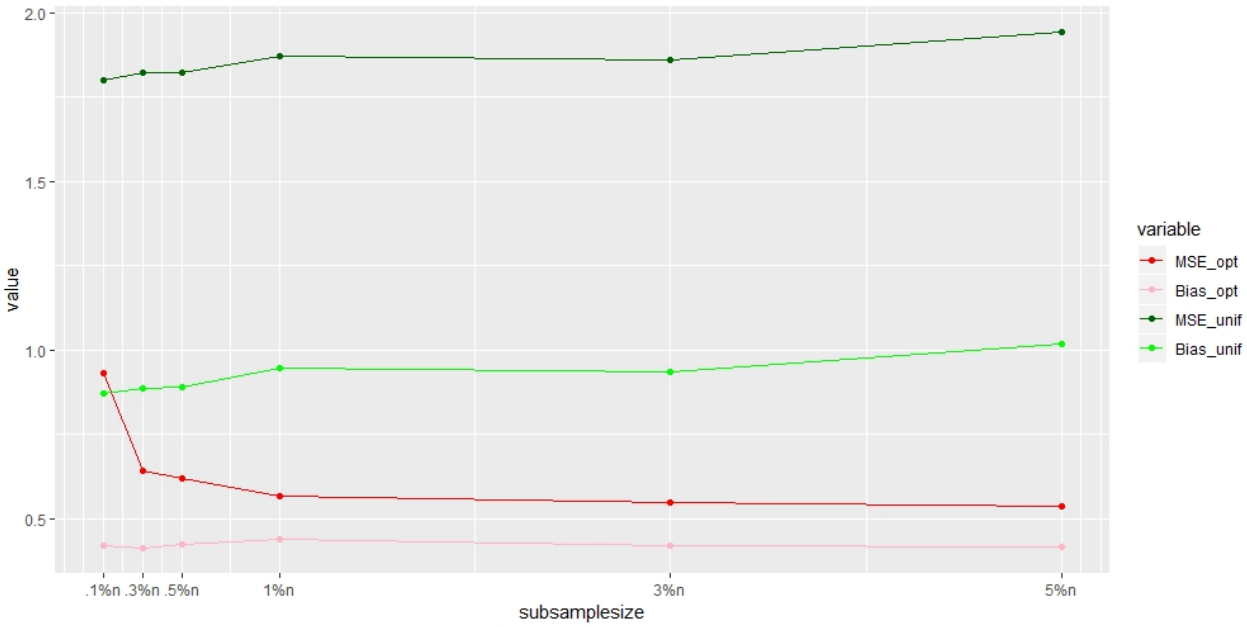

Fig. 4.8.: Plot of MSE and bias values from Table 4.19

Table 4.20.: The average time (in seconds) taken to calculate the subsampling estimator under each subsample size for P-spline SIM for Dataset 3 (the full sample estimation takes 44891.23s, calculating $\tilde{\boldsymbol{\pi}}$ (first step) takes 39.83s)

r	Time_opt	Time_unif
100(.1%n)	1.21686	1.34630
300(.3%n)	1.84940	2.62212
500(.5%n)	2.54674	4.22754
1000(1%n)	4.06008	7.58898
3000(3%n)	10.37228	19.86076
5000(5%n)	17.91362	34.76936

Table 4.21.: Simulated MSE of the optimal subsampling estimator and the uniform subsampling estimator and their ratios under different subsample sizes for P-spline SIM for Dataset 4 with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

r	MSE_opt	MSE_unif	MSE_ratio	Bias_opt	Bias_unif	Bias_ratio
100(.1%n)	0.0021188076	0.0000019481	1087.6187418199	0.0000087152	0.0000000388	224.8574914655
300(.3%n)	0.0000021381	0.0000011712	1.8254728788	0.0000000215	0.0000000355	0.6060937733
500(.5%n)	0.0000010284	0.0000011311	0.9091832430	0.0000000256	0.0000000266	0.9598913288
1000(1%n)	0.0000006347	0.0000009029	0.7029792220	0.0000000341	0.0000000330	1.0330516252
3000(3%n)	0.0000002739	0.0000006590	0.4156326534	0.0000000477	0.0000000331	1.4387592479
5000(5%n)	0.0000001771	0.0000005388	0.3287335498	0.0000000455	0.0000000249	1.8284127784

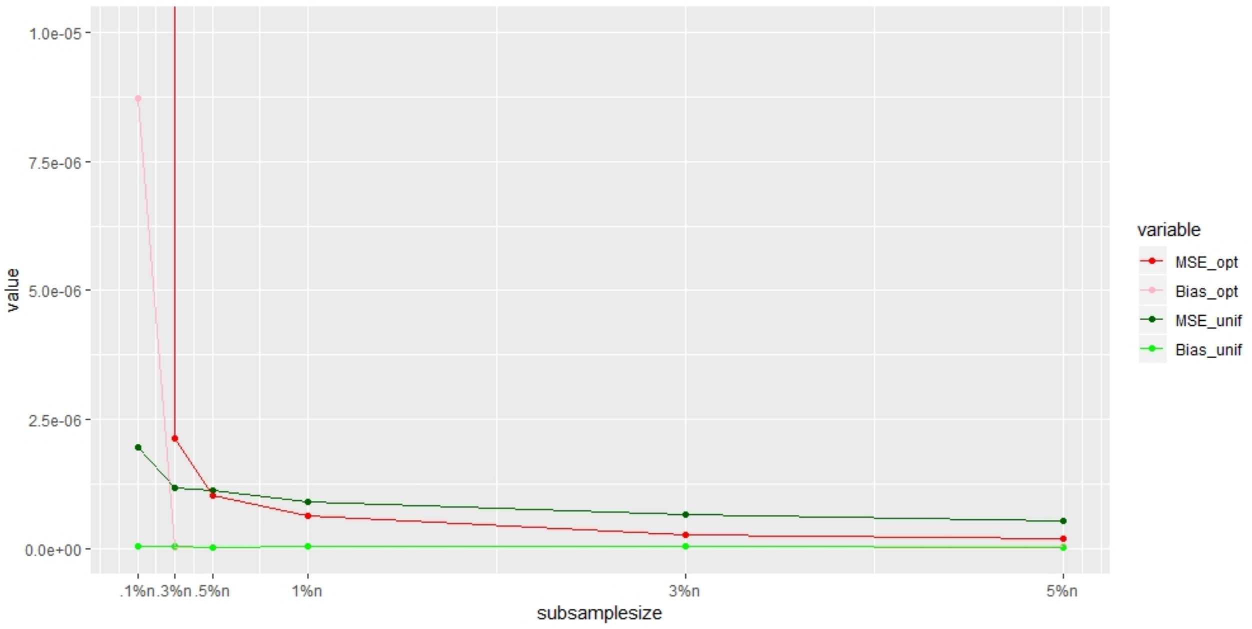

Fig. 4.9.: Plot of MSE and bias values from Table 4.21

Table 4.22.: The average time (in seconds) taken to calculate the subsampling estimator under each subsample size for P-spline SIM for Dataset 4 (the full sample estimation takes 975.5s, calculating $\tilde{\boldsymbol{\pi}}$ (first step) takes 13.45s)

r	Time_opt	Time_unif
100(.1%n)	0.62756	0.45714
300(.3%n)	0.82076	0.67632
500(.5%n)	1.10682	0.91296
1000(1%n)	1.83128	1.51506
3000(3%n)	4.92362	3.97588
5000(5%n)	8.20940	6.55708

Remark 4.2.1 In this thesis, we have considered the B spline SIM and the P spline SIM to investigate the performance of the A-optimal subsampling method in practice, we don't specifically analyze these two models though. Our goal is not to compare these two models either. However, if we only look at the TPF basis in the P spline and the B-spline basis, TPF as basis is simpler and practically useful for understanding spline regression, but is not numerically stable in the optimization when the number of knots is large, which increases the dimension of the unknown parameter, and when the penalty parameter λ is close to zero, which leads to extremely flexible curve fittings, and last, when the dataset is large. In this case, typical algorithms such as the Gauss-Newton algorithm (suggested by Yu and Ruppert 2002) doesn't work well. The B splines, however, are easy to calculate and numerically superior. For a close understanding of the difference of these two splines, see, e.g., Sharif and Kamal (2018). Both splines have problems for large datasets though as the dimension of the basis needs to increased correspondingly for precision, which increases the dimensionality of the optimization. From this point of view, our choice of using the subsampling method is quite suitable, hence the meaningfulness of the study of optimal subsampling follows.

4.3 Ridge Regression

As we discussed in Section 3, the A-optimal subsampling method should also work on linear regression. It is known that the single index model is an advanced generalized linear model with the unknown univariate function as its link, we want to consider linear regression as the simplest case of the single index model, for which, we conducted a simulation study for one type of linear regression, the ridge regression.

Ridge regression is the most popular technique to deal with multicollinearity problem in regression analysis. It was first proposed by Hoerl and Kennard (1970) to handle multicollinearity problem for engineering data. They discovered that with ridge parameter (penalty), the mean squared error for the ridge regression estimator is smaller than variance of the ordinary lease squares (OLS) estimator. To better describe the ridge regression, consider the multiple linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon,$$

where $\mathbf{y}$ is an n dimensional vector as the response variable, $\mathbf{X}$ is the $n \times p$ design matrix on the observed predictor variables, $\boldsymbol{\beta}$ is a p dimensional unknown vector parameter as regression coefficients, $\boldsymbol{\epsilon}$ is an n dimensional random error vector with mean $\mathbf{0}$ and covariance matrix $\sigma^2\mathbf{I}_n$, where $\mathbf{I}_n$ is the $n \times n$ identity matrix. The OLS estimator of $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

The covariance matrix of $\hat{\boldsymbol{\beta}}$ is $Cov(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$. So both $\hat{\boldsymbol{\beta}}$ and $Cov(\hat{\boldsymbol{\beta}})$ are dependent on the inverse of matrix $\mathbf{X}^T\mathbf{X}$, which is impractical as regressors can be dependent in real data. In this case, the matrix $\mathbf{X}^T\mathbf{X}$ is ill conditioned, i.e. $det(\mathbf{X}^T\mathbf{X}) \approx 0$, which can cause the sensitivity of $\hat{\boldsymbol{\beta}}$ to errors and hence difficulty of meaningful statistical inference. The propose of ridge regression by Hoerl and Kennard (1970) is to solve this problem by adding a small positive number to the diagonal elements of $\mathbf{X}^T\mathbf{X}$ to guarantee its invertibility, that is,

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{y}. \tag{4.8}$$

which is the solution to

$$\text{argmin}\{\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2 + \lambda\sum_{j=1}^{p}\beta_j^2\}. \tag{4.9}$$

We can see that ridge regression estimator is a biased estimator with a positive ridge parameter λ. The choice of λ is to conduct grid search. In Hoerl and Kennard's original paper, they introduced ridge traces, where they plotted the components of $\hat{\boldsymbol{\beta}}_\lambda$ against λ, then chose λ for which the coefficients are not rapidly changing and have sensible signs. The most common method now is the K-fold cross validation, the procedure is as follows:

I. Partition the data into K seperate sets of equal size, denote by $T = (T_1, T_2, ..., T_K)$. K is usually chosen to be 5 or 10;

II. For each $k = 1, 2, ..., K$, fit the model into the set excluding the k-th fold T_k;

III. Compute the cross-validation (CV) error for the k-th fold:

$$(CV)_k^{(\lambda)} = |T_k|^{-1}\sum_{y\in T_k}(y - \hat{y})^2,$$

then the overall cross-validation error is

$$CV^{(\lambda)} = K^{-1}\sum_{k=1}^{K}(CV)_k^{(\lambda)}.$$

From a grid of numbers, select one λ that has the minimum CV.

For the simulation study we simulate $\mathbf{X}_{n\times p}$ from $N(0, \boldsymbol{\Sigma}_p)$, where $\Sigma_{ij} = 0.5^{|i-j|}$, $n = 100,000$, $p = 12$; $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}_0 + \boldsymbol{\epsilon}$, where $\boldsymbol{\beta}_0 = (1, 0.001, 0.001, 1, 0.001, 0.001, \cdots 1, 0.001, 0.001)$ is the p-dimensional true parameter, $\boldsymbol{\epsilon} = (\epsilon_i)$, $i = 1, 2, ..., n$, ϵ_i's are independent and identically normally distributed with mean 0 and variance $\sigma_0^2 = 0.1$. Ridge regression can be implemented by ”glmnet” package in R. The simulation results are reported in Table 4.23 and Figure 4.10. The MSE ratios are consistently smaller than 1, showing significant improvement of the A-optimal subsampling method over the uniform subsampling method on the ridge regression model.

Table 4.23.: Simulated MSE of the optimal subsampling estimator and the uniform subsampling estimator and their ratios under different subsample sizes for ridge regression, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

r	MSE_opt	MSE_unif	MSE_ratio	Bias_opt	Bias_unif	Bias_ratio
100(.1%n)	0.003210	0.004004	0.801589	0.000583	0.000546	1.067013
300(.3%n)	0.000943	0.001572	0.599935	0.000107	0.000375	0.284744
500(.5%n)	0.000563	0.001268	0.444263	0.000128	0.000518	0.246232
1000(1%n)	0.000326	0.001008	0.323013	0.000120	0.000703	0.171106
3000(3%n)	0.000193	0.000819	0.235429	0.000122	0.000718	0.169760
5000(5%n)	0.000175	0.000775	0.225749	0.000127	0.000716	0.177417

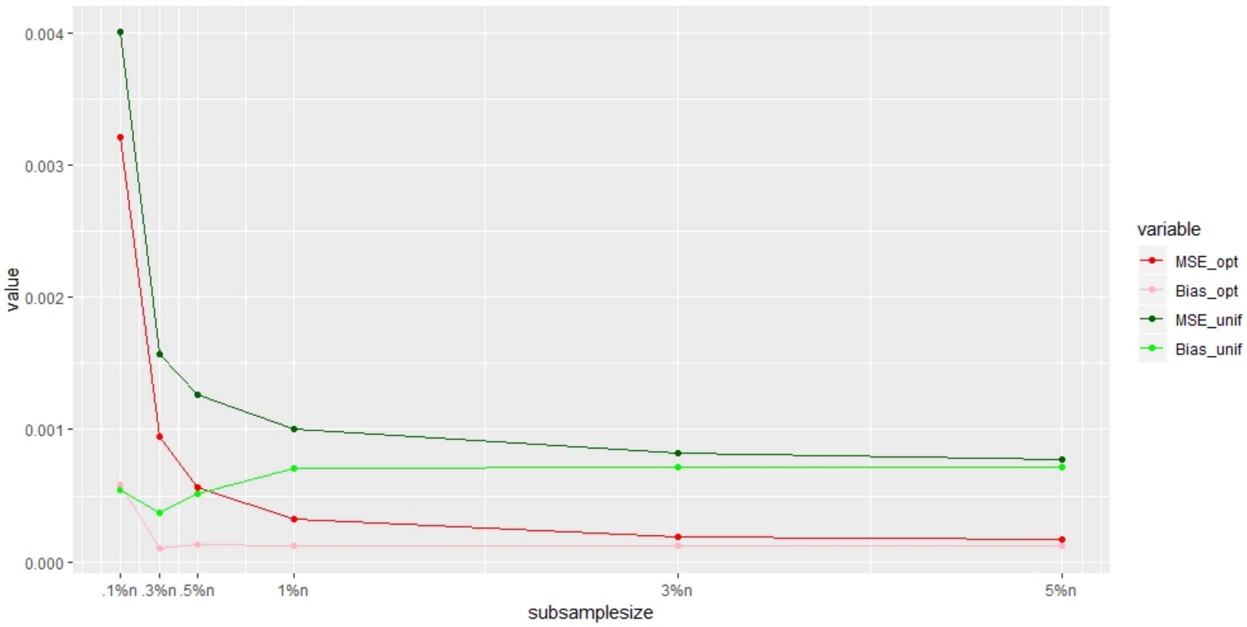

Fig. 4.10.: Plot of MSE and bias values from Table 4.23

4.4 Comparing Sampling Distributions

In this section, we give the simulated MSE results using the sampling probability distribution $\boldsymbol{\pi}$ in (3.1) for the B-spline SIM and the P-spline SIM respectively, and compare them with the results from Sections 4.1 and 4.2, which use the approximation $\tilde{\boldsymbol{\pi}}$ in (3.3) and are displayed in the columns under $\tilde{\boldsymbol{\pi}}$ in this section. Note that the MSE_{ratio} in this section still equals the ratio of MSE_{opt} over MSE_{unif} as in the previous sections.

The simulation results for the B-spline SIM are reported in Tables 4.24 – 4.27. Overall, the performance of $\tilde{\boldsymbol{\pi}}$ and $\boldsymbol{\pi}$ doesn't differ a lot. $\boldsymbol{\pi}$ performs a bit better than $\tilde{\boldsymbol{\pi}}$ for small r. This makes sense as $\tilde{\boldsymbol{\pi}}$ is the approximation of $\boldsymbol{\pi}$. For larger r, $\tilde{\boldsymbol{\pi}}$ catches up and has smaller MSE than $\boldsymbol{\pi}$. This is understandable as $\boldsymbol{\pi}$ itself is calculated approximately from a uniform small sample in the first step (see the A-optimal Scoring Algorithm 3). The error caused by this approximation becomes larger when the sample size r increases. Since $\tilde{\boldsymbol{\pi}}$ simplifies the calculation of $\boldsymbol{\pi}$, it reduces this error of approximation which turns out to benefit the performance of $\tilde{\boldsymbol{\pi}}$ compared to the original $\boldsymbol{\pi}$.

The above interpretations also apply to the simulation results for the P-spline SIM in Tables 4.28–4.31, except that the improving effect of $\tilde{\boldsymbol{\pi}}$ is more significant in Table 4.29 for Dataset 2, and Table 4.30 for Dataset 3, where the MSE_{opt} using $\tilde{\boldsymbol{\pi}}$ is smaller than the MSE_{opt} using $\boldsymbol{\pi}$ for all r. This difference is especially significant in Table 4.30 for Dataset 3 due to the large $Bias_{opt}$ using $\boldsymbol{\pi}$, even so, the MSE_{ratio} using $\boldsymbol{\pi}$ are all less than 1, which validates the theoretical result on the performance of our optimal sampling method.

Table 4.24.: Simulated MSE of the optimal subsampling estimator under different subsample sizes for B-spline SIM for Dataset 1 using $\tilde{\boldsymbol{\pi}}$ in (3.3) and $\boldsymbol{\pi}$ in (3.1) with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$

	$\tilde{\boldsymbol{\pi}}$			$\boldsymbol{\pi}$		
r	MSE_opt	Bias_opt	MSE_ratio	MSE_opt	Bias_opt	MSE_ratio
100(.1%n)	0.007611	0.000361	0.030902	0.039621	0.000789	0.173753
300(.3%n)	0.000301	0.000277	0.027708	0.001156	0.000142	0.233239
500(.5%n)	0.000298	0.000279	0.136721	0.000572	0.000110	0.280490
1000(1%n)	0.000290	0.000275	0.133564	0.000331	0.000075	0.157998
3000(3%n)	0.000267	0.000257	0.108674	0.000169	0.000039	0.064649
5000(5%n)	0.000262	0.000255	0.100932	0.000125	0.000028	0.045896

Table 4.25.: Simulated MSE of the optimal subsampling estimator under different subsample sizes for B-spline SIM for Dataset 2 using $\tilde{\boldsymbol{\pi}}$ in (3.3) and $\boldsymbol{\pi}$ in (3.1) with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

	$\tilde{\boldsymbol{\pi}}$			$\boldsymbol{\pi}$		
r	MSE_opt	Bias_opt	MSE_ratio	MSE_opt	Bias_opt	MSE_ratio
100(.1%n)	0.7437	0.2171	0.8412	0.3399	0.0817	0.3954
300(.3%n)	0.3003	0.0389	0.6092	0.1112	0.0120	0.2512
500(.5%n)	0.1363	0.0099	0.3611	0.1152	0.0145	0.2897
1000(1%n)	0.0555	0.0061	0.2212	0.1230	0.0384	0.4676
3000(3%n)	0.0231	0.0061	0.1730	0.0385	0.0066	0.2660
5000(5%n)	0.0315	0.0070	0.2577	0.0427	0.0057	0.3117

Table 4.26.: Simulated MSE of the optimal subsampling estimator under different subsample sizes for B-spline SIM for Dataset 3 using $\tilde{\boldsymbol{\pi}}$ in (3.3) and $\boldsymbol{\pi}$ in (3.1) with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

r	$\tilde{\boldsymbol{\pi}}$			$\boldsymbol{\pi}$		
	MSE_opt	Bias_opt	MSE_ratio	MSE_opt	Bias_opt	MSE_ratio
100(.1%n)	1.1369	0.4036	0.9703	0.9839	0.2864	0.7982
300(.3%n)	0.7980	0.2128	0.7967	0.6991	0.2215	0.6965
500(.5%n)	0.5631	0.1541	0.5622	0.6549	0.1987	0.6558
1000(1%n)	0.4542	0.1413	0.4600	0.5788	0.2009	0.5916
3000(3%n)	0.3491	0.1367	0.3782	0.5450	0.3277	0.5859
5000(5%n)	0.3373	0.1217	0.3865	0.5316	0.3895	0.6001

Table 4.27.: Simulated MSE of the optimal subsampling estimator under different subsample sizes for B-spline SIM for Dataset 4 using $\tilde{\boldsymbol{\pi}}$ in (3.3) and $\boldsymbol{\pi}$ in (3.1) with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

	$\tilde{\boldsymbol{\pi}}$			$\boldsymbol{\pi}$		
r	MSE_opt	Bias_opt	MSE_ratio	MSE_opt	Bias_opt	MSE_ratio
100(.1%n)	0.0006863387	1.6776e-06	452.3601813	0.0045871184	1.47937e-05	2877.2012047
300(.3%n)	8.014e-07	1.53e-08	0.9617051	1.4436e-06	2.99000e-08	1.8296614
500(.5%n)	4.995e-07	2.04e-08	0.7413186	9.765e-07	3.13000e-08	1.3582108
1000(1%n)	2.735e-07	2.06e-08	0.5003661	5.164e-07	3.08000e-08	0.8381556
3000(3%n)	1.060e-07	1.95e-08	0.254891	2.189e-07	2.85000e-08	0.5781220
5000(5%n)	7.04e-08	1.80e-08	0.2166383	1.508e-07	2.49000e-08	0.4937239

Table 4.28.: Simulated MSE of the optimal subsampling estimator under different subsample sizes for P-spline SIM for Dataset 1 using $\tilde{\boldsymbol{\pi}}$ in (3.3) and $\boldsymbol{\pi}$ in (3.1) with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

	$\tilde{\boldsymbol{\pi}}$			$\boldsymbol{\pi}$		
r	MSE_opt	Bias_opt	MSE_ratio	MSE_opt	Bias_opt	MSE_ratio
100(.1%n)	0.5601926	0.1046895	2.1646196	0.1007609	0.0033845	0.3294693
300(.3%n)	0.1042635	0.0084708	8.6377527	0.0502220	0.0009084	9.1719377
500(.5%n)	0.0137123	0.0012988	3.5356218	0.0037479	0.0004925	0.7378100
1000(1%n)	0.0010950	0.0010885	0.2771201	0.0020808	0.0005546	0.4995552
3000(3%n)	0.0011112	0.0011090	0.2677879	0.0003653	0.0003072	0.0899766
5000(5%n)	0.0010191	0.0010178	0.2849246	0.0002286	0.0001533	0.0601879

Table 4.29.: Simulated MSE of the optimal subsampling estimator under different subsample sizes for P-spline SIM for Dataset 2 using $\tilde{\boldsymbol{\pi}}$ in (3.3) and $\boldsymbol{\pi}$ in (3.1) with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

	$\tilde{\boldsymbol{\pi}}$			$\boldsymbol{\pi}$		
r	MSE_opt	Bias_opt	MSE_ratio	MSE_opt	Bias_opt	MSE_ratio
100(.1%n)	1.28833	0.44064	1.00072	1.38630	0.74675	1.01221
300(.3%n)	0.77724	0.20310	0.80889	0.88937	0.48446	0.87014
500(.5%n)	0.66962	0.22648	0.84528	0.69912	0.35328	0.84293
1000(1%n)	0.49564	0.20184	0.79203	0.51049	0.24038	0.84410
3000(3%n)	0.22898	0.07403	0.52293	0.42037	0.20868	0.92609
5000(5%n)	0.15672	0.05543	0.40462	0.37395	0.19059	0.93919

Table 4.30.: Simulated MSE of the optimal subsampling estimator under different subsample sizes for P-spline SIM for Dataset 3 using $\tilde{\boldsymbol{\pi}}$ in (3.3) and $\boldsymbol{\pi}$ in (3.1) with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

	$\tilde{\boldsymbol{\pi}}$			$\boldsymbol{\pi}$		
r	MSE_opt	Bias_opt	MSE_ratio	MSE_opt	Bias_opt	MSE_ratio
100(.1%n)	0.93231	0.42049	0.51767	1.69193	1.46669	0.92456
300(.3%n)	0.64008	0.41384	0.35096	1.65331	1.42813	0.91700
500(.5%n)	0.62072	0.42452	0.34058	1.70029	1.41116	0.93601
1000(1%n)	0.56786	0.43791	0.30317	1.71437	1.36026	0.92161
3000(3%n)	0.54606	0.41962	0.29367	1.73730	1.27693	0.91883
5000(5%n)	0.53593	0.41756	0.27591	1.74534	1.19295	0.89504

Table 4.31.: Simulated MSE of the optimal subsampling estimator under different subsample sizes for P-spline SIM for Dataset 4 using $\tilde{\boldsymbol{\pi}}$ in (3.3) and $\boldsymbol{\pi}$ in (3.1) with $n = 100,000$, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

	$\tilde{\boldsymbol{\pi}}$			$\boldsymbol{\pi}$		
r	MSE_opt	Bias_opt	MSE_ratio	MSE_opt	Bias_opt	MSE_ratio
100(.1%n)	0.0021188076	8.7152e-06	1087.6187418199	0.0002355085	5.323e-07	117.9397221
300(.3%n)	2.1381e-06	2.15e-08	1.8254728788	2.0622e-06	2.570e-08	1.7535417
500(.5%n)	1.0284e-06	2.56e-08	0.9091832430	1. 2973e-06	2.790e-08	1.1869519
1000(1%n)	6.347e-07	3.41e-08	0.7029792220	6. 871e-07	3.100e-08	0.7407815
3000(3%n)	2.739e-07	4.77e-08	0.4156326534	2. 909e-07	3.830e-08	0.4645388
5000(5%n)	1.771e-07	4. 55e-08	0.3287335498	2.115e-07	4.730e-08	0.3919317

5. REAL DATA APPLICATIONS

5.1 Video transcoding

Video content is being produced, transported and consumed in more ways and devices than ever. Meanwhile a seamless interaction is required between video content producing, transporting and consuming devices. The difference in device resources, network bandwidth and video representation types results in the necessary requirements for a mechanism for video content adoption. One such mechanism is called video transcoding. Video transcoding is a process of converting one compressed video representation to another. The basic idea of video transcoding is to convert unsupported video formats into supported ones. Unsupported videos include videos that are not playable by a given device due to lack of format support or those that require relatively higher system resources than the device can offer. Currently, transcoding is being utilized for such purposes as: bit-rate reduction in order to meet network bandwidth availability, resolution reduction for display size adoption, temporal transcoding for frame rate reduction and error resilience transcoding for insuring high quality of service (QoS).

Runtime scheduling of transcoding jobs in multicore and cloud environments is hard as their resource requirements may not be known before hand, thus the prediction of the transcoding time based on the input and output video features is in demand.

Let's consider the Youtube video transcoding time dataset from the UCI machine learning repository(https://archive.ics.uci.edu/ml/datasets.php), it has n=67,875 observations, and features including bitrate, framerate, resolution, codec, number of i frames, number of p frames, number of b frames, size of i frames, size of p frames, size of b frames of the input video and the desired bitrate, framerate, resolution and codec of the output video, which are treated as the predictors $X_1, X_2, ..., X_{19}$ ($p = 19$), the response variable is the total transcoding time.

We fit the data with the B-spline and P-spline single index models. The results are as follows. We can see that the A-optimal subsampling method outperformed the uniform subsampling method under both model settings and all listed subsample sizes. The MSE ratios are consistently smaller than 1.

Table 5.1.: MSE of the optimal subsampling estimator and the uniform subsampling estimator and their ratios under different subsample sizes for B-spline SIM for the video transcoding dataset, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$

r	MSE_opt	MSE_unif	MSE_ratio	Bias_opt	Bias_unif	Bias_ratio
200(.3%n)	1.19111	1.79193	0.66471	0.43577	0.82717	0.52682
350(.5%n)	1.07367	1.78433	0.60172	0.39044	0.81281	0.48036
680(1%n)	0.69675	1.66869	0.41755	0.31616	0.70462	0.44870
2000(3%n)	0.37747	1.72069	0.21937	0.32746	0.74311	0.44066
3400(5%n)	0.35149	1.64874	0.21319	0.33651	0.68150	0.49377
6800(10%n)	0.34296	1.55243	0.22092	0.33650	0.60342	0.55765
20000(30%n)	0.34001	1.22424	0.27773	0.33906	0.37494	0.90432

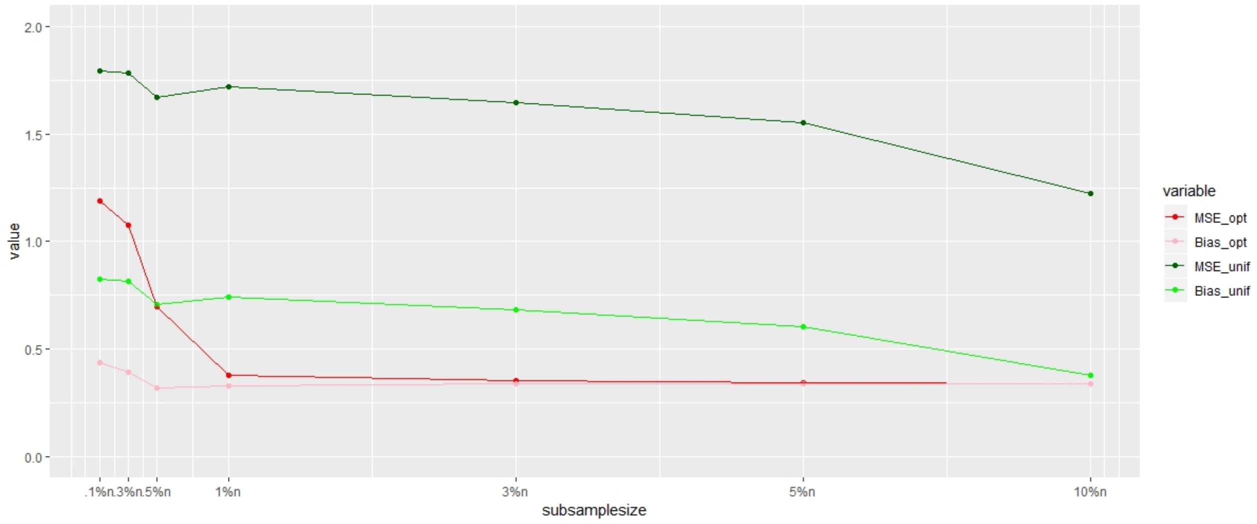

Fig. 5.1.: Plot of MSE and bias values from Table 5.1

Table 5.2.: The average time (in seconds) taken to calculate the subsampling estimator under each subsample size for B-spline SIM for video transcoding dataset (the full sample estimation takes 1029.09s, calculating $\boldsymbol{\pi}$ (first step) takes 46.51s)

r	Time_opt	Time_unif
200(.3%n)	14.62300	2.69424
350(.5%n)	4.09928	1.18566
680(1%n)	2.23788	1.07390
2000(3%n)	1.28136	1.74482
3400(5%n)	1.39596	2.54884
6800(10%n)	2.26600	5.17372
20000(30%n)	13.22626	16.17538

Table 5.3.: MSE of the optimal subsampling estimator and the uniform subsampling estimator and their ratios under different subsample sizes for P-spline SIM for video transcoding dataset, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$

r	MSE_opt	MSE_unif	MSE_ratio	Bias_opt	Bias_unif	Bias_ratio
200(.3%n)	1.52647	1.64479	0.92806	0.63165	0.70591	0.89480
350(.5%n)	1.39318	1.95854	0.71133	0.52222	0.97817	0.53387
680(1%n)	1.35133	1.75405	0.77041	0.47780	0.77783	0.61427
2000(3%n)	0.93755	1.69648	0.55265	0.22975	0.72307	0.31774
3400(5%n)	0.71443	1.69594	0.42126	0.13802	0.72093	0.19145
6800(10%n)	0.44914	1.46438	0.30671	0.06163	0.53695	0.11478
20000(30%n)	0.08919	1.48001	0.06026	0.01671	0.54788	0.03051

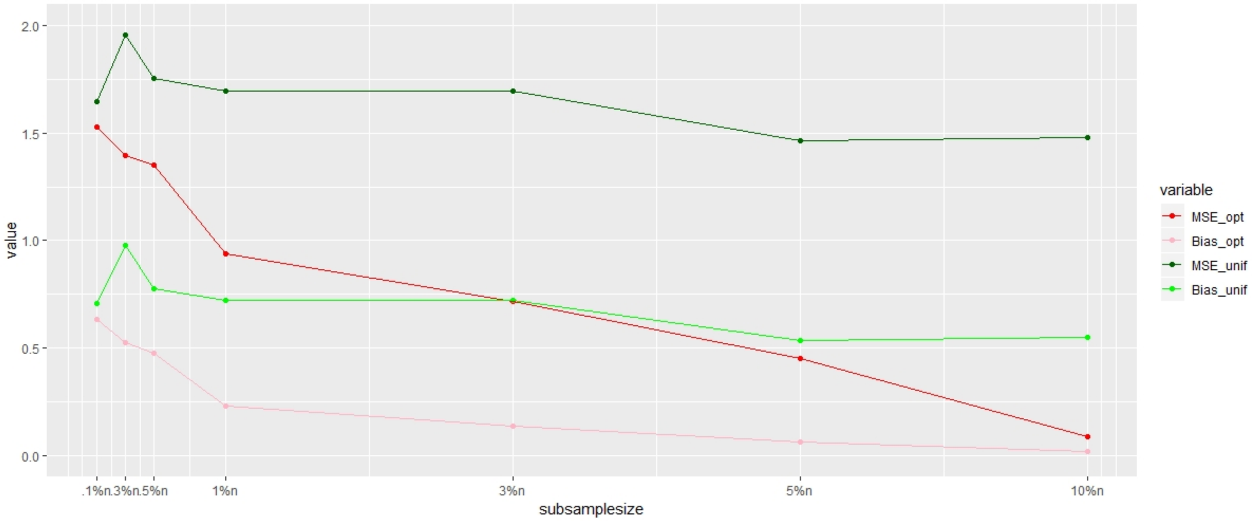

Fig. 5.2.: Plot of MSE and bias values from Table 5.3

Table 5.4.: The average time (in seconds) taken to calculate the subsampling estimator under each subsample size for P-spline SIM for the video transcoding dataset (the full sample estimation takes 129.75s, calculating $\boldsymbol{\pi}$ (first step) takes 11.18s)

r	Time_opt	Time_unif
200(.3%n)	0.19664	0.19864
350(.5%n)	0.22602	0.23694
680(1%n)	0.80252	1.01206
2000(3%n)	1.13454	1.57142
3400(5%n)	1.44596	2.21236
6800(10%n)	2.43032	4.16576
20000(30%n)	10.39942	18.06122

5.2 Online news popularity

As extensive amount of online news are available nowadays, social media companies want to know the popularity of news, which is indicated by the number of shares under each news from readers, before publication, thus comes the necessity of predicting the news popularity (number of shares). The possible factors that influence the new popularity are for example, number of words in the title and content respectively, number of videos, average length of the words in the content, the categories of channels (lifestyle, entertainment, social media, tech etc.), weekdays of the post, and so on. The dataset is from the UCI machine learning repository(https://archive.ics.uci.edu/ml/datasets.php) with n=39,644, p=58. We want to predict the number of shares using semiparametric single index models to show that the proposed subsampling method works better than the uniform subsampling method.

From Table 5.5, the MSE ratios are smaller than 1 for all the listed subsample sizes. However, the MSE for both the optimal subsampling and the uniform subsampling methods increase as the subsample size increases, which differs from the previous simulation and real data results. This could be caused by the increasing bias values, meaning that the B spline single index model may not be a good fit for this dataset. Even so, we want to display this result to show the optimal subsampling method improved on the subsampling estimation the uniform subsampling method.

This optimal subsampling method for the P spline single index modeling also shows desired results, see Table 5.7, Figure 5.4 and Table 5.8.

Table 5.5.: MSE of the optimal subsampling estimator and the uniform subsampling estimator and their ratios under different subsample sizes for the B-spline SIM on the online news popularity dataset, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

r	MSE_opt	MSE_unif	MSE_ratio	Bias_opt	Bias_unif	Bias_ratio
120(.3%n)	2.0474	2.0769	0.9858	1.1524	1.0808	1.0663
200(.5%n)	2.0693	2.2173	0.9333	1.2239	1.2439	0.9839
400(1%n)	2.1495	2.2710	0.9465	1.5217	1.3060	1.1651
1200(3%n)	2.2146	2.4423	0.9068	2.0115	1.5436	1.3031
2000(5%n)	2.2243	2.8201	0.7887	2.1343	2.1497	0.9928
4000(10%n)	2.2351	2.8494	0.7844	2.2181	2.2007	1.0079
12000(30%)	2.2390	2.8001	0.7996	2.2182	2.1626	1.0257

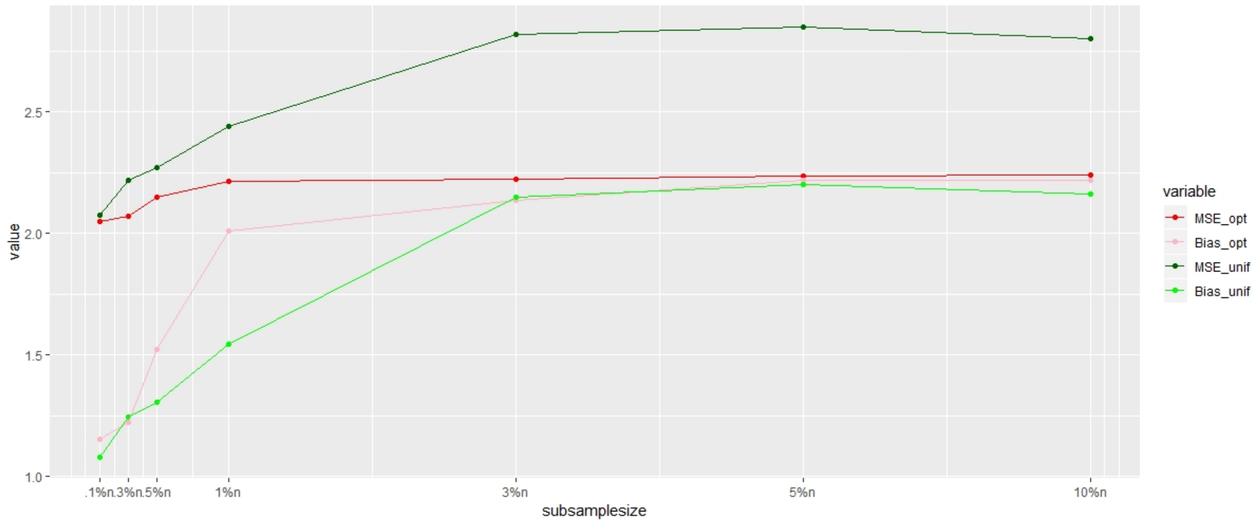

Fig. 5.3.: Plot of MSE and bias values from Table 5.5

Table 5.6.: The average time (in seconds) taken to calculate the subsampling estimator under each subsample size for the P-spline SIM for the online news popularity dataset (the full sample estimation takes 1152.13s, calculating $\boldsymbol{\pi}$ (first step) takes 92.85s)

r	Time_opt	Time_unif
120(.3%n)	85.0039	1.1724
200(.5%n)	80.4941	1.6149
400(1%n)	71.2637	1.8811
1200(3%n)	74.3716	3.4780
2000(5%n)	73.3290	5.8528
4000(10%n)	113.2929	11.8249
12000(30%)	306.1203	30.0621

Table 5.7.: MSE of the optimal subsampling estimator and the uniform subsampling estimator and their ratios under different subsample sizes for the P-spline SIM for the online news popularity dataset, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

r	MSE_opt	MSE_unif	MSE_ratio	Bias_opt	Bias_unif	Bias_ratio
120(.3%n)	1.7946	1.9430	0.9237	0.8077	0.9508	0.8495
200(.5%n)	1.4416	1.6874	0.8544	0.5231	0.7132	0.7334
400(1%n)	1.2323	1.5912	0.7745	0.3894	0.6334	0.6148
1200(3%n)	1.1023	1.1700	0.9422	0.3087	0.3424	0.9018
2000(5%n)	0.6810	0.8103	0.8404	0.3137	0.2355	1.3321
4000(10%n)	0.4799	0.6215	0.7722	0.3159	0.0968	3.2641
12000(30%)	0.5357	0.6025	0.8891	0.2960	0.1135	2.6066

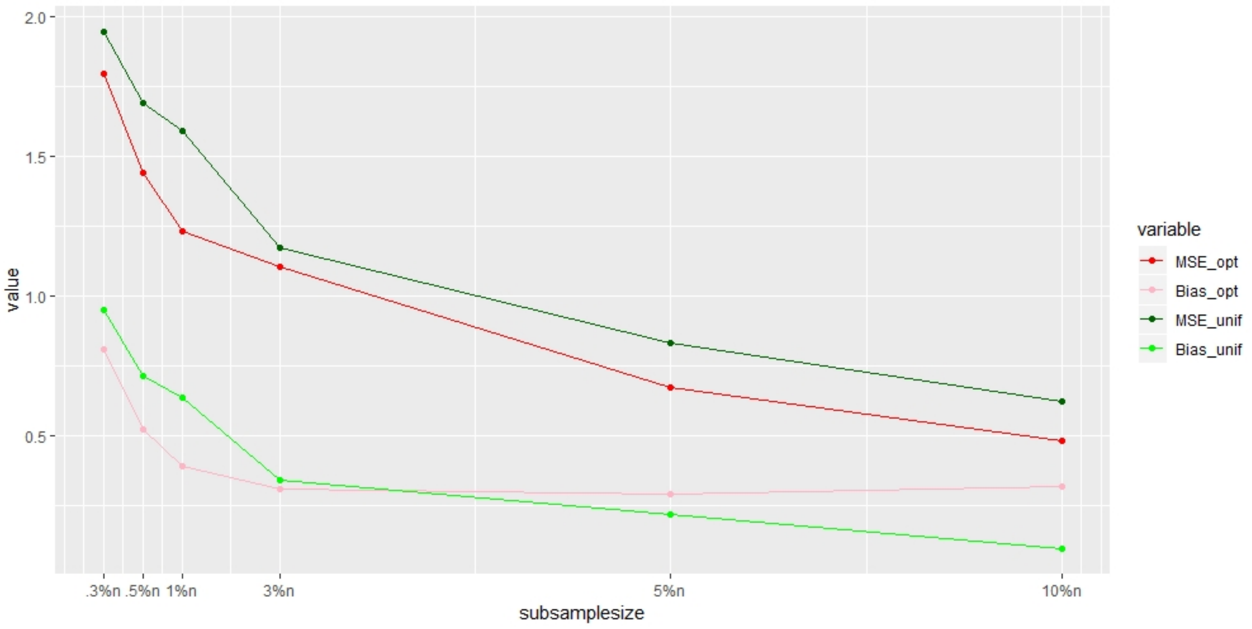

Fig. 5.4.: Plot of MSE and bias values from Table 5.7

Table 5.8.: The average time (in seconds) taken to calculate the subsampling estimator under each subsample size for the P-spline SIM for the online news popularity dataset (the full sample estimation takes 244.57s, calculating $\boldsymbol{\pi}$ (first step) takes 68.97s)

r	Time_opt	Time_unif
120(.3%n)	1.1941	1.1841
200(.5%n)	1.5109	1.5052
400(1%n)	2.3215	2.3095
1200(3%n)	5.6565	5.6099
2000(5%n)	9.3056	9.3634
4000(10%n)	18.5010	18.5362
12000(30%)	22.4944	22.5102

5.3 Gas sensor

In this section, we apply the subsampling method to the gas sensor array dataset from chemistry (https://archive.ics.uci.edu/ml/datasets.php). This dataset was collected by exposing 16 chemical sensors to a gas mixture of Ethylene and CO in air at varying concentration levels. For each gas mixture, the signals were recorded from the sensors. We excluding all the negative readings from each sensors and drop the first 20, 000 data points which correspond to the system run-in time. After these, there are totally $n = 1,605,003$ observations. The objective is to predict the concentration of enthylene with the 16 sensors readings as covariates. Note that the sensor reading are rescaled with factor 0.001. Due to the memory limitation of desktop computers, we used the super computer (Big Red II at Indiana University) to handle the full data estimation of the semiparametric single index model fittings, then compare the optimal subsampling method with the uniform subsampling method by applying the A-optimal Scoring Algorithm 3. The first step subsample size $r_0 = 800(.05\%n)$, second step size r ranges from $160(.01\%n)$ to $800(.05\%n)$(Due to the memory storage issue of the big data, we only take small subsample sizes). The results are reported in Table 5.9, Figure 5.9 and Table 5.10 for B spline SIM, and Table 5.11, Figure 5.11 and Table 5.12 for P spline SIM. For both models, MSE_{ratio}'s are consistently less than 1, showing the better performance of the optimal subsampling method over the uniform subsampling method.

Table 5.9.: MSE of the optimal subsampling estimator and the uniform subsampling estimator and their ratios under different subsample sizes for the B-spline SIM on the gas sensor dataset, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$

r	MSE_opt	MSE_unif	MSE_ratio	Bias_opt	Bias_unif	Bias_ratio
160(.01%n)	1.3914	1.4606	0.9526	0.4874	0.5690	0.8566
320(.02%n)	0.7157	0.8887	0.8054	0.1425	0.2269	0.6281
480(.03%n)	0.4875	0.6391	0.7628	0.0935	0.1213	0.7707
640(.04%n)	0.2548	0.2971	0.8576	0.0840	0.0518	1.6213
800(.05%n)	0.1640	0.2058	0.7968	0.1159	0.0932	1.2439

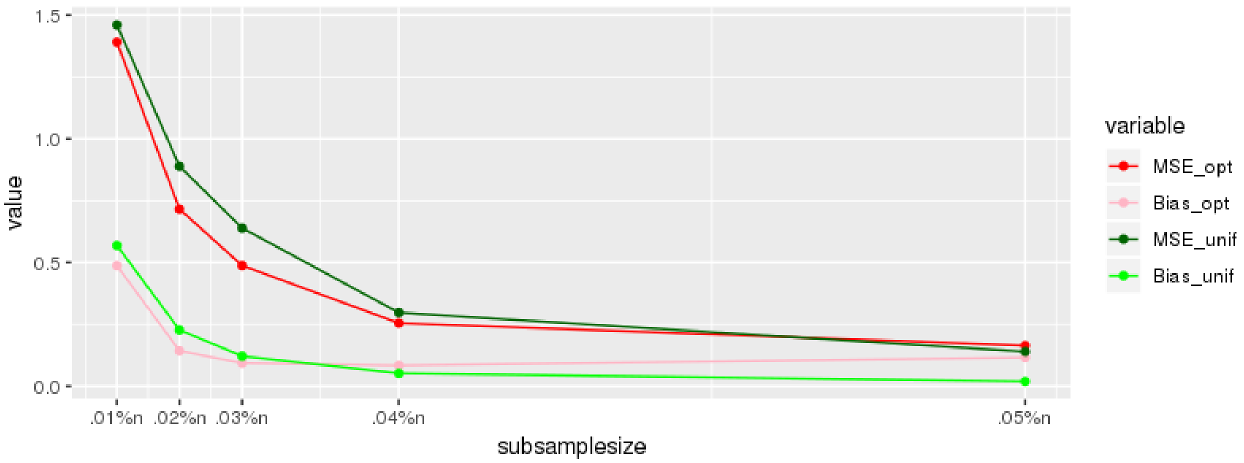

Fig. 5.5.: Plot of MSE and bias values from Table 5.9

Table 5.10.: The average time (in seconds) taken to calculate the subsampling estimator under each subsample size for the B-spline SIM on the gas sensor dataset (the full sample estimation takes 128798s, calculating $\boldsymbol{\pi}$ (first step) takes 645.614s)

r	Time_opt	Time_unif
.01%n	98.8942	77.4167
.02%n	148.4920	134.1387
.03%n	212.0116	166.5673
.04%n	351.6188	294.6663
.05%n	927.4393	697.4677

Table 5.11.: MSE of the optimal subsampling estimator and the uniform subsampling estimator and their ratios under different subsample sizes for the B-spline SIM on the gas sensor dataset, $MSE_{ratio} = \frac{MSE_{opt}}{MSE_{unif}}$.

r	MSE_opt	MSE_unif	MSE_ratio	Bias_opt	Bias_unif	Bias_ratio
.01%n	2.0226	2.2017	0.9186	1.0252	1.3364	0.7671
.02%n	2.0918	2.5296	0.8269	1.1012	1.6506	0.6672
.03%n	2.0877	2.6886	0.7765	1.1011	1.8544	0.5938
.04%n	2.0646	2.7964	0.7383	1.0734	2.0118	0.5335
.05%n	2.1808	2.6073	0.8364	1.2003	1.7441	0.6882

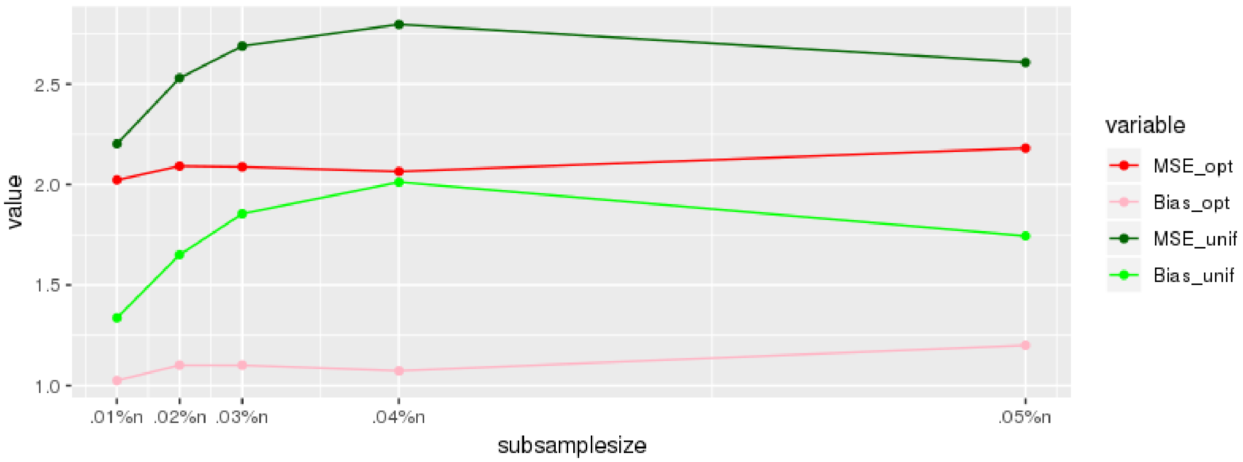

Fig. 5.6.: Plot of MSE and bias values from Table 5.11

Table 5.12.: The average time (in seconds) taken to calculate the subsampling estimator under each subsample size for the B-spline SIM on the gas sensor dataset (the full sample estimation takes 107523.9s, calculating $\boldsymbol{\pi}$ (first step) takes 319.458s)

r	Time_opt	Time_unif
.01%n	64.1687	35.9975
.02%n	170.3558	133.9499
.03%n	263.0385	201.5794
.04%n	538.8632	364.5358
.05%n	1673.112	1024.964

REFERENCES

REFERENCES

[1] Antoniadis, A., Gregoire G., and McKeague, I. W. (2004). Bayesian estimation in single-index models. Statist. Sinica, 14(4), 1147-1164.

[2] Bickel, P. J. and Freedman, D. A. (1983). Bootstrapping regression models with many parameters. In A Festschrift for Erich L. Lehmann (P. J. Bickel, K. A. Doksum and J. L. Hodges, Jr., eds.) 28-48. Wadsworth, Belmont, CA.

[3] Dennis, J.E. Jr. and Schnabel, R.B. Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[4] Djalil, C., Didier, C. (2007). On the strong consistency of asymptotic M Estimators. J. Statist. Plan. Infer. 137:2774-2783.

[5] 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.

[6] Drineas P., Mahoney M.W. and Muthukrishnan S. (2006b). Sampling algorithms for 2 regression and applications. Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1127-1136.

[7] Drineas, P., Mahoney, M., Muthukrishnan, S. Sarlos, T. (2011). Faster least squares approximation. Numerische Mathematik, 117: 219-249.

[8] Efron, B. (1979), Bootstrap methods: another look at the jackknife. Annals of Statistics, 7: 1-26.

[9] Eilers, P.H.C. and Marx, B.D. (1992). Generalized linear models with P-splines. In: Proceedings of GLIM 92 and 7th International Workshop on Statistical Modelling, Munich, Germany. Lecture Notes in Statistics, Vol. 78, Advances in GLIM and Statistical Modelling, Eds. L. Fahrmeir, B. Francis, R. Gilchrist, G. Tutz. SpringerVerlag, New York, 72-77.

[10] Eilers, P.H.C. and Marx, B.D. (1996). Flexible smoothing using B-splines and penalized likelihood (with Comments and Rejoinder). Statistical Science, 11(2): 89-121.

[11] Eilers, P.H.C, Marx, B.D. and Durban, Maria (2015). Twenty years of P-splines. Statistics and Operations Research Transactions, 39(2): 1-38.

[12] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American Statistical Association, 96(456): 1348-1360.

[13] Hardle, W., Hall, P. and Ichimura, H. (1993). Optimal smoothing in single-index models. Annals of Statistics, 21(1): 157-178.

[14] Hristache, M., Juditsky, A., Polzehl, J. and Spokoiny V. (2001). Structure adaptive approach for dimension reduction. Annals of Statistics, 29(6): 1537-1566.

[15] Hoerl, E. and Kennard, R. W. (1970). Ridge regression: biased estimation for nonorthogonal problems. Journal of the American Statistical Association, 12(1): 55-67.

[16] Ichimura, H. (1993). Semiparametric least squares (SLS) and weighted SLS estimation of single-index models. Journal of Ecomometrics, 58(1-2): 71-120.

[17] Mammen, E. (1989). Asymptotics with increasing dimension for robust regression with applications to the bootstrap. Annals of Statistics, 17: 382-400.

[18] Mammen, E. (1993). Bootstrap and wild bootstrap for high-dimensional linear models. Annals of Statistics, 21: 255-285.

[19] OSullivan, F. (1986). A Statistical Perspective on Ill-Posed Inverse Problems (with Discussion). Statistical Science, 1: 505-527.

[20] O'Sullivan, F. (1988). Fast computation of fully automated log-density and log-hazard estimators. SIAM Journal of Scientic and Statistical Computation, 9: 363-379.

[21] Ortega, J. M. and Rheinboldt, W. G. (1970). Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York.

[22] Peng, Hanxiang and Tan, Fei (2018). Big data linear regression via A-optimal subsampling. Submitted to the Annals of Statistics, 2018.

[23] Ruppert, D. (2002). Selecting the number of knots for penalized splines. Journal of the American Statistical Association, 11(4): 735-757.

[24] Ruppert, D., and Carroll, R. (1997), Penalized Regression Splines. working paper, Cornell University, School of Operations Research and Industrial Engineering. (available at www.orie.cornell.edu/ davidr/papers).

[25] Ruppert, D., and Carroll, R. (2000). Spatially adaptive penalties for spline fitting. Australian and New Zealand Journal of Statistics, 42: 205-223.

[26] Ruppert D., Wand M., Carroll R. Semiparametric Regression. New York: Cambridge University Press, 2003.

[27] Sharif S. and kamal S. (2018). Comparison of significant approaches of penalized spline regression (P-splines). Pakistan Journal of Statistics and Operation Research, 14(2): 310-315.

[28] Stoker, T. M. (1986). Consistent estimation of scaled coefficients. Econometrica 54: 1461-1481.

[29] Wang, G. and Wang, L. (2015). Spline estimation and variable selection for single index prediction models with diverging number of index parameters. Journal of Statistical Planning and Inference, 162: 119.

[30] Wang, H., Zhu, R. and Ma, P. (2018). Optimal subsampling for large sample logistic regression. Journal of the American Statistical Association, 113(522): 829-844.

[31] Wu, J., and Tu, W. (2016). A multivariate single-index model for longitudinal data. Statistical Modelling, 16(5): 392-408.

[32] Xia, Y. C. and Hardle, W. (2006). Semi-parametric estimation of partially linear single-index models. Journal of Multivariate Analysis. 97: 1162-1184.

[33] Xia, Y. C., Tong, H., LI, W. K. and Zhu, L. (2002). An adaptive estimation of dimension reduction space (with discussion). Journal of the Royal Statistical Society. Series B, Statistical methodology, 64: 363-410.

[34] Yu, Y. and Ruppert, D. (2002). Penalized spline estimation for partially linear single-index models. Journal of the American Statistical Association, 97(460): 1042-1054.

[35] Yurinskii, V. (1976). Exponential Inequalities for Sums of Random Vectors. Journal of Multivariate Analysis, 6: 473-499.

[36] Yu, Y., Wu, C., and Zhang, Y. (2017). Penalised spline estimation for generalised partially linear single-index models. Statistics and Computing, 27: 571-582.

[37] Y. le Cun, Modeles Connexionnistes de l'Apprentissage. PhD thesis, Universite Pierre et Marie Curie, Paris, France, 1987.

[38] Zhu, R., Ma, P., Mahoney, M.W. and Yu, B. (2015). Optimal sub-sampling approaches for large sample linear regression. arXiv: 1509.0511. v1[stat.ME].