Electronic Journal of Statistics
Vol. 18 (2024) 674–705
ISSN: 1935-7524
https://doi.org/10.1214/24-EJS2223

Renewable Huber estimation method for streaming datasets

Rong Jiang*, Lei Liang, and Keming Yu†

Shanghai Polytechnic University, Donghua University, Anqing Normal University and Brunel University London
e-mail: jiangrong@sspu.edu.cn; 15779057317@163.com; keming.yu@brunel.ac.uk

Abstract: Streaming data refers to a data collection scheme where observations arrive sequentially and perpetually over time, making it challenging to fit into computer memory for statistical analysis. The ordinary least squares estimate for linear regression is sensitive to heavy-tailed errors and outliers, which are commonly encountered in applications. In this case, the Huber loss function is a useful criterion for robust regression. In this paper, we propose robust regression estimation and variable selection for streaming datasets. Unlike the renewable estimation generalized linear regression for streaming datasets, however, the Huber loss function is only first-order differentiable, which poses challenges to renewable estimation in both computation and theoretical development. To address the challenge, we introduce a new smoothed version of the Huber first derivative, which admits a fast and scalable algorithm to perform optimization for streaming data sets and achieves the best fitting of Huber function among different versions. Theoretically, the proposed statistics are shown to have the same asymptotic properties as the standard version computed on an entire data stream with the data batches pooled into one data set, without additional condition. The proposed methods are illustrated using current data and the summary statistics of historical data. Both simulations and real data analysis are conducted to illustrate the finite sample performance of the proposed methods.

MSC2020 subject classifications: Primary 60G08; secondary 62G20.
Keywords and phrases: Huber loss, streaming data, online updating, high-dimensional estimation.

Received February 2023.

Contents

*The Ministry of Education of the People's Republic of China, Humanities and Social Science Foundation (No. 22YJC910005).
†The National Social Science Foundation of China (No. 21BTJ040).

1. Introduction

Our era has witnessed the massive explosion of data and a dramatic improvement of technology in collecting and processing big data. Due to the explosive growth of data from non-traditional sources such as mobile phones, social networks, and e-commerce, streaming data is becoming a core component of big data analysis. As streaming data grows rapidly in volume and velocity, storing and combing data becomes increasingly challenging. To reduce the demand on computing memory and achieve real-time processing, the nature of streaming data calls for the development of algorithms that require only "one pass" over the data.

In big data streams, data arrives as $\{D_1, \ldots, D_b\}$ up to the b-th batch, where D_j is the j-th batch data set with a sample size of n_j. Then, the total sample size is $N_b = \sum_{j=1}^{b} n_j$. The data can exceed even a supercomputer's memory when the number of blocks b is large enough. The primary goal of processing such streaming data is to sequentially update some statistics of interest upon the arrival of a new data batch, in the hope of not only freeing up space for the storage of massive historical individual-level data, but also providing real-time inference and decision-making. Stochastic gradient descent (SGD) algorithm (Robbins and Monro, 1951) and streaming stochastic variance reduced gradient (Streaming SVRG) algorithm (Frostig et al., 2015) in the field of machine learning can quickly updates of parameter estimates along with sequentially arriving data. However, they are not useful for statistical inference because only part of the information matrix (i.e. the Hessian's diagonal elements) is recorded and updated over iterations (Luo and Song, 2020). Statisticians have proposed several cumulative update methods specifically for the sequential update of regression coefficient estimators. For example, Schifano et al. (2016) developed online-updating algorithms for linear models and estimating equations. However, the estimation consistency of the methods in Schifano et al. (2016) has been established based on a strong regularity condition: the total number of streaming data sets b needs to satisfy the order of $b = O(n_j^c)$ with $c < 1/3$ for

all js. This is a very strong restriction. For example, the estimation consistency may not be guaranteed in the situation where streaming data sets arrive perpetually with $b \to \infty$. Luo and Song (2020) proposed renewable estimation for the generalized linear model, which overcame the above unnatural restriction. Other references can see Chen, Liu and Zhang (2019); Jiang and Yu (2022); Luo, Zhou and Song (2022); Yang and Yao (2022) and Quan and Lin (2022).

Big data is easily contaminated by outliers and often suffer heavy-tailed errors. This has a significant impact on many statistical inference processes. Robust procedures in statistical inference aim to produce reliable estimates that are not seriously affected by outliers or small deviations from model assumptions. So it has great interest in practice. Huber's M-estimator (Huber, 1973) is one of the most widely used robust alternatives to the least square estimation (LSE). Actually, Huber type of regression has recently received considerable interest in dealing with data outliers, see Fan, Li and Wang (2017); Zhou et al. (2018); Sun, Zhou and Fan (2020) and among others, but these methods don't deal with streaming datasets. Consider the following linear regression model:

$$\mathbf{Y} = \mathbf{X}^\top \boldsymbol{\beta}_0 + \boldsymbol{\varepsilon}, \tag{1.1}$$

where $\mathbf{X}$ is a random vector of p-dimensional covariates, $\boldsymbol{\beta}_0$ is a vector of unknown parameters of interest, and $\boldsymbol{\varepsilon}$ is an independent regression error with zero mean and finite variance σ^2. The distribution function $F(\cdot)$ of $\boldsymbol{\varepsilon}$ is symmetric, as is customary in classical robust statistics to ensure consistency of regression M-estimators. The above settings of the model (1.1) are common for Huber regression, see Yohai and Maronna (1979); Loh (2017); Jiang et al. (2019); Zheng (2021); Loh (2021); Han et al. (2022a) and so on. Suppose that the batches up to the b-th batch of streaming data can be pooled into one data set. We denote $N = N_b$ and let $\{\mathbf{X}_i, Y_i\}_{i=1}^N$ be an i.i.d. sample from $(\mathbf{X}, \mathbf{Y})$ in the model (1.1). Given some $\tau > 0$, referred to as the robustification parameter, Huber's regression M-estimator for estimating $\boldsymbol{\beta}_0$ is defined as

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \ell_\tau\big(Y_i - \mathbf{X}_i^\top \boldsymbol{\beta}\big), \tag{1.2}$$

where the Huber loss $\ell_\tau(\cdot)$ is defined as

$$\ell_\tau(r) = \begin{cases} r^2/2, & \text{if } |r| \le \tau, \\ \tau|r| - \tau^2/2, & \text{if } |r| > \tau. \end{cases} \tag{1.3}$$

The shape parameter τ is chosen to be 1.345σ in order to achieve 95% asymptotic relative efficiency for normally distributed data, see Western (1995); Huber and Ronchetti (2009) and Lambert-Lacroix and Zwald (2011).

Note that the Huber loss function (1.3) is only first-order differentiable, so existing procedures explored in the state of the art on the topic such as Luo and Song (2020) does not work. Because their method requires its loss function to be twice continuously differentiable in order to perform a Taylor expansion of the estimating equation. While some newly developed alternative versions

(Jiang et al., 2019; Yu, 2020) of Huber loss functions are not twice continuously differentiable either, we adopt a smoothing technique to smooth the first derive of the ordinary Huber loss function into a twice continuously differentiable loss function, which helps to produce a renewable estimator for Huber regression. Chen (2007) and Hampel, Hennig and Ronchetti (2011) have also developed smoothing algorithms for Huber regression, but they have no theoretical results, so the asymptotic effect cannot be verified. Our proposed smoothing method is different from theirs, and the large sample properties of the method are given.

Furthermore, many data streams are high-dimensional, such as genomic data used to explain variation in biological phenotypes and their genetic profiles. One of the major challenges in high-dimensional data analysis is deciding which of the many potential forecasters to include in the model. Variable selection methods have been successfully applied to high-dimensional regression problems, such as the least absolute shrinkage and selection operator (LASSO) (Tibshirani, 1996), smoothly clipped absolute deviation (SCAD) (Fan and Li, 2001), adaptive LASSO (Zou, 2006) and so on. Several algorithms were developed for analyzing high-dimensional streaming datasets. Fan et al. (2018a) applied the truncated stochastic gradient descent to a linear model. Shi et al. (2021) proposed an inference procedure for high-dimensional linear models via recursive online-score estimation. Han et al. (2021) studied an online debiased LASSO method for high-dimensional linear models with streaming datasets based on the least squares method. Luo et al. (2021) and Ma, Lin and Gai (2023) investigated online updating variable selection in a generalized linear model with streaming datasets. Deshpande, Javanmard and Mehrabi (2023) considered a class of online estimators in a high-dimensional autoregressive model. This paper also studies a renewable variable selection method of Huber regression.

In summary, we develop renewable Huber estimation and variable selection for high-dimensional linear regression models. Our statistical contributions include: (i) the renewable Huber estimation and variable selection are real-time estimations, which require only the availability of the current data batch in the data stream and sufficient statistics on the historical data at each stage of the analysis, and inference procedure that is highly scalable with respect to rapidly growing data volumes; and (ii) the asymptotic properties of the proposed renewable Huber estimators under the conditions similar to those in an offline setting and no restrictions on n_j and b, which means that the new methods are adaptive to the situation where streaming data sets arrive fast and perpetually.

The remainder of this paper is organized as follows. In Section 2, the renewable smoothing Huber estimation is proposed. The renewable high-dimensional estimation method is developed in Section 3. Both simulation examples and the application on real data are given in Section 4 to illustrate the proposed procedures. We conclude this paper with a brief conclusion in Section 5. All technical proofs are provided in Appendix A.

2. Renewable parameter estimation

2.1. Smoothing the first derivative of the Huber loss function

Note that the first and second derivative functions of the Huber loss function (1.3) are

$$\ell'_\tau(r) = \begin{cases} r, & \text{if } |r| \le \tau, \\ \tau \operatorname{sign}(r), & \text{if } |r| > \tau, \end{cases}$$

and

$$\ell''_\tau(r) = \begin{cases} 1, & \text{if } |r| \le \tau, \\ 0, & \text{if } |r| > \tau, \end{cases}$$

where $\operatorname{sign}(\cdot)$ is a sign function. Since $\ell''_\tau(\cdot)$ is not a continuous function and the Huber estimator $\hat{\boldsymbol{\beta}}$ (1.2) does not display an expression, so it is impossible to construct a renewable estimator for streaming data sets based on the Huber loss function (1.3). In order to construct a renewable estimator for streaming data, we first smooth the derivative of the Huber loss function as

$$\tilde{\ell}'_{\tau,h}(r) = \begin{cases} -\tau, & \text{if } r < -\tau - h, \\ A(r,h,\tau), & \text{if } -\tau - h \le r \le -\tau + h, \\ r, & \text{if } -\tau + h < r < \tau - h, \\ -A(-r,h,\tau), & \text{if } \tau - h \le r \le \tau + h, \\ \tau, & \text{if } r > \tau + h, \end{cases}$$

where $A(r,h,\tau)$ is a smooth function and h is the bandwidth. If $A(r,h,\tau)$ satisfies condition **C1** in Section 2.3, for any $r \in \mathrm{R}$, we have

$$\left|\ell'_\tau(r) - \tilde{\ell}'_{\tau,h}(r)\right| \le h,$$

and the derivative of $\tilde{\ell}'_{\tau,h}(r)$,

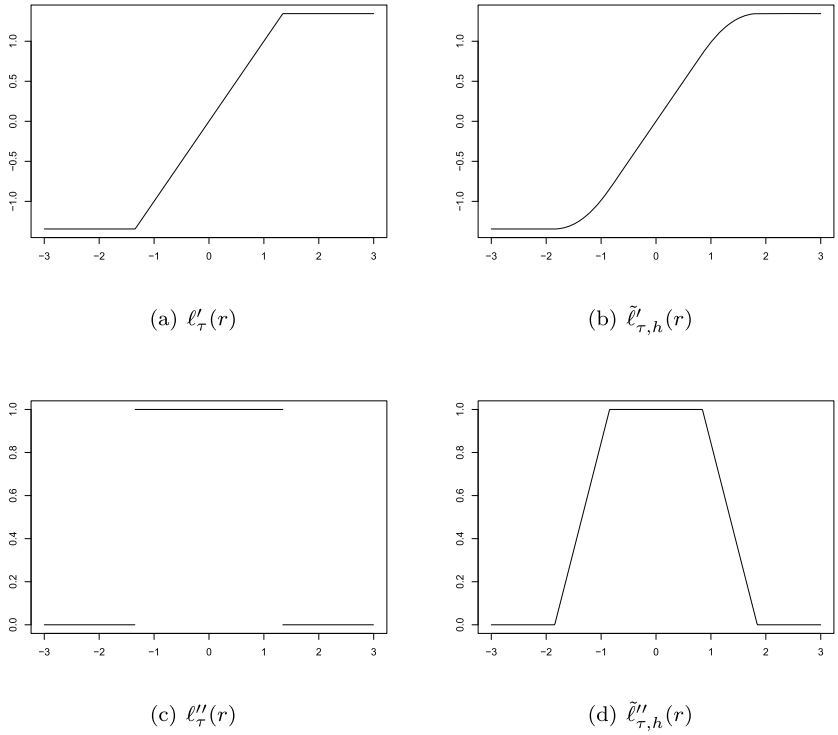
$$\tilde{\ell}''_{\tau,h}(r) = \begin{cases} A'(r,h,\tau), & \text{if } -\tau - h \le r \le -\tau + h, \\ 1, & \text{if } -\tau + h < r < \tau - h, \\ A'(-r,h,\tau), & \text{if } \tau - h \le r \le \tau + h, \\ 0, & \text{if } |r| > \tau + h, \end{cases}$$

is a continuous bounded function, where $A'(r,h,\tau)$ is the derivative of $A(r,h,\tau)$. For instance, we can take

$$A(r,h,\tau) = \frac{h}{4} - \tau + \frac{1}{2}(r+\tau) + \frac{1}{4h}(r+\tau)^2, \tag{2.1}$$

which satisfies condition **C1**, and $\tilde{\ell}''_{\tau,h}(r)$ is a continuous bounded function (see Fig. 1) as

$$\tilde{\ell}''_{\tau,h}(r) = \begin{cases} \frac{1}{2} + \frac{1}{2h}(r+\tau), & \text{if } -\tau - h \le r \le -\tau + h, \\ 1, & \text{if } -\tau + h < r < \tau - h, \\ \frac{1}{2} - \frac{1}{2h}(r-\tau), & \text{if } \tau - h \le r \le \tau + h, \\ 0, & \text{if } |r| > \tau + h. \end{cases}$$



(a) $\ell'_\tau(r)$ (b) $\tilde{\ell}'_{\tau,h}(r)$

(c) $\ell''_\tau(r)$ (d) $\tilde{\ell}''_{\tau,h}(r)$

FIG 1. *The figures of* $\ell'_\tau(r)$, $\tilde{\ell}'_{\tau,h}(r)$, $\ell''_\tau(r)$ *and* $\tilde{\ell}''_{\tau,h}(r)$ *with* r *rang from* $[-3,3]$, $\tau = 1.345$ *and* $h = 0.5$.

By the definition of $\tilde{\ell}'_{\tau,h}(r)$ and $A(r,h,\tau)$ in (2.1), it maintains the robustness of the original Huber method because the value of $\tilde{\ell}'_{\tau,h}(r)$ is $\tau\,\text{sign}(r)$ if $|r| > \tau+h$ and $\tilde{\ell}'_{\tau,h}(r)$ is a bounded function if $|r| \le \tau + h$.

It should point out that the Pseudo-Huber loss function (Hartley and Zisserman, 2004) may also be used as a smooth approximation of the Huber loss function, which has derivatives of all degrees. From its definition

$$\bar{\ell}_\tau(r) = \tau^2\big(\sqrt{1 + r^2/\tau^2} - 1\big),$$

we have

$$\bar{\ell}'_\tau(r) = r/\sqrt{1 + r^2/\tau^2}.$$

After comparing $\ell'_\tau(r)$ and $\bar{\ell}'_\tau(r)$ in Fig. 2, we can find that there is a certain deviation between $\ell'_\tau(r)$ and $\bar{\ell}'_\tau(r)$, and only when r/τ is very large or small, $\ell'_\tau(r)$ and $\bar{\ell}'_\tau(r)$ are close. However, r often contains unknown parameters, so it is difficult to choose a proper τ. Therefore, the Pseudo-Huber loss function is not a good smoothing method for the Huber loss function, but our proposed smoothed version of Huber loss function method can be very close to $\ell'_\tau(r)$ as h decreases.

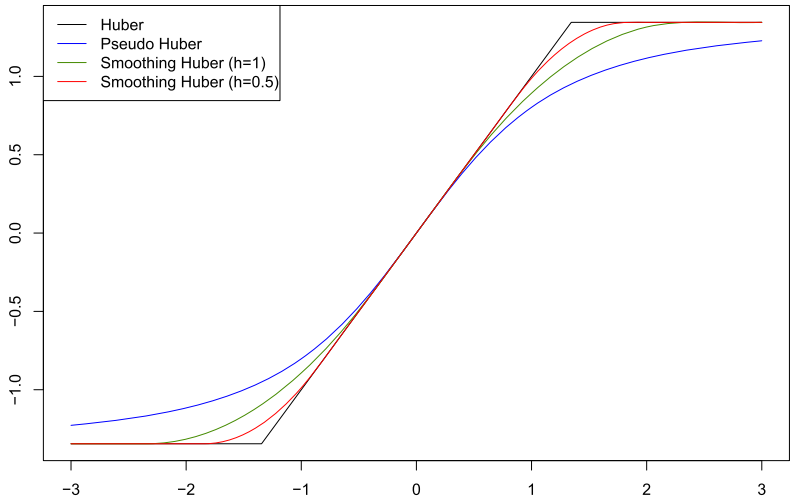


FIG 2. *The figures of* $\ell'_\tau(r)$, $\tilde{\ell}'_{\tau,h}(r)$ *and* $\bar{\ell}'_\tau(r)$ *with* r *range from* $[-3,3]$, $\tau = 1.345$, $h = 1$ *and* $h = 0.5$.

2.2. *Huber estimation for streaming data sets*

For model (1.1), $D_j = \{\mathbf{X}_j, \mathbf{Y}_j\}$ is the j-th batch data set, where $\mathbf{Y}_j = (Y_{1,j}, \ldots, Y_{n_j,j})^\top$ and $\mathbf{X}_j = (\mathbf{X}_{1,j}, \ldots, \mathbf{X}_{n_j,j})^\top$. We suppose that the $(\mathbf{X}_{i,j}, Y_{i,j})$ for all is and js are i.i.d. samples from $(\mathbf{X}, \mathbf{Y})$. We begin with a simple scenario of two batches of data D_1 and D_2, where D_2 arrives after D_1. We want to update the initial Huber estimator (HE) $\hat{\boldsymbol{\beta}}_1$ (or $\hat{\boldsymbol{\beta}}_1^*$) by (1.2) to a renewed HE $\hat{\boldsymbol{\beta}}_2^*$ without using any subject-level data but only some summary statistics from D_1. By (1.2), the HE $\hat{\boldsymbol{\beta}}_1$ satisfies,

$$\frac{1}{N_1}\mathbf{U}(D_1; \hat{\boldsymbol{\beta}}_1) = \mathbf{0}, \tag{2.2}$$

where $\mathbf{U}(D_j; \boldsymbol{\beta}) = \sum_{i \in D_j} \mathbf{X}_i \ell'_\tau(Y_i - \mathbf{X}_i^\top \boldsymbol{\beta})$ and $N_1 = n_1$ is the sample size of D_1. Then, $\hat{\boldsymbol{\beta}}_2^*$ satisfies the following aggregated score equation:

$$\frac{1}{N_2}\mathbf{U}\big(D_1; \hat{\boldsymbol{\beta}}_2^*\big) + \frac{1}{N_2}\mathbf{U}\big(D_2; \hat{\boldsymbol{\beta}}_2^*\big) = \mathbf{0}. \tag{2.3}$$

Solving equation (2.3) for $\hat{\boldsymbol{\beta}}_2^*$ actually involves the use of subject-level data in both D_1 and D_2, where D_1 may no longer to accessible. Our renewable estimation is able to handle this issue. To proceed, by the Lemma 1 in Appendix A, we can obtain

$$\mathbf{U}\big(D_1; \hat{\boldsymbol{\beta}}_2^*\big) = \tilde{\mathbf{U}}\big(D_1; \hat{\boldsymbol{\beta}}_2^*; h_1\big) + O_p(n_1 h_1 \sqrt{p} \log p), \tag{2.4}$$

where $\tilde{\mathbf{U}}(D_j;\boldsymbol{\beta};h)=\sum_{i\in D_j}\mathbf{X}_i\tilde{\ell}'_{\tau,h}(Y_i-\mathbf{X}_i^\top\boldsymbol{\beta})$. Because $\tilde{\ell}''_{\tau,h}(\cdot)$ exists, we can take the first-order Taylor series expansion of the $\tilde{\mathbf{U}}(D_1;\hat{\boldsymbol{\beta}}_2^*;h_1)$ around $\hat{\boldsymbol{\beta}}_1$ as

$$\tilde{\mathbf{U}}\big(D_1;\hat{\boldsymbol{\beta}}_2^*;h_1\big)=\tilde{\mathbf{U}}(D_1;\hat{\boldsymbol{\beta}}_1;h_1)+\mathbf{J}(D_1;\hat{\boldsymbol{\beta}}_1;h_1)\big(\hat{\boldsymbol{\beta}}_2^*-\hat{\boldsymbol{\beta}}_1\big)+O_p(\mathbf{R}_{n_1}),\quad(2.5)$$

where $\mathbf{R}_{n_1}=\|\hat{\boldsymbol{\beta}}_2^*-\hat{\boldsymbol{\beta}}_1\|_2(n_1h_1+\sqrt{n_1h_1\log p}+n_1\|\hat{\boldsymbol{\beta}}_2^*-\hat{\boldsymbol{\beta}}_1\|_2+\sqrt{n_1\log p}\|\hat{\boldsymbol{\beta}}_2^*-\hat{\boldsymbol{\beta}}_1\|_2^{1/2})$, $\mathbf{J}(D_j;\boldsymbol{\beta};h)=\partial\tilde{\mathbf{U}}(D_j;\boldsymbol{\beta};h)/\partial\boldsymbol{\beta}=-\sum_{i\in D_j}\mathbf{X}_i\mathbf{X}_i^\top\tilde{\ell}''_{\tau,h}(Y_i-\mathbf{X}_i^\top\boldsymbol{\beta})$ and $\|\cdot\|_2$ is the Euclidean norm. By (2.2), (2.4) and (2.5), we have

$$\mathbf{U}\big(D_1;\hat{\boldsymbol{\beta}}_2^*\big)=\mathbf{J}(D_1;\hat{\boldsymbol{\beta}}_1;h_1)\big(\hat{\boldsymbol{\beta}}_2^*-\hat{\boldsymbol{\beta}}_1\big)+O_p(n_1h_1\sqrt{p}\log p+\mathbf{R}_{n_1}).\quad(2.6)$$

By placing (2.6) into (2.3), we obtain

$$\frac{1}{N_2}\mathbf{J}(D_1;\hat{\boldsymbol{\beta}}_1;h_1)\big(\hat{\boldsymbol{\beta}}_2^*-\hat{\boldsymbol{\beta}}_1\big)+\frac{1}{N_2}\mathbf{U}\big(D_2;\hat{\boldsymbol{\beta}}_2^*\big)+O_p\left(\frac{n_1h_1\sqrt{p}\log p+\mathbf{R}_{n_1}}{N_2}\right)=\mathbf{0}.$$

When n_1 is sufficiently large, under some mild regularity conditions, both $\hat{\boldsymbol{\beta}}_1$ and $\hat{\boldsymbol{\beta}}_2^*$ are consistent estimators of $\boldsymbol{\beta}_0$. Moreover, taking sufficiently small bandwidth h_1, the error term may be asymptotically ignored. Removing such a term, we propose a new estimator $\hat{\boldsymbol{\beta}}_2$ as a solution to the equation of the form

$$\frac{1}{N_2}\mathbf{J}(D_1;\hat{\boldsymbol{\beta}}_1;h_1)(\hat{\boldsymbol{\beta}}_2-\hat{\boldsymbol{\beta}}_1)+\frac{1}{N_2}\mathbf{U}(D_2;\hat{\boldsymbol{\beta}}_2)=\mathbf{0}.\quad(2.7)$$

Through equation (2.7), the initial $\hat{\boldsymbol{\beta}}_1$ is renewed by $\hat{\boldsymbol{\beta}}_2$ only using the historical summary statistics, including sample variance matrix $\mathbf{J}(D_1;\hat{\boldsymbol{\beta}}_1;h_1)$ and estimate $\hat{\boldsymbol{\beta}}_1$, instead of the subject-level raw data D_1.

Generalizing the equation (2.7) to streaming data sets $\{D_1,\ldots,D_b\}$, a renewable estimator $\hat{\boldsymbol{\beta}}_b$ of $\boldsymbol{\beta}_0$ is defined as a solution to the following incremental estimation equation:

$$\frac{1}{N_b}\sum_{j=1}^{b-1}\mathbf{J}(D_j;\hat{\boldsymbol{\beta}}_j;h_j)(\hat{\boldsymbol{\beta}}_b-\hat{\boldsymbol{\beta}}_{b-1})+\frac{1}{N_b}\mathbf{U}(D_b;\hat{\boldsymbol{\beta}}_b)=\mathbf{0}.\quad(2.8)$$

2.3. Large sample properties

To establish the asymptotic properties of the proposed estimator, the following technical conditions are imposed.

C1. $A'(r,h,\tau)$ is a continuous, monotonically increasing nonnegative function on interval $[-\tau-h,-\tau+h]$, and $A(-\tau-h,h,\tau)=-\tau$, $A(-\tau+h,h,\tau)=-\tau+h$, $A'(-\tau-h,h,\tau)=0$, $A'(-\tau+h,h,\tau)=1$.

C2. The random vector $\mathbf{X}$ is sub-Gaussian: there exist $c_1>0$ such that

$$P\big(|\mathbf{X}^\top\boldsymbol{\delta}|\geq c_1\|\boldsymbol{\delta}\|_2t\big)\leq 2\exp(-t),$$

for all $\boldsymbol{\delta}\in R^p$ and $t\geq 0$. $\boldsymbol{\Sigma}=\mathrm{E}(\mathbf{X}\mathbf{X}^\top)$ is a positive definite matrix. Moreover, assume that $0<\Lambda_{\min}(\boldsymbol{\Sigma})\leq\Lambda_{\max}(\boldsymbol{\Sigma})<\infty$, where $\Lambda_{\min}(\boldsymbol{\Sigma})$ and $\Lambda_{\max}(\boldsymbol{\Sigma})$ are the smallest and largest eigenvalues of Σ, respectively.

C3. The density function $f(\cdot)$ and the variance of $\boldsymbol{\varepsilon}$ are bounded. In addition, the distribution function $F(\cdot)$ of $\boldsymbol{\varepsilon}$ is symmetric.

Remark 2.1. Condition **C1** is a mild condition on $A(r, h, \tau)$ for smoothing approximation. For example, (2.1) satisfies condition **C1**. Condition **C2** assumes a sub-Gaussian condition on the random covariates, which encompasses the bounded case considered by Loh (2021). The boundedness condition in **C3** is assumed for asymptotic covariance, see condition **C5** in Han et al. (2022b).

Theorem 2.1. *Suppose that conditions* **C1**–**C2** *are satisfied. If* $p = O(N_1^{c_2})$ *with* $0 < c_2 < 1$, $h_j = O(N_j^{-1/2}(\log p)^{-1})$ *with* $N_j = \sum_{i=1}^{j} n_i$ *for* $j = 1, \ldots, b$ *and* $N_1 \to \infty$, *and* $\tau \geq c_\tau \sigma$ *with* c_τ *being an appropriately constant, we have*

$$\|\hat{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0\|_2 = O_p(\sqrt{p/N_b}).$$

Remark 2.2. Here and also in Theorem 2.2 and Theorem 3.1, the condition $\tau \geq c_\tau \sigma$ incorporates $\tau = 1.345\sigma$ as a constant. Based on the simulation studies in Section 4.1, we could designate $\tau = 1.345 C_\tau \sigma$, where C_τ assumes values within the range of (0.1, 20).

Theorem 2.2. *Suppose that all conditions in Theorem 2.1 and condition* **C3** *hold. If* $p = o(\min\{N_1, \sqrt{N_b/\log N_b}\})$ *and* $h_j = o((pN_j)^{-1/2}(\log p)^{-1})$, *for any* $\boldsymbol{\alpha} \in \mathrm{R}^p$ *with* $\boldsymbol{\alpha} \neq \mathbf{0}$, *we have*

$$\sqrt{N_b}\boldsymbol{\alpha}^\top(\hat{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0)/\tilde{\sigma} \xrightarrow{L} \mathcal{N}(0, 1),$$

where $\tilde{\sigma}^2 = \boldsymbol{\alpha}^\top \Sigma^{-1} \boldsymbol{\alpha} \mathrm{E}[\ell'_\tau(\boldsymbol{\varepsilon})]^2 / \{\mathrm{E}[\ell''_\tau(\boldsymbol{\varepsilon})]\}^2$ *and* $\xrightarrow{L}$ *represents the convergence in the distribution.*

Note that the renewable estimator $\hat{\boldsymbol{\beta}}_b$ achieves optimal efficiency ($\sqrt{N_b}$-consistent) and its asymptotic covariance matrix is the same as that of estimator $\hat{\boldsymbol{\beta}}$ by (1.2), which is a direct one-time use of full data, as shown in Corollary 2.1 in He and Shao (2000).

Finally, we provide consistent analytical renewable estimators of the components of the variances: $\hat{\boldsymbol{\Sigma}} = N_b^{-1} \sum_{j=1}^{b} \sum_{i \in D_j} \mathbf{X}_i \mathbf{X}_i^\top$, $\hat{\mathrm{E}}[\ell'_\tau(\boldsymbol{\varepsilon})]^2 = N_b^{-1} \sum_{j=1}^{b} \sum_{i \in D_j} [\ell'_\tau(\hat{\varepsilon}_{i,j})]^2$ and $\hat{\mathrm{E}}[\ell''_\tau(\boldsymbol{\varepsilon})] = N_b^{-1} \sum_{j=1}^{b} \sum_{i \in D_j} \ell''_\tau(\hat{\varepsilon}_{i,j})$, where $\hat{\varepsilon}_{i,j} = Y_i - \mathbf{X}_i^\top \hat{\boldsymbol{\beta}}_j$. The consistency of these estimators follows from the law of large numbers and consistency of $\hat{\boldsymbol{\beta}}_j$ for $j = 1, \ldots, b$. As a result, a $1 - \alpha$ confidence interval for $\boldsymbol{\beta}_0$ is

$$\big[\hat{\boldsymbol{\beta}}_b - \Phi^{-1}(1 - \alpha/2) \times \hat{\mathrm{Sd}}, \hat{\boldsymbol{\beta}}_b + \Phi^{-1}(1 - \alpha/2) \times \hat{\mathrm{Sd}}\big], \tag{2.9}$$

where $\hat{\mathrm{Sd}} = \sqrt{\mathrm{diag}(\hat{\boldsymbol{\Sigma}}^{-1}) \hat{\mathrm{E}}[\ell'_\tau(\boldsymbol{\varepsilon})]^2 / \{\hat{\mathrm{E}}[\ell''_\tau(\boldsymbol{\varepsilon})]\}^2 / N_b}$ and $\Phi(\cdot)$ is the standard normal cumulative distribution function.

2.4. Algorithm

Numerically, it is quite straightforward to find $\hat{\boldsymbol{\beta}}_b$ from (2.8) using the Newton-Raphson method at the $(r+1)$-th iteration:

$$\hat{\boldsymbol{\beta}}_b^{(r+1)} = \hat{\boldsymbol{\beta}}_b^{(r)} - \big\{\hat{\mathbf{J}}_{b-1} + \mathbf{J}\big(D_b; \hat{\boldsymbol{\beta}}_b^{(r)}; h_b\big)\big\}^{-1} \hat{\mathbf{U}}_b^{(r)}, \tag{2.10}$$

where $\hat{\mathbf{J}}_{b-1} = \sum_{j=1}^{b-1} \mathbf{J}(D_j; \hat{\boldsymbol{\beta}}_j; h_j)$ and $\hat{\mathbf{U}}_b^{(r)} = \hat{\mathbf{J}}_{b-1}(\hat{\boldsymbol{\beta}}_b^{(r)} - \hat{\boldsymbol{\beta}}_{b-1}) + \mathbf{U}(D_b; \hat{\boldsymbol{\beta}}_b^{(r)})$.

Recall (2.10), which can be computationally expensive for calculating the inverse matrix of $\hat{\mathbf{J}}_{b-1} + \mathbf{J}(D_b; \hat{\boldsymbol{\beta}}_b^{(r)}; h_b)$ under a large p and needs modifications if it is ill-conditioned. For this reason, we use a Barzilai-Borwein (BB) method proposed by Barzilai and Borwein (1988) to get a simple approximation of the inverse matrix of $\hat{\mathbf{J}}_{b-1} + \mathbf{J}(D_b; \hat{\boldsymbol{\beta}}_b^{(r)}; h_b)$. The BB method chooses ω_r so that $\omega_r \hat{\mathbf{U}}_b^{(r)} = (\omega_r^{-1}\mathbf{I})^{-1}\hat{\mathbf{U}}_b^{(r)}$ approximates $\{\hat{\mathbf{J}}_{b-1} + \mathbf{J}(D_b; \hat{\boldsymbol{\beta}}_b^{(r)}; h_b)\}^{-1}\hat{\mathbf{U}}_b^{(r)}$, where $\mathbf{I}$ is a $p \times p$ unit matrix. Therefore, we compute ω_r such that

$$\big(\omega_r^{-1}\mathbf{I}\big)\boldsymbol{\eta}^{(r)} = \Psi^{(r)},$$

where $\Psi^{(r)} = \hat{\mathbf{U}}_b^{(r)} - \hat{\mathbf{U}}_b^{(r-1)}$ and $\boldsymbol{\eta}^{(r)} = \hat{\boldsymbol{\beta}}_b^{(r)} - \hat{\boldsymbol{\beta}}_b^{(r-1)}$. Via least squares approximations, we can obtain the ω_r by

$$\omega_r = \arg\min_{\gamma} \|\boldsymbol{\eta}^{(r)} - \gamma\Psi^{(r)}\|_2 = \frac{(\boldsymbol{\eta}^{(r)})^\top \Psi^{(r)}}{(\Psi^{(r)})^\top \Psi^{(r)}}.$$

Moreover, the ω_r computed in BB may sometimes vibrate drastically, causing instability of the algorithm. Therefore, as suggested by Luo, Sun and Zhou (2022), we take $\min\{\omega_r, 10\}$. Consequently, the iteration of (2.10) by the BB method is

$$\hat{\boldsymbol{\beta}}_b^{(r+1)} = \hat{\boldsymbol{\beta}}_b^{(r)} - \min\{\omega_r, 10\}\hat{\mathbf{U}}_b^{(r)}. \tag{2.11}$$

We summarize the general algorithm for the proposed renewable Huber estimation by (2.11) as follows.

Note that in step 7 in Algorithm 1, we only need to save $\hat{\boldsymbol{\beta}}_b$ and $\hat{\mathbf{J}}_b$, which are $p \times 1$ and $p \times p$, respectively. The scale of the data to be stored is $(p+1)p$ instead of $N_b p$, which is the sample size of the streaming data sets up to b batches. Because $p = O(N_1^{c_2})$ with $0 < c_2 < 1$ in Theorem 2.1, our method greatly reduces the amount of data storage.

3. High-dimensional estimation for Huber regression with streaming datasets

3.1. Methodologies

To avoid overfitting and improve the generalization ability, we first consider the penalized Huber estimator (PHE) based on all data (the batches up to the b-th

Algorithm 1 Renewable Huber estimation for streaming data sets.

1: **Input:** streaming data sets $D_1, \ldots, D_b, \ldots$, the Huber parameter τ and bandwidths h_b with $b = 1, 2 \ldots$
2: **Initialize:** calculate $\hat{\boldsymbol{\beta}}_1$ by minimizing (1.2) with D_1 and compute $\mathbf{J}(D_1; \hat{\boldsymbol{\beta}}_1; h_1)$;
3: **for** $b = 2, 3, \ldots$ **do**
4: read in data set D_b;
5: select the initial estimator $\hat{\boldsymbol{\beta}}_b^{(0)} = \hat{\boldsymbol{\beta}}_{b-1}$ and run the following iterations until convergence:

$$\hat{\boldsymbol{\beta}}_b^{(r+1)} = \hat{\boldsymbol{\beta}}_b^{(r)} - \min\{\omega_r, 10\}\hat{\mathbf{U}}_b^{(r)};$$

6: update $\hat{\mathbf{J}}_b = \hat{\mathbf{J}}_{b-1} + \mathbf{J}(D_b; \hat{\boldsymbol{\beta}}_b; h_b)$;
7: save $\hat{\boldsymbol{\beta}}_b$ and $\hat{\mathbf{J}}_b$, release $\hat{\boldsymbol{\beta}}_{b-1}$, $\hat{\mathbf{J}}_{b-1}$ and data set D_b from the memory;
8: **end for**
9: **Output:** $\hat{\boldsymbol{\beta}}_b$ for $b = 2, 3, \ldots$

batch of streaming data can be pooled into one data set):

$$\tilde{\boldsymbol{\beta}}^* = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \ell_\tau\big(Y_i - \mathbf{X}_i^\top \boldsymbol{\beta}\big) + \lambda \|\boldsymbol{\beta}\|_1 \right\}, \tag{3.1}$$

where λ is a regularization parameter and $\|\cdot\|_1$ is the absolute norm.

For streaming data sets, we begin with two batches of data D_1 and D_2. We can obtain the initial PHE $\tilde{\boldsymbol{\beta}}_1$ (or $\tilde{\boldsymbol{\beta}}_1^*$) based on D_1 as:

$$\tilde{\boldsymbol{\beta}}_1 = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{N_1} \sum_{i \in D_1} \ell_\tau\big(Y_i - \mathbf{X}_i^\top \boldsymbol{\beta}\big) + \lambda_1 \|\boldsymbol{\beta}\|_1 \right\}.$$

By convex optimization theory, $\tilde{\boldsymbol{\beta}}_1$ also satisfies the first-order condition

$$-\frac{1}{N_1} \mathbf{U}(D_1; \tilde{\boldsymbol{\beta}}_1) + \lambda_1 \, \mathbf{sgn}(\tilde{\boldsymbol{\beta}}_1) = \mathbf{0}, \tag{3.2}$$

where $\mathbf{sgn}(\boldsymbol{\beta}) = (\mathrm{sgn}(\beta_1), \ldots, \mathrm{sgn}(\beta_p))^\top$ and $\mathrm{sgn}(\beta_k)$ is the subgradient of $|\beta_k|$. Then, $\tilde{\boldsymbol{\beta}}_2^*$ satisfies the following aggregated score equation:

$$-\frac{1}{N_2} \mathbf{U}\big(D_1; \tilde{\boldsymbol{\beta}}_2^*\big) - \frac{1}{N_2} \mathbf{U}\big(D_2; \tilde{\boldsymbol{\beta}}_2^*\big) + \lambda_2 \, \mathbf{sgn}\big(\tilde{\boldsymbol{\beta}}_2^*\big) = \mathbf{0}. \tag{3.3}$$

By analysis similar to Section 2.2 and (3.2), we can get

$$\mathbf{U}\big(D_1; \tilde{\boldsymbol{\beta}}_2^*\big) = N_1 \lambda_1 \, \mathbf{sgn}(\tilde{\boldsymbol{\beta}}_1) + \mathbf{J}(D_1; \tilde{\boldsymbol{\beta}}_1; h_1)\big(\tilde{\boldsymbol{\beta}}_2^* - \tilde{\boldsymbol{\beta}}_1\big) + \Re_{N_1}, \tag{3.4}$$

where $\Re_{N_1}$ is an asymptotically ignored error term. By substituting (3.4) into (3.3) and removing the asymptotically ignored term $\Re_{N_1}$, we propose a new estimator $\tilde{\boldsymbol{\beta}}_2$ as a solution to the equation of the form

$$-\frac{1}{N_2} \mathbf{J}(D_1; \tilde{\boldsymbol{\beta}}_1; h_1)(\tilde{\boldsymbol{\beta}}_2 - \tilde{\boldsymbol{\beta}}_1) - \frac{1}{N_2} \mathbf{U}(D_2; \tilde{\boldsymbol{\beta}}_2) - \frac{N_1}{N_2} \lambda_1 \, \mathbf{sgn}(\tilde{\boldsymbol{\beta}}_1) + \lambda_2 \, \mathbf{sgn}(\tilde{\boldsymbol{\beta}}_2) = \mathbf{0}. \tag{3.5}$$

Through equation (3.5), the initial $\tilde{\boldsymbol{\beta}}_1$ is renewed by $\tilde{\boldsymbol{\beta}}_2$ using statistics $\mathbf{J}(D_1;\tilde{\boldsymbol{\beta}}_1;h_1)$, $\tilde{\boldsymbol{\beta}}_1$ and λ_1 instead of D_1.

Generalizing the above procedure to streaming data sets $\{D_1,\ldots,D_b\}$, a renewable penalized estimator $\tilde{\boldsymbol{\beta}}_b$ of $\boldsymbol{\beta}_0$ is defined as a solution to the following incremental estimating equation:

$$-\frac{1}{N_b}\sum_{j=1}^{b-1}\mathbf{J}(D_j;\tilde{\boldsymbol{\beta}}_j;h_j)(\tilde{\boldsymbol{\beta}}_b-\tilde{\boldsymbol{\beta}}_{b-1})-\frac{1}{N_b}\mathbf{U}(D_b;\tilde{\boldsymbol{\beta}}_b)$$
$$-\frac{N_{b-1}}{N_b}\lambda_{b-1}\,\mathbf{sgn}(\tilde{\boldsymbol{\beta}}_{b-1})+\lambda_b\,\mathbf{sgn}(\tilde{\boldsymbol{\beta}}_b)=\mathbf{0}. \tag{3.6}$$

The following theorem shows the asymptotic property of the estimator $\tilde{\boldsymbol{\beta}}_b$ in (3.6).

Theorem 3.1. *Let* $p=o(\exp(N_1^{c_3}))$ *with* $c_3>0$, $s=o(\min\{N_1/\log p,\sqrt{N_b/\log p}/\log N_b\})$ *and* $h_j=o((N_j\log p)^{-1/2})$ *for* $j=1,\ldots,b$. *Take* $\lambda_b=\bar{C}\sqrt{\log p/N_b}$ *with* $\bar{C}$ *being a sufficiently large constant and* $\tau\geq c_\tau\sigma$ *with* c_τ *being an appropriately constant. Under conditions* **C1**–**C2** *and* $N_1\to\infty$, *we can obtain*

$$\|\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|_2=O_p(\sqrt{s\log p/N_b}),$$

where s *is the number of non-zero coefficients* $\{k:\boldsymbol{\beta}_{0,k}\neq 0\}$.

Theorem 3.1 shows that the renewable estimator $\tilde{\boldsymbol{\beta}}_b$ in (3.6) achieves the same asymptotic property as the estimator $\tilde{\boldsymbol{\beta}}^*$ in (3.1), which is directly computed using all the samples ($n=N_b$), see Theorem 1 in Loh (2021).

3.2. Algorithm

This section is devoted to computational algorithm and numerical implementation for (3.6). Note that (3.6) is equal to

$$\tilde{\boldsymbol{\beta}}_b=\arg\min_{\boldsymbol{\beta}}\{\mathbf{H}_b(\boldsymbol{\beta})+\lambda_b\|\boldsymbol{\beta}\|_1\}, \tag{3.7}$$

where

$$\mathbf{H}_b(\boldsymbol{\beta})=\frac{1}{N_b}\sum_{i\in D_b}\ell_\tau\big(Y_i-\mathbf{X}_i^\top\boldsymbol{\beta}\big)-\frac{1}{2N_b}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_{b-1})^\top\tilde{\mathbf{J}}_{b-1}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_{b-1})$$
$$-\frac{N_{b-1}}{N_b}\lambda_{b-1}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_{b-1})^\top\,\mathbf{sgn}(\tilde{\boldsymbol{\beta}}_{b-1}), \tag{3.8}$$

and $\tilde{\mathbf{J}}_{b-1}=\sum_{j=1}^{b-1}\mathbf{J}(D_j;\tilde{\boldsymbol{\beta}}_j;h_j)$. We describe a fast and easily implementable method using the local adaptive majorization-minimization (LAMM) principle (Fan et al., 2018b). We therefore locally majorize $\mathbf{H}_b(\boldsymbol{\beta})$ in (3.8) at $\tilde{\boldsymbol{\beta}}_b^{(r)}$ using an isotropic quadratic function

$$g_b\big(\boldsymbol{\beta}|\tilde{\boldsymbol{\beta}}_b^{(r)}\big)=\mathbf{H}_b\big(\tilde{\boldsymbol{\beta}}_b^{(r)}\big)+\big(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_b^{(r)}\big)^\top\mathbf{H}_b'\big(\tilde{\boldsymbol{\beta}}_b^{(r)}\big)+\frac{\phi}{2}\|\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_b^{(r)}\|_2^2,$$

where ϕ is a quadratic parameter such that $g_b(\tilde{\boldsymbol{\beta}}_b^{(r+1)}|\tilde{\boldsymbol{\beta}}_b^{(r)}) \geq \mathbf{H}_b(\tilde{\boldsymbol{\beta}}_b^{(r+1)})$ and

$$\mathbf{H}_b'(\boldsymbol{\beta}) = -\frac{1}{N_b}\mathbf{U}(D_b, \boldsymbol{\beta}) - \frac{1}{N_b}\tilde{\mathbf{J}}_{b-1}(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}_{b-1}) - \frac{N_{b-1}}{N_b}\lambda_{b-1}\,\mathbf{sgn}(\tilde{\boldsymbol{\beta}}_{b-1}). \quad (3.9)$$

We can take $\tilde{\boldsymbol{\beta}}_{b-1}$ as an initial estimator $\tilde{\boldsymbol{\beta}}_b^{(0)}$. For simplicity, we take $\mathrm{sgn}(r) = \mathrm{sign}(r)$ for $|r| > 0$ and 0 for $r = 0$.

To find the smallest ϕ such that $g_b(\tilde{\boldsymbol{\beta}}_b^{(r+1)}|\tilde{\boldsymbol{\beta}}_b^{(r)}) \geq \mathbf{H}_b(\tilde{\boldsymbol{\beta}}_b^{(r+1)})$, the basic idea of LAMM is to start from a relatively small isotropic parameter $\phi = \phi_0 = 10^{-6}$, and then successfully inflate ϕ by a factor $\omega > 1$. We set $\omega = 10$ in the numerical studies. The isotropic form also allows a simple analytic solution to the subsequent majorized optimization problem:

$$\min_{\boldsymbol{\beta}}\left\{(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}_b^{(r)})^\top \mathbf{H}_b'(\tilde{\boldsymbol{\beta}}_b^{(r)}) + \frac{\phi}{2}\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}_b^{(r)}\|_2^2 + \lambda_b\|\boldsymbol{\beta}\|_1\right\}. \quad (3.10)$$

It can be shown that (3.10) is minimized at

$$\tilde{\boldsymbol{\beta}}_b^{(r+1)} = \mathrm{Soft}\big(\tilde{\boldsymbol{\beta}}_b^{(r)} - \phi^{-1}\mathbf{H}_b'(\tilde{\boldsymbol{\beta}}_b^{(r)}), \phi^{-1}\lambda_b\big), \quad (3.11)$$

where $\mathrm{Soft}(\boldsymbol{\mu}, \boldsymbol{\nu})$ is the soft-thresholding operator, defined by $\mathrm{Soft}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \mathrm{sign}(\boldsymbol{\mu})\max(|\boldsymbol{\mu}| - \boldsymbol{\nu}, 0)$. A simple stopping criterion for (3.11) is $\|\tilde{\boldsymbol{\beta}}_b^{(r+1)} - \tilde{\boldsymbol{\beta}}_b^{(r)}\|_2 \leq \epsilon$ for a sufficiently small ϵ, say 10^{-4}.

It is well known that the regularization parameter plays an important role in the penalized method. Following Wang, Li and Tsai (2007), we use Bayesian information criterion (BIC) to choose the optimal value of the regularization parameter λ_b in (3.6). Specifically, the BIC statistic is defined as

$$\mathrm{BIC}(\lambda_b) = \ln\big\{\mathbf{H}_b(\tilde{\boldsymbol{\beta}}_{b,\lambda_b})\big\} + df_{\lambda_b}\ln(N_b)/N_b, \quad (3.12)$$

where $\tilde{\boldsymbol{\beta}}_{b,\lambda_b}$ is the penalized estimator of $\boldsymbol{\beta}_0$ by (3.6) given λ_b and df_{λ_b} is the number of nonzero coefficients in $\tilde{\boldsymbol{\beta}}_{b,\lambda_b}$.

We summarize the general algorithm for the proposed renewable PHE estimation as follows.

Note that in step 14 in Algorithm 2, we only need to save $\tilde{\boldsymbol{\beta}}_b$, $\tilde{\mathbf{J}}_b$ and λ_b. The scale of data to be stored is $p^2 + p + 1$ instead of $N_b p$ (the sample size of streaming data sets up to b batches). Our proposed method for variable selection also greatly reduces the amount of data storage.

The convergence of Algorithms 1 and 2 can be found in Section 3 (pages 3303–3305) of Pan, Sun and Zhou (2021).

4. Numerical studies

In this section, we first use Monte Carlo simulation studies to assess the finite sample performance of the proposed procedures and then demonstrate the application of the proposed methods with a real data analysis. All programs are written in R code.

Algorithm 2 The renewable PHE estimation for streaming data sets.

Input: streaming data sets $D_1, \ldots, D_b, \ldots$, the Huber parameter τ and bandwidths h_b with $b = 1, 2 \ldots$
Initialize: calculate $\tilde{\boldsymbol{\beta}}_1$ and λ_1 with D_1 and compute $\mathbf{J}(D_1; \tilde{\boldsymbol{\beta}}_1; h_1)$;
for $b = 2, 3, \ldots$ **do**
  read in data set D_b;
  select the initial estimator $\tilde{\boldsymbol{\beta}}_b^{(0)} = \tilde{\boldsymbol{\beta}}_{b-1}$ and choose λ_b via (3.12);
  for $r = 0, 1, \ldots,$ until $\|\tilde{\boldsymbol{\beta}}_b^{(r+1)} - \tilde{\boldsymbol{\beta}}_b^{(r)}\|_2 \leq \epsilon$ **do**
    repeat
      $\tilde{\boldsymbol{\beta}}_b^{(r+1)} = \text{Soft}(\tilde{\boldsymbol{\beta}}_b^{(r)} - \phi^{-1}\mathbf{H}_b'(\tilde{\boldsymbol{\beta}}_b^{(r)}), \phi^{-1}\lambda_b)$;
      if $g_b(\tilde{\boldsymbol{\beta}}_b^{(r+1)}|\tilde{\boldsymbol{\beta}}_b^{(r)}) < \mathbf{H}_b(\tilde{\boldsymbol{\beta}}_b^{(r+1)})$ **then** $\phi \leftarrow 10\phi$;
    until $g_b(\tilde{\boldsymbol{\beta}}_b^{(r+1)}|\tilde{\boldsymbol{\beta}}_b^{(r)}) \geq \mathbf{H}_b(\tilde{\boldsymbol{\beta}}_b^{(r+1)})$;
    return $\tilde{\boldsymbol{\beta}}_b^{(r+1)}$ and $\phi \leftarrow \max\{10^{-6}, \phi/10\}$;
  end for
  update $\tilde{\mathbf{J}}_b = \tilde{\mathbf{J}}_{b-1} + \mathbf{J}(D_b; \tilde{\boldsymbol{\beta}}_b; h_b)$, where $\tilde{\boldsymbol{\beta}}_b = \tilde{\boldsymbol{\beta}}_b^{(r+1)}$;
  save $\tilde{\boldsymbol{\beta}}_b$, $\tilde{\mathbf{J}}_b$ and λ_b, release $\tilde{\boldsymbol{\beta}}_{b-1}$, $\tilde{\mathbf{J}}_{b-1}$, λ_{b-1} and data set D_b from the memory;
end for
Output: $\tilde{\boldsymbol{\beta}}_b$ for $b = 2, 3, \ldots$

4.1. Simulation example 1: smoothing Huber estimation

In this section, we study the performance of the smoothing Huber estimation (SHE) proposed in Section 2.1. We generate data from the following linear model:

$$\mathbf{Y} = \mathbf{X}^\top \boldsymbol{\beta}_0 + \boldsymbol{\varepsilon}, \tag{4.1}$$

where $\mathbf{X} = (\mathbf{1}, \mathbf{X}_1, \ldots, \mathbf{X}_{p-1})^\top$ is a p-dimensional covariate vector and $\mathbf{X}_j, j = 1, \ldots, p-1$ are drawn from a normal distribution $N(0, 1)$. The true value of the parameter is $\boldsymbol{\beta}_0 = (1, 1, 1, 1, 1, 0, \ldots, 0)^\top$ with $p = 10$ and 100. Two error distributions of $\boldsymbol{\varepsilon}$ are considered: a standard normal distribution $N(0, 1)$ and a t distribution with 5 degrees of freedom $t(5)$. The sample size is $N = 200$.

To evaluate the performance of the estimation method, we calculate the mean squared error (MSE): $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_2$. We also investigate the sensitivity of the proposed SHE method to the bandwidth selection h and shape parameter τ. Recall that the bandwidth is $h = C_h N^{-1/2}(\log p)^{-1}$ in Theorem 2.1 with $C_h > 0$ being the scaling constant. We take $\tau = 1.345 C_\tau \hat{\sigma}$, where $\hat{\sigma} = \text{median}|\hat{\boldsymbol{\varepsilon}} - \text{median}(\hat{\boldsymbol{\varepsilon}})|$, $\hat{\boldsymbol{\varepsilon}} = \mathbf{Y} - \mathbf{X}^\top \hat{\boldsymbol{\beta}}_{LS}$ and $\hat{\boldsymbol{\beta}}_{LS}$ denotes the least squares estimator. He et al. (2023) also considered this type of τ as $\tau = 1.345\hat{\sigma}$ ($C_\tau = 1$). We vary the constant C_h from 0.001 to 10 and C_τ from 0.1 to 20, respectively.

Simulation results of the SHE and Huber estimation (HE) by (1.2) are based on 500 simulation replications. The results are shown in Table 1. As can be seen from Table 1 that the MSEs of SHE and HE are very close under different errors and dimensions p. In addition, SHE is insensitive to bandwidth h. The Huber estimators (HE and SHE) depends on τ as expected, but the effect is not large according to MSEs in Table 1.

TABLE 1
The means and standard deviations (in parentheses) of MSEs under different C_τ, C_h, methods, dimensions p and errors for simulation example 1.

			$N(0,1)$		$t(5)$	
C_τ	Method	C_h	$p=10$	$p=100$	$p=10$	$p=100$
0.1	HE	–	0.273 (0.061)	1.042 (0.101)	0.290 (0.070)	1.288 (0.142)
	SHE	0.001	0.273 (0.061)	1.042 (0.101)	0.290 (0.070)	1.288 (0.142)
		0.01	0.273 (0.061)	1.042 (0.101)	0.290 (0.070)	1.288 (0.142)
		0.1	0.273 (0.061)	1.042 (0.101)	0.290 (0.070)	1.288 (0.142)
		1	0.273 (0.061)	1.045 (0.100)	0.290 (0.070)	1.289 (0.142)
		10	0.258 (0.057)	1.011 (0.102)	0.286 (0.069)	1.282 (0.144)
0.5	HE	–	0.251 (0.057)	1.091 (0.104)	0.272 (0.065)	1.291 (0.142)
	SHE	0.001	0.251 (0.057)	1.091 (0.104)	0.272 (0.065)	1.291 (0.142)
		0.01	0.251 (0.057)	1.091 (0.104)	0.272 (0.065)	1.291 (0.142)
		0.1	0.251 (0.057)	1.091 (0.104)	0.272 (0.065)	1.291 (0.142)
		1	0.251 (0.056)	1.092 (0.105)	0.272 (0.065)	1.291 (0.142)
		10	0.247 (0.057)	1.097 (0.105)	0.269 (0.064)	1.294 (0.143)
1	HE	–	0.240 (0.060)	1.061 (0.102)	0.261 (0.063)	1.254 (0.137)
	SHE	0.001	0.240 (0.060)	1.061 (0.102)	0.261 (0.063)	1.254 (0.137)
		0.01	0.240 (0.060)	1.061 (0.102)	0.261 (0.063)	1.254 (0.137)
		0.1	0.240 (0.060)	1.061 (0.102)	0.261 (0.063)	1.254 (0.137)
		1	0.240 (0.060)	1.061 (0.102)	0.261 (0.063)	1.254 (0.137)
		10	0.239 (0.056)	1.062 (0.103)	0.261 (0.062)	1.254 (0.137)
5	HE	–	0.225 (0.051)	1.005 (0.096)	0.278 (0.066)	1.281 (0.151)
	SHE	0.001	0.225 (0.051)	1.005 (0.096)	0.278 (0.066)	1.281 (0.151)
		0.01	0.225 (0.051)	1.005 (0.096)	0.278 (0.066)	1.281 (0.151)
		0.1	0.225 (0.051)	1.005 (0.096)	0.278 (0.066)	1.281 (0.151)
		1	0.225 (0.051)	1.005 (0.096)	0.278 (0.066)	1.281 (0.151)
		10	0.225 (0.051)	1.005 (0.096)	0.278 (0.066)	1.281 (0.151)
10	HE	–	0.223 (0.050)	1.000 (0.095)	0.292 (0.069)	1.290 (0.157)
	SHE	0.001	0.223 (0.050)	1.000 (0.095)	0.292 (0.069)	1.290 (0.157)
		0.01	0.223 (0.050)	1.000 (0.095)	0.292 (0.069)	1.290 (0.157)
		0.1	0.223 (0.050)	1.000 (0.095)	0.292 (0.069)	1.290 (0.157)
		1	0.223 (0.050)	1.000 (0.095)	0.292 (0.069)	1.290 (0.157)
		10	0.223 (0.050)	1.000 (0.095)	0.292 (0.069)	1.290 (0.157)
20	HE	–	0.224 (0.051)	1.001 (0.097)	0.285 (0.068)	1.295 (0.174)
	SHE	0.001	0.224 (0.051)	1.001 (0.097)	0.285 (0.068)	1.295 (0.174)
		0.01	0.224 (0.051)	1.001 (0.097)	0.285 (0.068)	1.295 (0.174)
		0.1	0.224 (0.051)	1.001 (0.097)	0.285 (0.068)	1.295 (0.174)
		1	0.224 (0.051)	1.001 (0.097)	0.285 (0.068)	1.295 (0.174)
		10	0.224 (0.051)	1.001 (0.097)	0.285 (0.068)	1.295 (0.174)

4.2. Simulation example 2: renewable smoothing Huber estimation

In this section, we study the performance of the renewable smoothing Huber estimation (RSHE) proposed in Section 2.

The data are generated from the linear model (4.1). We fix the sample size of each batch to $n = 200$ and vary the number of batches $b = 100, 200, 500, 1000, 2000, 5000, 10000, 20000$. Simulation results are based on 100 simulation replications. Based on the analysis in Simulation example 1, we adopt $\tau = 1.345\hat{\sigma}$ ($C_\tau = 1$) as chose in He et al. (2023), where $\hat{\sigma}$ is based on the first streaming data D_1.

We first study the sensitivity of RSHE to bandwidth. Recall that the bandwidth is $h_j = CN_j^{-1/2}(\log p)^{-1}$ in Theorem 2.1 with $C > 0$ being the scaling

TABLE 2
The means and standard deviations (in parentheses) of the MSEs (×100) under different C for simulation example 2 with $\varepsilon \sim N(0,1)$.

p	b	$C = 0.001$	0.01	0.1	1	10
10	100	2.242 (0.511)	2.260 (0.514)	2.261 (0.514)	2.261 (0.514)	2.261 (0.514)
	200	1.643 (0.359)	1.642 (0.363)	1.642 (0.363)	1.642 (0.363)	1.642 (0.363)
	500	1.019 (0.215)	1.022 (0.215)	1.022 (0.215)	1.022 (0.215)	1.022 (0.215)
	1000	0.754 (0.156)	0.755 (0.156)	0.755 (0.156)	0.755 (0.156)	0.755 (0.156)
	2000	0.517 (0.124)	0.518 (0.124)	0.518 (0.124)	0.518 (0.124)	0.518 (0.124)
	5000	0.334 (0.070)	0.334 (0.070)	0.334 (0.070)	0.334 (0.070)	0.334 (0.070)
	10000	0.240 (0.054)	0.240 (0.054)	0.240 (0.054)	0.240 (0.054)	0.240 (0.054)
	20000	0.158 (0.038)	0.158 (0.037)	0.158 (0.037)	0.158 (0.037)	0.158 (0.037)
100	100	7.778 (0.624)	7.799 (0.619)	7.803 (0.618)	7.801 (0.618)	7.799 (0.618)
	200	5.473 (0.426)	5.477 (0.414)	5.478 (0.414)	5.478 (0.414)	5.475 (0.414)
	500	3.462 (0.250)	3.462 (0.249)	3.462 (0.249)	3.462 (0.248)	3.462 (0.248)
	1000	2.419 (0.166)	2.414 (0.168)	2.414 (0.168)	2.414 (0.168)	2.414 (0.168)
	2000	1.697 (0.139)	1.698 (0.138)	1.698 (0.138)	1.698 (0.138)	1.698 (0.138)
	5000	1.090 (0.076)	1.091 (0.076)	1.091 (0.076)	1.091 (0.076)	1.091 (0.076)
	10000	0.761 (0.039)	0.761 (0.040)	0.761 (0.040)	0.761 (0.040)	0.761 (0.040)
	20000	0.556 (0.039)	0.556 (0.039)	0.556 (0.039)	0.556 (0.039)	0.556 (0.039)

TABLE 3
The means and standard deviations (in parentheses) of the MSEs (×100) under different C for simulation example 2 with $\varepsilon \sim t(5)$.

p	b	$C = 0.001$	0.01	0.1	1	10
10	100	2.558 (0.590)	2.569 (0.604)	2.569 (0.605)	2.569 (0.605)	2.569 (0.605)
	200	1.795 (0.364)	1.799 (0.366)	1.799 (0.366)	1.799 (0.366)	1.799 (0.366)
	500	1.152 (0.269)	1.152 (0.267)	1.152 (0.267)	1.152 (0.267)	1.152 (0.267)
	1000	0.800 (0.189)	0.799 (0.189)	0.799 (0.189)	0.799 (0.189)	0.799 (0.190)
	2000	0.577 (0.114)	0.577 (0.114)	0.577 (0.114)	0.577 (0.114)	0.577 (0.114)
	5000	0.359 (0.084)	0.359 (0.084)	0.359 (0.084)	0.359 (0.084)	0.359 (0.084)
	10000	0.255 (0.057)	0.255 (0.057)	0.255 (0.057)	0.255 (0.057)	0.255 (0.057)
	20000	0.178 (0.041)	0.178 (0.041)	0.178 (0.041)	0.178 (0.041)	0.178 (0.041)
100	100	8.352 (0.611)	8.371 (0.644)	8.374 (0.644)	8.373 (0.643)	8.368 (0.641)
	200	5.828 (0.390)	5.839 (0.401)	5.839 (0.401)	5.839 (0.400)	5.837 (0.401)
	500	3.664 (0.226)	3.670 (0.221)	3.670 (0.222)	3.670 (0.221)	3.670 (0.221)
	1000	2.620 (0.191)	2.625 (0.194)	2.625 (0.194)	2.625 (0.194)	2.625 (0.194)
	2000	1.830 (0.108)	1.830 (0.107)	1.830 (0.107)	1.830 (0.106)	1.830 (0.107)
	5000	1.182 (0.079)	1.183 (0.079)	1.183 (0.079)	1.183 (0.079)	1.183 (0.079)
	10000	0.829 (0.061)	0.828 (0.061)	0.828 (0.061)	0.828 (0.061)	0.828 (0.061)
	20000	0.572 (0.030)	0.572 (0.030)	0.572 (0.030)	0.572 (0.030)	0.572 (0.030)

constant. We vary the constant C from 0.001 to 10. The results in Tables 2 and 3 show that RSHE is also insensitive to bandwidth h.

Furthermore, we compare our proposed RSHE (with $h_j = N_j^{-1/2}(\log p)^{-1}$) with the following two competitors: (1) the Huber estimator (HE) with full data; and (2) the average Huber estimator (AHE) for the streaming data set, that is, estimate each streaming data separately and then take its average. To evaluate the performance of the three methods, we calculate the MSE and computation time (in seconds). From Tables 4–7, the following conclusions can be drawn:

(1) In terms of the MSEs in Tables 4 and 5, we note that (i) all the estimators are close to the true value because the MSEs are very small; and (ii) for

TABLE 4
The means and standard deviations (in parentheses) of the MSEs (×100) under different methods for simulation example 2 with $\varepsilon \sim N(0,1)$.

p	b	HE	AHE	RSHE
10	100	2.311 (0.510)	2.369 (0.504)	2.313 (0.512)
	200	1.597 (0.356)	1.640 (0.351)	1.597 (0.355)
	500	1.044 (0.238)	1.065 (0.250)	1.044 (0.238)
	1000	0.729 (0.143)	0.751 (0.150)	0.730 (0.142)
	2000	0.513 (0.130)	0.523 (0.140)	0.513 (0.130)
	5000	0.329 (0.070)	0.334 (0.076)	0.329 (0.070)
	10000	0.237 (0.047)	0.239 (0.045)	0.237 (0.047)
	20000	0.163 (0.036)	0.166 (0.039)	0.163 (0.036)
100	100	7.722 (0.605)	10.349 (0.765)	7.806 (0.611)
	200	5.542 (0.357)	7.532 (0.445)	5.579 (0.355)
	500	3.447 (0.250)	4.878 (0.352)	3.455 (0.252)
	1000	2.505 (0.206)	3.676 (0.297)	2.509 (0.205)
	2000	1.728 (0.122)	2.715 (0.298)	1.730 (0.123)
	5000	1.075 (0.081)	2.020 (0.227)	1.076 (0.081)
	10000	0.780 (0.053)	1.740 (0.291)	0.780 (0.053)
	20000	0.557 (0.037)	1.591 (0.194)	0.557 (0.037)

TABLE 5
The means and standard deviations (in parentheses) of the MSEs (×100) under different methods for simulation example 2 with $\varepsilon \sim t(5)$.

p	b	HE	AHE	RSHE
10	100	2.542 (0.574)	2.633 (0.597)	2.541 (0.574)
	200	1.764 (0.371)	1.836 (0.395)	1.764 (0.372)
	500	1.137 (0.245)	1.185 (0.266)	1.137 (0.245)
	1000	0.824 (0.164)	0.843 (0.166)	0.824 (0.164)
	2000	0.561 (0.126)	0.590 (0.131)	0.561 (0.126)
	5000	0.349 (0.081)	0.367 (0.085)	0.349 (0.081)
	10000	0.247 (0.057)	0.256 (0.055)	0.247 (0.057)
	20000	0.209 (0.040)	0.222 (0.041)	0.209 (0.040)
100	100	8.151 (0.495)	12.370 (0.853)	8.207 (0.503)
	200	5.937 (0.454)	8.754 (0.627)	5.950 (0.460)
	500	3.717 (0.268)	5.784 (0.417)	3.703 (0.267)
	1000	2.672 (0.257)	4.283 (0.372)	2.670 (0.253)
	2000	1.853 (0.126)	3.222 (0.359)	1.857 (0.125)
	5000	1.158 (0.087)	2.408 (0.324)	1.158 (0.088)
	10000	0.811 (0.061)	1.994 (0.287)	0.819 (0.061)
	20000	0.613 (0.037)	1.832 (0.249)	0.613 (0.029)

any given number of batches b, p and errors, the MSEs of the proposed estimator (RSHE) are very close to those of HE and better than those of AHE.

(2) In terms of the computation time (in seconds) in Table 6, we note that (i) under $p = 10$, the computation times of the RSHE are close to those of HE and less than those of AHE, and (ii) under $p = 100$ and large $b = 20000$, the RSHE are faster than HE under different errors.

(3) In Table 7, we study the coverage probability of the interval estimate in (2.9). Since the results are similar for all components in $\boldsymbol{\beta}_0$, only the results on $\boldsymbol{\beta}_2 = 1$ is reported in Table 7. It can be seen that the coverage proba-

TABLE 6
The mean computing time (in seconds) under different methods for simulation example 2.

		$N(0,1)$			$t(5)$		
p	b	HE	AHE	RSHE	HE	AHE	RSHE
10	100	0.008	0.039	0.022	0.008	0.040	0.027
	200	0.017	0.079	0.042	0.016	0.080	0.042
	500	0.052	0.197	0.099	0.042	0.199	0.097
	1000	0.119	0.393	0.183	0.089	0.399	0.183
	2000	0.269	0.784	0.301	0.191	0.794	0.302
	5000	0.724	1.951	0.644	0.571	1.985	0.637
	10000	1.467	3.935	1.205	1.125	3.965	1.191
	20000	2.923	7.881	2.384	2.344	7.924	2.335
100	100	0.108	0.244	0.233	0.109	0.238	0.231
	200	0.226	0.481	0.446	0.215	0.476	0.432
	500	0.572	1.191	1.022	0.568	1.189	1.012
	1000	1.124	2.380	1.962	1.126	2.377	1.950
	2000	2.622	4.763	3.844	2.641	4.759	3.865
	5000	7.053	12.048	8.878	7.887	11.898	8.951
	10000	31.657	24.123	37.603	27.909	24.155	35.565
	20000	400.682	48.203	81.181	433.805	47.913	76.680

TABLE 7
The coverage probability of 90% confidence interval under different batches b and errors for simulation example 2.

Errors	p	$b = 100$	200	500	1000	2000	5000	10000	20000
$N(0,1)$	10	0.95	0.87	0.84	0.88	0.92	0.90	0.85	0.84
	100	0.91	0.90	0.95	0.91	0.91	0.91	0.92	0.90
$t(5)$	10	0.86	0.86	0.92	0.94	0.95	0.89	0.84	0.91
	100	0.88	0.85	0.90	0.85	0.90	0.89	0.86	0.92

bility are all around the nominal level (0.90). Therefore, the construction of confidence interval is valid.

4.3. Simulation example 3: renewable penalized smoothing Huber estimation

In this section, we study the performances of the renewable penalized smoothing Huber estimator (RPSHE) method proposed in Section 3. The data are generated from the following linear model:

$$\mathbf{Y} = \mathbf{X}^\top \boldsymbol{\beta}_0 + 0.2\boldsymbol{\varepsilon},$$

where $\boldsymbol{\beta}_0 = (1, 1, 1, 0.5, 0.3, 0.1, 0, \ldots, 0)$ and $p = 200$. We fix the sample size of each batch as $n = 100$. Other settings are the same as in simulation example 2. According to Theorem 3.1, we choose $h_j = N_j^{-1/2}(\log p)^{-1}$ and $\lambda_j = 0.5\tau\sqrt{\log p/N_j}$ for simplicity.

To evaluate the performance of RPSHE, we calculate the MSE in simulation example 1, the average proportion of nonzero coefficients correctly estimated to be nonzero (denoted as $\mathbf{C}$), and the average proportion of zero coefficients

TABLE 8
The means and standard deviations (in parentheses) of the MSEs (×100) under different batches b, methods and errors for simulation example 3.

Errors	b	PHE	PSHE	RPSHE
$N(0,1)$	100	1.129 (0.190)	1.146 (0.205)	0.654 (0.188)
	200	0.778 (0.099)	0.794 (0.112)	0.424 (0.124)
	500	0.494 (0.071)	0.502 (0.076)	0.249 (0.069)
	1000	0.354 (0.050)	0.378 (0.058)	0.180 (0.056)
	2000	0.234 (0.057)	0.246 (0.064)	0.135 (0.034)
	5000	0.158 (0.029)	0.165 (0.041)	0.088 (0.018)
	10000	0.110 (0.025)	0.111 (0.023)	0.054 (0.019)
	20000	0.073 (0.009)	0.087 (0.017)	0.030 (0.011)
$t(5)$	100	1.210 (0.162)	1.220 (0.165)	0.756 (0.157)
	200	0.876 (0.151)	0.889 (0.159)	0.495 (0.139)
	500	0.542 (0.088)	0.573 (0.126)	0.272 (0.077)
	1000	0.402 (0.062)	0.424 (0.080)	0.203 (0.075)
	2000	0.273 (0.034)	0.291 (0.041)	0.129 (0.043)
	5000	0.177 (0.024)	0.206 (0.040)	0.069 (0.027)
	10000	0.132 (0.018)	0.133 (0.018)	0.051 (0.014)
	20000	0.093 (0.006)	0.096 (0.005)	0.045 (0.007)

TABLE 9
The means of IC under different batches b, methods and errors for simulation example 3.

	$N(0,1)$			$t(5)$		
b	PHE	PSHE	RPSHE	PHE	PSHE	RPSHE
100	0.112	0.113	0.010	0.113	0.113	0.015
200	0.109	0.109	0.009	0.119	0.120	0.001
500	0.108	0.108	0.005	0.116	0.116	0.007
1000	0.104	0.104	0.005	0.118	0.119	0.004
2000	0.109	0.110	0.003	0.106	0.106	0.003
5000	0.126	0.126	0.002	0.117	0.117	0.002
10000	0.119	0.119	0.001	0.127	0.127	0.001
20000	0.115	0.115	0.001	0.115	0.115	0.002

incorrectly estimated to be nonzero (denoted as **IC**). Note that $\mathbf{C} = 1$ and $\mathbf{IC} = 0$ imply perfect recovery. Moreover, we compare our proposed method with PHE in (3.1) and penalized smoothing Huber estimator (PSHE), which is used the smoothing Huber loss instead of the ordinary Huber loss in (3.1). PHE and PSHE are directly used all data. Simulation results are presented in Tables 8 and 9 based on 100 simulation replications.

From Tables 8 and 9, the following conclusions can be drawn: (i) three estimators are close to the true value because the MSEs are very small and **C**s are all equal to one; (ii) the results of MSE and **IC** show that PSHE is close to PHE; (iii) an interesting result is that the MSE values of RPSHE are less than those of PHE and PSHE. The reason for this should be that RPSHE correctly selects the variable (with smaller **IC**).

4.4. Real data example: YearPredictionMSD data set

As an illustration, we now apply the proposed methodologies in Sections 2 and 3 to the YearPredictionMSD dataset. The dataset is collected from the public

TABLE 10
The MAPEs, MAPEs (Penalized) and NOVSs of the LSE, HE, SHE, AHE, and RSHE estimators under different bs for real data example.

Method	MAPE	MAPE (Penalized)	NOVS
LSE (All data)	6.906	6.906	83
QR (All data)	6.694	6.694	51
HE (All data)	6.709	6.743	89
SHE (All data)	6.698	6.743	89
RQR ($b = 100$)	6.703	6.704	52
RQR ($b = 200$)	6.718	6.717	52
RQR ($b = 500$)	6.821	6.819	50
RQR ($b = 1000$)	6.961	6.963	48
AHE ($b = 100$)	6.704	6.777	90
AHE ($b = 200$)	6.710	6.800	90
AHE ($b = 500$)	6.729	6.854	90
AHE ($b = 1000$)	6.759	6.911	90
RSHE ($b = 100$)	6.695	6.697	63
RSHE ($b = 200$)	6.696	6.705	54
RSHE ($b = 500$)	6.695	6.725	48
RSHE ($b = 1000$)	6.696	6.747	39

database of the UCI machine learning repository (https://archive.ics.uci.edu/dataset/203/yearpredictionmsd). It is extracted from the million song dataset, which consists of 515,345 songs ranging 1922–2011 with a peak in the year 2000s. The research problem is to retrieve songs released in a particular year based on the features of audio content. Feature extraction is perform using Echo Nest API, and produces 90 audio features in total, including 12 timbre averages and 78 timbre covariances. The average and covariance are calculated over a set of segments of a song, where each segment being described as a 12-dimensional timbre vector. The target value is the release year of song tracks (between 1922 to 2011). Jiang and Yu (2022) also studied this dataset by quantile regression (QR).

In this study, model (1.1), where the year of a song is the dependent variable ($\mathbf{Y}$) and the 12 average timbre and 78 timbre covariance variables are the covariate variables, is used to fit the data. To evaluate the performances of our proposed methods (RSHE and RPSHE) in Sections 2 and 3, we calculate the mean absolute prediction error (MAPE) of the predictions. The first 500000 data points are used for the estimation, and the remaining 15345 data points are used for the prediction. Therefore,

$$\text{MAPE} = \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} |Y_i - \hat{Y}_i|,$$

where $\hat{Y}_i$ is the fitted value of Y_i and $\tilde{n} = 15345$.

We choose $\tau = 1.345\hat{\sigma}$, where $\hat{\sigma} = \text{median}|\hat{\boldsymbol{\varepsilon}} - \text{median}(\hat{\boldsymbol{\varepsilon}})|$ anf $\hat{\boldsymbol{\varepsilon}}$ is an error estimation based on LSE. In addition, we take $h_j = N_j^{-1/2}(\log p)^{-1}$ for RSHE, and $h_j = N_j^{-1/2}(\log p)^{-1}$ and $\lambda_j = 0.5\tau\sqrt{\log p/N_j}$ for RPSHE, respectively. The results of MAPEs are presented in Table 10, and the table clearly shows

that (i) in terms of MAPE and MAPE (Penalized) which is the MAPE based on penalized estimation method, the performances of RSHE are very close to QR, HE and SHE and better than LSE, AHE and RQR (Jiang and Yu, 2022), and (ii) we also study the number of variables selected (NOVS), which indicates that the LASSO produces a small model because the numbers of variables selected under different batches are all smaller than the case with $p = 90$ variables. The performances of penalized AHE method are as poor as expected because of $p = 90$ under different batches.

5. Conclusion

The goal of this work is to develop renewable parameter estimation and variable selection for a Huber regression with high-dimensional streaming data sets. One key insight from this work is that a smoothing technique is adopted to transform the ordinary Huber loss function into a twice continuously differentiable loss function, which helps to produce renewable estimators for Huber regression. The renewable estimators require only the availability of the current data batch in the data stream and sufficient statistics on the historical data (the latest estimator, the cumulative Hessian matrix and the latest regularization parameter for variable selection) in each stage of the analysis. Theoretically, the proposed estimators achieve optimal efficiency, and their asymptotic properties are the same as those of the estimators with full data. In addition, the proposed renewable methods are all free of the constraint on the number of batches, which means that the new methods are adaptive to the situation where streaming data sets arrive fast and perpetually. Algorithms 1–2 for proposed methods are all fast and scalable.

It can be seen from numerical studies that the performance of smoothing Huber estimation is very close to that of ordinary Huber estimation, and smoothing Huber estimation is not sensitive to bandwidth selection. In terms of estimation accuracy, our proposed renewable parameter estimation and variable selection are similar to the Huber estimator with full data directly, but the running time is smaller than that of Huber estimator with full data directly.

For the analysis of streaming data sets, the smoothing technique of the sign function for Huber estimator in this paper can be also used for other estimation methods, such as logistic regression.

Appendix A: Proof of main results

Lemma 1. *Assume that conditions* **C1** *and* **C2** *are satisfied, for any* $\boldsymbol{\beta} \in \mathrm{R}^p$ *and* $j = 1, \ldots, b$, *we have*

$$\left\|\mathbf{U}(D_j; \boldsymbol{\beta}) - \tilde{\mathbf{U}}(D_j; \boldsymbol{\beta}; h_j)\right\|_2 = O_p(n_j h_j \sqrt{p} \log p).$$

Proof. Let $\mathbf{Q} = \mathbf{U}(D_j; \boldsymbol{\beta}) - \tilde{\mathbf{U}}(D_j; \boldsymbol{\beta}; h_j)$. For each $1 \leq k \leq p$, based on the

condition **C1**, we see that

$$\begin{aligned}\mathbf{e}_k^\top \mathbf{Q} &= \sum_{i\in D_j} \mathbf{e}_k^\top \mathbf{X}_i \{\ell'_\tau(Y_i - \mathbf{X}_i^\top \boldsymbol{\beta}) - \tilde{\ell}'_{\tau,h}(Y_i - \mathbf{X}_i^\top \boldsymbol{\beta})\} \\ &\le \sum_{i\in D_j} |\mathbf{e}_k^\top \mathbf{X}_i| |\ell'_\tau(Y_i - \mathbf{X}_i^\top \boldsymbol{\beta}) - \tilde{\ell}'_{\tau,h}(Y_i - \mathbf{X}_i^\top \boldsymbol{\beta})| \\ &\le h \sum_{i\in D_j} |\mathbf{e}_k^\top \mathbf{X}_i|\end{aligned}$$

where $\mathbf{e}_k$ is the k-th one, other zero. Then by condition **C2** and $p \to \infty$, we have

$$\mathrm{P}(\|\mathbf{Q}\|_\infty \ge 2c_1 n_j h_j \log p) \le \mathrm{P}\left(h_j \max_{1\le k\le p} \sum_{i\in D_j} |\mathbf{e}_k^\top \mathbf{X}_i| \ge 2c_1 n_j h_j \log p\right) \le 2p^{-1} \to 0,$$

where $\|\cdot\|_\infty$ is the maximal absolute value in the components of a vector. Finally, by $\|\mathbf{Q}\|_2 \le \sqrt{p}\|\mathbf{Q}\|_\infty$, we can prove the Lemma 1. □

Lemma 2. *Assume that conditions* **C1**–**C3** *are satisfied. Then, for any* $j = 1, \ldots, b$, *we have*

$$\left\| \sum_{i\in D_j} \mathbf{X}_i \mathbf{X}_i^\top \tilde{\ell}''_{\tau,h_j}(\varepsilon_i) - \sum_{i\in D_j} \mathbf{X}_i \mathbf{X}_i^\top \ell''_\tau(\varepsilon_i) \right\| = O_p(n_j h_j + \sqrt{n_j h_j \log p}),$$

where $\|\cdot\|$ *is spectral norm.*

Proof. By the proof of Lemma 3 in Cai, Zhang and Zhou (2010), we have

$$\begin{aligned}&\left\| \sum_{i\in D_j} \mathbf{X}_i \mathbf{X}_i^\top \tilde{\ell}''_{\tau,h_j}(\varepsilon_i) - \sum_{i\in D_j} \mathbf{X}_i \mathbf{X}_i^\top \ell''_\tau(\varepsilon_i) \right\| \\ &\quad\le 5 \sup_{k\le C_1} \left| \boldsymbol{\nu}_k^\top \sum_{i\in D_j} \mathbf{X}_i \mathbf{X}_i^\top \{\tilde{\ell}''_{\tau,h_j}(\varepsilon_i) - \ell''_\tau(\varepsilon_i)\} \boldsymbol{\nu}_k \right|,\end{aligned}$$

where $\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_{C_1}$ are some non-random vectors with $\|\boldsymbol{\nu}_k\|_2 = 1$ and $C_1 \le 5^p$. Now let

$$\mathbf{H}_k = \sum_{i\in D_j} (\boldsymbol{\nu}_k^\top \mathbf{X}_i)^2 \{\tilde{\ell}''_{\tau,h_j}(\varepsilon_i) - \ell''_\tau(\varepsilon_i)\}.$$

By conditions **C1**–**C3** and a large enough constant C_2, we have

$$\begin{aligned}\mathrm{E}(\mathbf{H}_k) &= \sum_{i\in D_j} \mathrm{E}\big[(\boldsymbol{\nu}_k^\top \mathbf{X}_i)^2 \{\tilde{\ell}''_{\tau,h_j}(\varepsilon_i) - \ell''_\tau(\varepsilon_i)\}\big] \\ &\le \sum_{i\in D_j} \mathrm{E}(\boldsymbol{\nu}_k^\top \mathbf{X}_i)^2 \mathrm{E}|\tilde{\ell}''_{\tau,h_j}(\varepsilon_i) - \ell''_\tau(\varepsilon_i)| \\ &\le \Lambda_{\max}(\Sigma) \sum_{i\in D_j} \mathrm{E}\{\mathrm{I}(-\tau - h_j \le \varepsilon_i \le -\tau + h_j) + \mathrm{I}(\tau - h_j \le \varepsilon_i \le \tau + h_j)\}\end{aligned}$$

$$
\begin{aligned}
&= 2\Lambda_{\max}(\Sigma)\sum_{i\in D_j}\big\{P(-\tau-h_j\le\varepsilon_i\le-\tau+h_j)+\mathrm{P}(\tau-h_j\le\varepsilon_i\le\tau+h_j)\big\}\\
&= n_j\Lambda_{\max}(\Sigma)\big\{\mathrm{F}(-\tau+h_j)-\mathrm{F}(-\tau-h_j)+\mathrm{F}(\tau+h_j)-\mathrm{F}(\tau-h_j)\big\}\\
&= 2h_jn_j\Lambda_{\max}(\Sigma)\big\{f(s_1)+f(s_2)\big\}\\
&\le C_2h_jn_j,
\end{aligned}
$$

where s_1 and s_2 are in $(-\tau-h_j,-\tau+h_j)$ and $(\tau-h_j,\tau+h_j)$, respectively. Then by Lemma 1 in Cai and Liu (2011), we can obtain that

$$
\sup_k P\big(\big|\mathbf{H}_k-\mathrm{E}(\mathbf{H}_k)\big|\ge 4\sqrt{h_jn_j\log p}\big)\le p^{-3}.
$$

Thus, we can prove the Lemma. □

Lemma 3. *Assume that conditions* **C1**–**C3** *are satisfied. Then, for any* $\boldsymbol{\beta}_1,\boldsymbol{\beta}_2\in\mathrm{R}^p$ *and* $j=1,\ldots,b$, *we have*

$$
\big\|\mathbf{J}(D_j;\boldsymbol{\beta}_2;h_j)-\mathbf{J}(D_j;\boldsymbol{\beta}_1;h_j)\big\|=O_p\big(R_j(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2)\big),
$$

where $R_j(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2)=n_jh_j+\sqrt{n_jh_j\log p}+n_j\|\boldsymbol{\beta}_2-\boldsymbol{\beta}_1\|_2+\sqrt{n_j\log p}\|\boldsymbol{\beta}_2-\boldsymbol{\beta}_1\|_2^{1/2}$.

Proof. Denote $r_{1i}=\mathbf{X}_i^\top(\boldsymbol{\beta}_1-\boldsymbol{\beta}_0)$, $r_{2i}=\mathbf{X}_i^\top(\boldsymbol{\beta}_2-\boldsymbol{\beta}_0)$. Without loss of generality, we assume that $r_{1i}\le r_{2i}$ (the same result can be obtained in other cases). By conditions **C1**–**C3** and a large enough constant C_3, we have

$$
\begin{aligned}
&\mathrm{E}\big[\boldsymbol{\nu}_k^\top\big\{\mathbf{J}(D_j;\boldsymbol{\beta}_2;h_j)-\mathbf{J}(D_j;\boldsymbol{\beta}_1;h_j)\big\}\boldsymbol{\nu}_k\big]\\
&\quad=\sum_{i\in D_j}\mathrm{E}\big[\big(\boldsymbol{\nu}_k^\top\mathbf{X}_i\big)^2\big\{\tilde{\ell}''_{\tau,h_j}(\varepsilon_i-r_{1i})-\tilde{\ell}''_{\tau,h_j}(\varepsilon_i-r_{2i})\big\}\big]\\
&\quad\le\Lambda_{\max}(\Sigma)\sum_{i\in D_j}\mathrm{E}\big|\tilde{\ell}''_{\tau,h_j}(\varepsilon_i-r_{1i})-\tilde{\ell}''_{\tau,h_j}(\varepsilon_i-r_{2i})\big|\\
&\quad\le\Lambda_{\max}(\Sigma)\sum_{i\in D_j}\mathrm{E}\big\{\mathrm{I}(-\tau-h_j+r_{1i}\le\varepsilon_i\le-\tau+h_j+r_{2i})\\
&\qquad+\mathrm{I}(\tau-h_j+r_{1i}\le\varepsilon_i\le\tau+h_j+r_{2i})\big\}\\
&\quad\le C_3n_j(2h_j+\|\boldsymbol{\beta}_2-\boldsymbol{\beta}_1\|_2).
\end{aligned}
$$

By using the method of Lemma 2, we can prove this lemma. □

Proof of Theorem 2.1. Define a function

$$
\mathbf{G}_b(\boldsymbol{\beta})=\frac{1}{N_b}\sum_{j=1}^{b-1}\mathbf{J}(D_j;\hat{\boldsymbol{\beta}}_j;h_j)(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}_{b-1})+\frac{1}{N_b}\mathbf{U}(D_b;\boldsymbol{\beta}).\tag{A.1}
$$

Thus,

$$
\mathbf{G}_b(\hat{\boldsymbol{\beta}}_b)-\mathbf{G}_b(\boldsymbol{\beta}_0)=\frac{1}{N_b}\sum_{j=1}^{b-1}\mathbf{J}(D_j;\hat{\boldsymbol{\beta}}_j;h_j)(\hat{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)+\frac{1}{N_b}\big\{\mathbf{U}(D_b;\hat{\boldsymbol{\beta}}_b)-\mathbf{U}(D_b;\boldsymbol{\beta}_0)\big\}.\tag{A.2}
$$

Considering the second term in (A.2), By Lemma 1, we have

$$\begin{aligned}
&\mathbf{U}(D_b;\hat{\boldsymbol{\beta}}_b)-\mathbf{U}(D_b;\boldsymbol{\beta}_0)\\
&\quad=\tilde{\mathbf{U}}(D_b;\hat{\boldsymbol{\beta}}_b;h_b)-\tilde{\mathbf{U}}(D_b;\boldsymbol{\beta}_0;h_b)+O_p(n_bh_b\sqrt{p}\log p)\\
&\quad=\mathbf{J}(D_b;\boldsymbol{\beta}_0;h_b)(\hat{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)+\{\mathbf{J}(D_b;\bar{\boldsymbol{\beta}};h_b)-\mathbf{J}(D_b;\boldsymbol{\beta}_0;h_b)\}(\hat{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)\\
&\qquad+O_p(n_bh_b\sqrt{p}\log p),
\end{aligned}\tag{A.3}$$

where $\bar{\boldsymbol{\beta}}$ lies in between $\hat{\boldsymbol{\beta}}_b$ and $\boldsymbol{\beta}_0$. By Lemma 3, we can obtain

$$\big\|\mathbf{J}(D_b;\bar{\boldsymbol{\beta}};h_b)-\mathbf{J}(D_b;\boldsymbol{\beta}_0;h_b)\big\|=O_p\big(R_b(\hat{\boldsymbol{\beta}}_b,\boldsymbol{\beta}_0)\big).\tag{A.4}$$

Using equation (A.4), we can rewrite (A.3) as

$$\mathbf{U}(D_b;\hat{\boldsymbol{\beta}}_b)-\mathbf{U}(D_b;\boldsymbol{\beta}_0)=\mathbf{J}(D_b;\boldsymbol{\beta}_0;h_b)(\hat{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)+O_p(\hat{R}_b),\tag{A.5}$$

where $\hat{R}_j=n_jh_j\sqrt{p}\log p+R_j(\hat{\boldsymbol{\beta}}_j,\boldsymbol{\beta}_0)\|\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_0\|_2$ for $j=1,\ldots,b$. Combining equations (A.2) and (A.5) yields

$$\begin{aligned}
&\mathbf{G}_b(\hat{\boldsymbol{\beta}}_b)-\mathbf{G}_b(\boldsymbol{\beta}_0)\\
&\quad=\frac{1}{N_b}\left\{\sum_{j=1}^{b-1}\mathbf{J}(D_j;\hat{\boldsymbol{\beta}}_j;h_j)+\mathbf{J}(D_b;\boldsymbol{\beta}_0;h_b)\right\}(\hat{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)+\frac{1}{N_b}O_p(\hat{R}_b).
\end{aligned}\tag{A.6}$$

According to equation (2.8), the renewable estimator $\hat{\boldsymbol{\beta}}_b$ satisfies

$$\mathbf{G}_b(\hat{\boldsymbol{\beta}}_b)=\mathbf{0}.\tag{A.7}$$

From equations (A.1), (A.6) and (A.7), we know that

$$\begin{aligned}
\mathbf{G}_b(\boldsymbol{\beta}_0)&=\frac{1}{N_b}\sum_{j=1}^{b-1}\mathbf{J}(D_j;\hat{\boldsymbol{\beta}}_j;h_j)(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_{b-1})+\frac{1}{N_b}\mathbf{U}(D_b;\boldsymbol{\beta}_0)\\
&=\frac{1}{N_b}\left\{\sum_{j=1}^{b-1}\mathbf{J}(D_j;\hat{\boldsymbol{\beta}}_j;h_j)+\mathbf{J}(D_b;\boldsymbol{\beta}_0;h_b)\right\}(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_b)+\frac{1}{N_b}O_p(\hat{R}_b).
\end{aligned}$$

It follows that

$$\begin{aligned}
&-\frac{1}{N_b}\left\{\sum_{j=1}^{b-1}\mathbf{J}(D_j;\hat{\boldsymbol{\beta}}_j;h_j)+\mathbf{J}(D_b;\boldsymbol{\beta}_0;h_b)\right\}(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_b)\\
&\quad+\frac{1}{N_b}\sum_{j=1}^{b-1}\mathbf{J}(D_j;\hat{\boldsymbol{\beta}}_j;h_j)(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_{b-1})\\
&\quad+\frac{1}{N_b}\mathbf{U}(D_b;\boldsymbol{\beta}_0)+\frac{1}{N_b}O_p(\hat{R}_b)=\mathbf{0}.
\end{aligned}\tag{A.8}$$

By $\mathbf{U}(D_1;\hat{\boldsymbol{\beta}}_1)=\mathbf{0}$ and Lemmas 1 and 3, we have

$$\begin{aligned}\mathbf{U}(D_1;\boldsymbol{\beta}_0) &= \tilde{\mathbf{U}}(D_1;\boldsymbol{\beta}_0;h_1)+O_p(n_1h_1\sqrt{p}\log p)\\ &= \tilde{\mathbf{U}}(D_1;\hat{\boldsymbol{\beta}}_1;h_1)+\mathbf{J}(D_1;\hat{\boldsymbol{\beta}}_1;h_1)(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_1)+O_p(\hat{R}_1)\\ &= \mathbf{U}(D_1;\hat{\boldsymbol{\beta}}_1)+\mathbf{J}(D_1;\hat{\boldsymbol{\beta}}_1;h_1)(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_1)+O_p(\hat{R}_1)\\ &= \mathbf{J}(D_1;\hat{\boldsymbol{\beta}}_1;h_1)(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_1)+O_p(\hat{R}_1).\end{aligned} \tag{A.9}$$

By (2.7), we can obtain

$$\begin{aligned}\mathbf{U}(D_2;\boldsymbol{\beta}_0) &= \tilde{\mathbf{U}}(D_2;\boldsymbol{\beta}_0;h_2)+O_p(n_2h_2\sqrt{p}\log p)\\ &= \tilde{\mathbf{U}}(D_2;\hat{\boldsymbol{\beta}}_2;h_2)+\mathbf{J}(D_2;\hat{\boldsymbol{\beta}}_2;h_2)(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_2)+O_p(\hat{R}_2)\\ &= \mathbf{U}(D_2;\hat{\boldsymbol{\beta}}_2)+\mathbf{J}(D_2;\hat{\boldsymbol{\beta}}_2;h_2)(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_2)+O_p(\hat{R}_2)\\ &= -\mathbf{J}(D_1;\hat{\boldsymbol{\beta}}_1;h_1)(\hat{\boldsymbol{\beta}}_2-\hat{\boldsymbol{\beta}}_1)+\mathbf{J}(D_2;\hat{\boldsymbol{\beta}}_2;h_2)(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_2)+O_p(\hat{R}_2).\end{aligned} \tag{A.10}$$

Thus, combining (A.9) and (A.10),

$$\begin{aligned}&\mathbf{U}(D_1;\boldsymbol{\beta}_0)+\mathbf{U}(D_2;\boldsymbol{\beta}_0)\\ &\quad=\{\mathbf{J}(D_1;\hat{\boldsymbol{\beta}}_1;h_1)+\mathbf{J}(D_2;\hat{\boldsymbol{\beta}}_2;h_2)\}(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_2)+\sum_{j=1}^{2}O_p(\hat{R}_j).\end{aligned} \tag{A.11}$$

Similarly to equation (A.11), at the $(b-1)$-th data batch, it is easy to shown that

$$\sum_{j=1}^{b-1}\mathbf{U}(D_j;\boldsymbol{\beta}_0)=\sum_{j=1}^{b-1}\mathbf{J}(D_j;\hat{\boldsymbol{\beta}}_j;h_j)(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_{b-1})+\sum_{j=1}^{b-1}O_p(\hat{R}_j). \tag{A.12}$$

Plugging equation (A.12) into equation (A.8), we get

$$\begin{aligned}&-\frac{1}{N_b}\left\{\sum_{j=1}^{b-1}\mathbf{J}(D_j;\hat{\boldsymbol{\beta}}_j;h_j)+\mathbf{J}(D_b;\boldsymbol{\beta}_0;h_b)\right\}(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_b)\\ &\quad+\frac{1}{N_b}\sum_{j=1}^{b}\mathbf{U}(D_j;\boldsymbol{\beta}_0)+\frac{1}{N_b}\sum_{j=1}^{b}O_p(\hat{R}_j)=\mathbf{0}.\end{aligned} \tag{A.13}$$

Under condition $N_1\to\infty$, by Corollary 2.1 in He and Shao (2000), $\hat{\boldsymbol{\beta}}_1$ is $\sqrt{p/N_1}$-consistent. If $\{\hat{\boldsymbol{\beta}}_j\}_{j=1}^{b-1}$ are $\sqrt{p/N_j}$-consistent, by Lemma 4 and conditions $h_j=O(N_j^{-1/2}(\log p)^{-1})$ for $j=1,\ldots,b$ hold, we have

$$\frac{1}{N_b}\sum_{j=1}^{b}O_p(\hat{R}_j)=\frac{1}{N_b}O_p\left(\sum_{j=1}^{b}n_jh_j\sqrt{p}\log p\right)=\frac{\sqrt{p}}{N_b}O_p\left(\sum_{j=1}^{b}\frac{n_j}{\sqrt{N_j}}\right)=O_p\left(\sqrt{\frac{p}{N_b}}\right), \tag{A.14}$$

where the last equation is based on $\sum_{j=1}^{b} n_j/\sqrt{N_j} \le 2\sqrt{N_b}$ by Lemma 3 in Han et al. (2021).

It is easy to prove that $N_b^{-1}\{\sum_{j=1}^{b-1}\mathbf{J}(D_j;\hat{\boldsymbol{\beta}}_j;h_j)+\mathbf{J}(D_b;\boldsymbol{\beta}_0;h_b)\} = O_p(1)$. By (A.9) in Zhou et al. (2018), we can obtain

$$\left\|\frac{1}{N_b}\sum_{j=1}^{b}\mathbf{U}(D_j;\boldsymbol{\beta}_0)\right\|_2 = O_p(\sqrt{p/N_b}). \tag{A.15}$$

Then, equations (A.13)–(A.15), we can obtain $\|\hat{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|_2 = O_p(\sqrt{p/N_b})$. □

Proof of Theorem 2.2. By Theorem 2.1 and Lemmas 2 and 3, we can obtain

$$\begin{aligned}\mathbf{J}(D_j;\hat{\boldsymbol{\beta}}_j;h_j) &= \mathbf{J}(D_j;\boldsymbol{\beta}_0;h_j)+O_p\big(R_j(\hat{\boldsymbol{\beta}}_j,\boldsymbol{\beta}_0)\big)\\ &= -\sum_{i\in D_j}\mathbf{X}_i\mathbf{X}_i^{\top}\ell_\tau''(\varepsilon_i)+O_p\big(R_j(\hat{\boldsymbol{\beta}}_j,\boldsymbol{\beta}_0)\big).\end{aligned}$$

Then, we have

$$\begin{aligned}&\frac{1}{N_b}\left\{\sum_{j=1}^{b-1}\mathbf{J}(D_j;\hat{\boldsymbol{\beta}}_j;h_j)+\mathbf{J}(D_b;\boldsymbol{\beta}_0;h_b)\right\}\\ &\quad= -\frac{1}{N_b}\sum_{i\in N_b}\mathbf{X}_i\mathbf{X}_i^{\top}\ell_\tau''(\varepsilon_i)+\frac{1}{N_b}\sum_{j=1}^{b}O_p\big(R_j(\hat{\boldsymbol{\beta}}_j,\boldsymbol{\beta}_0)\big)\\ &\quad= -\boldsymbol{\Sigma}\mathrm{E}\big[\ell_\tau''(\varepsilon)\big]+o_p(1).\end{aligned} \tag{A.16}$$

Plugging (A.16) into equation (A.13), and by conditions $p = o(\min\{N_1,\sqrt{N_b/\log N_b}\})$ and $h_j = o((pN_j)^{-1/2}(\log p)^{-1})$, we have

$$\sqrt{N_b}\mathrm{E}\big[\ell_\tau''(\varepsilon)\big]\boldsymbol{\Sigma}(\hat{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0) = \frac{1}{\sqrt{N_b}}\sum_{j=1}^{b}\mathbf{U}(D_j;\boldsymbol{\beta}_0)+o_p(1).$$

Thus, by the central limit theorem, we prove the theorem. □

Proof of Theorem 3.1. By equation (3.2) and the proof of Lemma 1, we have

$$\begin{aligned}\mathbf{e}_k^{\top}\mathbf{U}(D_1;\boldsymbol{\beta}_0) &= \mathbf{e}_k^{\top}\tilde{\mathbf{U}}(D_1;\boldsymbol{\beta}_0;h_1)+O_p(n_1h_1\log p)\\ &= \mathbf{e}_k^{\top}\tilde{\mathbf{U}}(D_1;\tilde{\boldsymbol{\beta}}_1;h_1)+\mathbf{e}_k^{\top}\mathbf{J}(D_1;\tilde{\boldsymbol{\beta}}_1;h_1)(\boldsymbol{\beta}_0-\tilde{\boldsymbol{\beta}}_1)+O_p(\tilde{R}_1)\\ &= \mathbf{e}_k^{\top}\mathbf{U}(D_1;\tilde{\boldsymbol{\beta}}_1)+\mathbf{e}_k^{\top}\mathbf{J}(D_1;\tilde{\boldsymbol{\beta}}_1;h_1)(\boldsymbol{\beta}_0-\tilde{\boldsymbol{\beta}}_1)+O_p(\tilde{R}_1)\\ &= \mathbf{e}_k^{\top}\mathbf{J}(D_1;\tilde{\boldsymbol{\beta}}_1;h_1)(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_1)+\mathbf{e}_k^{\top}N_1\lambda_1\,\mathbf{sgn}(\tilde{\boldsymbol{\beta}}_1)+O_p(\tilde{R}_1),\end{aligned} \tag{A.17}$$

where $\tilde{R}_j = n_jh_j\log p + R_j(\tilde{\boldsymbol{\beta}}_j,\boldsymbol{\beta}_0)\|\tilde{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_0\|_2$. By (3.5), we can obtain

$$\mathbf{e}_k^{\top}\mathbf{U}(D_2;\boldsymbol{\beta}_0) = \mathbf{e}_k^{\top}\mathbf{U}(D_2;\tilde{\boldsymbol{\beta}}_2)+\mathbf{e}_k^{\top}\mathbf{J}(D_2;\tilde{\boldsymbol{\beta}}_2;h_2)(\boldsymbol{\beta}_0-\tilde{\boldsymbol{\beta}}_2)+O_p(\tilde{R}_2)$$

$$= -\mathbf{e}_k^\top \mathbf{J}(D_1; \tilde{\boldsymbol{\beta}}_1; h_1)(\tilde{\boldsymbol{\beta}}_2 - \tilde{\boldsymbol{\beta}}_1) + \mathbf{e}_k^\top \mathbf{J}(D_2; \tilde{\boldsymbol{\beta}}_2; h_2)(\boldsymbol{\beta}_0 - \tilde{\boldsymbol{\beta}}_2) + \mathbf{e}_k^\top N_2 \lambda_2 \operatorname{\mathbf{sgn}}(\tilde{\boldsymbol{\beta}}_2) - \mathbf{e}_k^\top N_1 \lambda_1 \operatorname{\mathbf{sgn}}(\tilde{\boldsymbol{\beta}}_1) + O_p(\tilde{R}_2). \quad \text{(A.18)}$$

Thus, combining (A.17) and (A.18),

$$\mathbf{e}_k^\top \mathbf{U}(D_1; \boldsymbol{\beta}_0) + \mathbf{e}_k^\top \mathbf{U}(D_2; \boldsymbol{\beta}_0) = \mathbf{e}_k^\top \{\mathbf{J}(D_1; \tilde{\boldsymbol{\beta}}_1; h_1) + \mathbf{J}(D_2; \tilde{\boldsymbol{\beta}}_2; h_2)\}(\boldsymbol{\beta}_0 - \tilde{\boldsymbol{\beta}}_2) + \mathbf{e}_k^\top N_2 \lambda_2 \operatorname{\mathbf{sgn}}(\tilde{\boldsymbol{\beta}}_2) + \sum_{j=1}^{2} O_p(\tilde{R}_j). \quad \text{(A.19)}$$

Under conditions in Theorem 3.1, we have $\|\tilde{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_0\|_2 = O_p(\sqrt{s \log p / N_1})$ by Loh (2021). If $\|\tilde{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_0\|_2 = O_p(\sqrt{s \log p / N_j})$ for $j = 1, \ldots, b-1$, then by (A.19) and condition $h_j = o((N_j \log p)^{-1/2})$, at the $(b-1)$-th data batch, it is easy to shown that

$$\sum_{j=1}^{b-1} \mathbf{e}_k^\top \mathbf{U}(D_j; \boldsymbol{\beta}_0) = \sum_{j=1}^{b-1} \mathbf{e}_k^\top \mathbf{J}(D_j; \tilde{\boldsymbol{\beta}}_j; h_j)(\boldsymbol{\beta}_0 - \tilde{\boldsymbol{\beta}}_{b-1}) + \mathbf{e}_k^\top N_{b-1} \lambda_{b-1} \operatorname{\mathbf{sgn}}(\tilde{\boldsymbol{\beta}}_{b-1}) + \sum_{j=1}^{b-1} O_p\big(n_j h_j \log p + s n_j \log p / N_j + (s \log p / N_j)^{3/4} \sqrt{n_j \log p}\big). \quad \text{(A.20)}$$

Plugging equation (A.20) into equation (3.9), we get

$$\begin{aligned} \mathbf{e}_k^\top \mathbf{H}_b'(\boldsymbol{\beta}_0) &= -\frac{1}{N_b} \mathbf{e}_k^\top \mathbf{U}(D_b, \boldsymbol{\beta}_0) - \frac{1}{N_b} \mathbf{e}_k^\top \tilde{\mathbf{J}}_{b-1}(\boldsymbol{\beta}_0 - \tilde{\boldsymbol{\beta}}_{b-1}) \\ &\quad - \frac{N_{b-1}}{N_b} \lambda_{b-1} \mathbf{e}_k^\top \operatorname{\mathbf{sgn}}(\tilde{\boldsymbol{\beta}}_{b-1}) \\ &= -\frac{1}{N_b} \sum_{j=1}^{b} \mathbf{e}_k^\top \mathbf{U}(D_j; \boldsymbol{\beta}_0) \\ &\quad + \frac{1}{N_b} \sum_{j=1}^{b-1} O_p\big(n_j h_j \log p + s n_j \log p / N_j + (s \log p / N_j)^{3/4} \sqrt{n_j \log p}\big). \end{aligned} \quad \text{(A.21)}$$

By conditions $s \log N_b \sqrt{\log p / N_b} \to 0$ and $h_j = o((N_j \log p)^{-1/2})$ for $j = 1, \ldots, b$, (A.21) can be rewritten as

$$\mathbf{e}_k^\top \mathbf{H}_b'(\boldsymbol{\beta}_0) = -\frac{1}{N_b} \sum_{j=1}^{b} \mathbf{e}_k^\top \mathbf{U}(D_j; \boldsymbol{\beta}_0) + o_p(\sqrt{\log p / N_b}). \quad \text{(A.22)}$$

Similar to the proof (B.1) in Loh (2021), for large enough constant $\bar{C}$, we can obtain

$$\left\|\mathbf{H}_b'(\boldsymbol{\beta}_0)\right\|_\infty \leq \lambda_b/2, \tag{A.23}$$

with probability tending to one. By equations (3.8) and (3.9), we have

$$\begin{aligned}
&\mathbf{H}_b(\tilde{\boldsymbol{\beta}}_b) - \mathbf{H}_b(\boldsymbol{\beta}_0) - (\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0)^\top \mathbf{H}_b'(\boldsymbol{\beta}_0) \\
&\quad = \frac{1}{N_b}\sum_{i\in D_b} \ell_\tau\big(Y_i - \mathbf{X}_i^\top \tilde{\boldsymbol{\beta}}_b\big) - \frac{1}{N_b}\sum_{i\in D_b} \ell_\tau\big(Y_i - \mathbf{X}_i^\top \boldsymbol{\beta}_0\big) \\
&\qquad + \frac{1}{N_b}(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0)^\top \mathbf{U}(D_b;\boldsymbol{\beta}_0) - \frac{1}{2N_b}(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0)^\top \tilde{\mathbf{J}}_{b-1}(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0).
\end{aligned} \tag{A.24}$$

Based on equation (3.7), we have the basic inequality

$$\mathbf{H}_b(\tilde{\boldsymbol{\beta}}_b) + \lambda_b\|\tilde{\boldsymbol{\beta}}_b\|_1 \leq \mathbf{H}_b(\boldsymbol{\beta}_0) + \lambda_b\|\boldsymbol{\beta}_0\|_1. \tag{A.25}$$

Hence, by (A.24) and (A.25), and the convexity of $\ell_\tau(\cdot)$, we can obtain

$$(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0)^\top \mathbf{H}_b'(\boldsymbol{\beta}_0) \leq \mathbf{H}_b(\tilde{\boldsymbol{\beta}}_b) - \mathbf{H}_b(\boldsymbol{\beta}_0) \leq \lambda_b(\|\boldsymbol{\beta}_0\|_1 - \|\tilde{\boldsymbol{\beta}}_b\|_1).$$

Therefore, by (A.23), with probability tending to one, we have

$$\begin{aligned}
0 &\leq \lambda_b(\|\boldsymbol{\beta}_0\|_1 - \|\tilde{\boldsymbol{\beta}}_b\|_1) + \left\|H_b'(\boldsymbol{\beta}_0)\right\|_\infty \|\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0\|_1 \\
&\leq \lambda_b\bigg(\|\boldsymbol{\beta}_0\|_1 - \|\tilde{\boldsymbol{\beta}}_b\|_1 + \frac{1}{2}\|\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0\|_1\bigg).
\end{aligned} \tag{A.26}$$

Since

$$\begin{aligned}
\|\boldsymbol{\beta}_0\|_1 - \|\tilde{\boldsymbol{\beta}}_b\|_1 &= \|\boldsymbol{\beta}_{0,S}\|_1 - \|\tilde{\boldsymbol{\beta}}_{b,S}\|_1 - \|\tilde{\boldsymbol{\beta}}_{b,S^c}\|_1 \\
&\leq \|(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0)_S\|_1 - \|(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0)_{S^c}\|_1.
\end{aligned} \tag{A.27}$$

Combing (A.26) and (A.27),

$$\|(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0)_{S^c}\|_1 \leq 3\|(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0)_S\|_1, \tag{A.28}$$

with probability tending to one. Based on (A.28) and by the proof (B.2) in Loh (2021), we can obtain

$$\begin{aligned}
&\frac{1}{n_b}\sum_{i\in D_b} \ell_\tau\big(Y_i - \mathbf{X}_i^\top \tilde{\boldsymbol{\beta}}_b\big) - \frac{1}{n_b}\sum_{i\in D_b} \ell_\tau\big(Y_i - \mathbf{X}_i^\top \boldsymbol{\beta}_0\big) + \frac{1}{n_b}(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0)^\top \mathbf{U}(D_b;\boldsymbol{\beta}_0) \\
&\quad \geq \frac{1}{4}\Lambda_{\min}(\Sigma)\|\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0\|_2^2,
\end{aligned} \tag{A.29}$$

with probability tending to one. Note that the last term in (A.24), with probability tending to one, we have

$$-(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0)^\top \tilde{\mathbf{J}}_{b-1}(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) = \sum_{j=1}^{b-1}\sum_{i\in D_j} \big\{\mathbf{X}_i^\top(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0)\big\}^2 \tilde{\ell}''_{\tau,h_j}\big(Y_i - \mathbf{X}_i^\top \tilde{\boldsymbol{\beta}}_j\big)$$

$$\geq \sum_{j=1}^{b-1}\sum_{i\in D_j}\left\{\mathbf{X}_i^\top(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)\right\}^2\mathrm{I}\big(|Y_i-\mathbf{X}_i^\top\tilde{\boldsymbol{\beta}}_j|\leq\tau-h_j\big)$$
$$\geq \sum_{j=1}^{b-1}\sum_{i\in D_j}\left\{\mathbf{X}_i^\top(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)\right\}^2\mathrm{I}\big(|Y_i-\mathbf{X}_i^\top\tilde{\boldsymbol{\beta}}_j|\leq\tau/2\big)$$
$$\geq \frac{\sigma^2}{\tau^2}N_{b-1}\Lambda_{\min}(\Sigma)\|\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|_2^2, \tag{A.30}$$

where $\sigma^2 = Var(\boldsymbol{\varepsilon})$ and the last inequality be proved by using the inequality (B.15) in Loh (2021). By (A.24), (A.29) and (A.30), with probability tending to one, we have

$$\mathbf{H}_b(\tilde{\boldsymbol{\beta}}_b)-\mathbf{H}_b(\boldsymbol{\beta}_0)-(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)^\top\mathbf{H}_b'(\boldsymbol{\beta}_0)\geq C_4\|\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|_2^2, \tag{A.31}$$

where $C_4 = \min\{1/4,\sigma^2/(2\tau^2)\}\Lambda_{\min}(\Sigma)$. Therefore, the (A.31) together with the basic inequality (A.25) implies that

$$(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)^\top\mathbf{H}_b'(\boldsymbol{\beta}_0)+C_4\|\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|_2^2\leq\mathbf{H}_b(\tilde{\boldsymbol{\beta}}_b)-\mathbf{H}_b(\boldsymbol{\beta}_0)\leq\lambda_b(\|\boldsymbol{\beta}_0\|_1-\|\tilde{\boldsymbol{\beta}}_b\|_1), \tag{A.32}$$

so combing with inequalities (A.23) and (A.32), we can obtain

$$\begin{aligned}C_4\|\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|_2^2 &\leq \big\|\mathbf{H}_b'(\boldsymbol{\beta}_0)\big\|_\infty\|\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|_1+\lambda_b(\|\boldsymbol{\beta}_0\|_1-\|\tilde{\boldsymbol{\beta}}_b\|_1)\\ &\leq \lambda_b\left(\frac{1}{2}\|\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|_1+\big\|(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)_S\big\|_1-\big\|(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)_{S^c}\big\|_1\right)\\ &\leq \frac{3}{2}\lambda_b\big\|(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)_S\big\|_1\leq\frac{3}{2}\lambda_b\sqrt{s}\|\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|_2,\end{aligned}$$

implying that

$$\|\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|_2\leq\frac{3\lambda_b\sqrt{s}}{2C_4},$$

with probability tending to one. Therefore, we can obtain

$$\|\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|_2 = O_p(\sqrt{s\log p/N_b}).$$

This concludes the proof of the theorem. □

References

Barzilai, J. and Borwein, J. (1988). Two-point step size gradient methods. *IMA Journal of Numerical Analysis* **8** 141–148. https://doi.org/10.1093/imanum/8.1.141

Cai, T. and Liu, W. (2011). Adaptive thresholding for sparse covariance matrix estimation. *Journal of the American Statistical Association* **106** 672–684. https://doi.org/10.1198/jasa.2011.tm10560

Cai, T., Zhang, C. and Zhou, H. (2010). Optimal rates of convergence for covariance matrix estimation. *Annals of Statistics* **38** 2118–2144. https://doi.org/10.1214/09-AOS752

Chen, C. (2007). A finite smoothing algorithm for quantile regression. *Journal of Computational and Graphical Statistics* **16** 136–164. https://doi.org/10.1198/106186007X180336

Chen, X., Liu, W. and Zhang, Y. (2019). Quantile regression under memory constraint. *Annals of Statistics* **47** 3244–3273. https://doi.org/10.1214/18-AOS1777

Deshpande, Y., Javanmard, A. and Mehrabi, M. (2023). Online debiasing for adaptively collected high-dimensional data with applications to time series analysis. *Journal of the American Statistical Association* **118** 1126–1139. https://doi.org/10.1080/01621459.2021.1979011

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96** 1348–1360.

Fan, J., Li, Q. and Wang, Y. (2017). Estimation of high dimensional mean regression in the absence of symmetry and light tail assumptions. *Journal of the Royal Statistical Society: Series B* **79** 247–265. https://doi.org/10.1111/rssb.12166

Fan, J., Gong, W., Li, C. J. and Sun, Q. (2018a). Statistical sparse online regression: a diffusion approximation perspective. In: *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics* **84** 1017–1026.

Fan, J., Liu, H., Sun, Q. and Zhang, T. (2018b). I-LAMM for sparse learning: Simultaneous control of algorithmic complexity and statistical error. *Annals of Statistics* **96** 1348–1360. https://doi.org/10.1214/17-AOS1568

Frostig, R., Ge, R., Kakade, S. and Sidford, A. (2015). Competing with the empirical risk minimizer in a single pass. In: *Proceedings of The 28th Conference on Learning Theory* **40** 728–763.

Hampel, F., Hennig, C. and Ronchetti, E. (2011). A smoothing principle for the Huber and other location M-estimators. *Computational Statistics & Data Analysis* **55** 324–337. https://doi.org/10.1016/j.csda.2010.05.001

Han, R., Luo, L., Lin, Y. and Huang, J. (2021). Online debiased lasso for streaming data. arXiv:2106.05925v2.

Han, D., Huang, J., Lin, Y. and Shen, G. (2022a). Robust post-selection inference of high-dimensional mean regression with heavy-tailed asymmetric or heteroskedastic errors. *Journal of Econometrics* **230** 416–431. https://doi.org/10.1016/j.jeconom.2021.05.006

Han, D., Huang, J., Lin, Y. and Shen, G. (2022b). Robust post-selection inference of high-dimensional mean regression with heavy-tailed asymmetric or heteroskedastic errors. *Journal of Econometrics* **230** 416–431. https://doi.org/10.1016/j.jeconom.2021.05.006

Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press. https://doi.org/10.1017/CBO9780511811685

He, X. and Shao, Q. (2000). On parameters of increasing dimensions. *Journal of Multivariate Analysis* **73** 120–135. https://doi.org/10.1006/jmva.1999.1873

He, X., Pan, X., Tan, K. and Zhou, W. (2023). Smoothed quantile regression with large-scale inference. *Journal of Econometrics* **232** 367–388. https://doi.org/10.1016/j.jeconom.2021.07.010

Huber, P. J. (1973). Robust regression: asymptotics, conjectures and monte carlo. *Annals of Statistics* **1** 799–821. https://doi.org/10.1214/aos/1176342503

Huber, P. J. and Ronchetti, E. (2009). *Robust Statistics*, Second Edition. Wiley, New York. https://doi.org/10.1002/9780470434697

Jiang, R. and Yu, K. (2022). Renewable quantile regression for streaming data sets. *Neurocomputing* **508** 208–224.

Jiang, Y., Wang, Y., Fu, L. and Wang, X. (2019). Robust estimation using modifed Huber's functions with new tails. *Technometrics* **61** 111–122. https://doi.org/10.1080/00401706.2018.1470037

Lambert-Lacroix, S. and Zwald, L. (2011). Robust regression through the Huber's criterion and adaptive lasso penalty. *Electronic Journal of Statistics* **5** 1015–1053. https://doi.org/10.1214/11-EJS635

Loh, P. (2017). Statistical consistency and asymptotic normality for high-dimensional robust M-estimators. *Annals of Statistics* **45** 866–896. https://doi.org/10.1214/16-AOS1471

Loh, P. (2021). Scale calibration for high-dimensional robust regression. *Electronic Journal of Statistics* **15** 5933–5994. https://doi.org/10.1214/21-EJS1936

Luo, L. and Song, P. (2020). Renewable estimation and incremental inference in generalized linear models with streaming data sets. *Journal of the Royal Statistical Society: Series B* **82** 69–97. https://doi.org/10.1111/rssb.12352

Luo, J., Sun, Q. and Zhou, W. (2022). Distributed adaptive Huber regression. *Computational Statistics & Data Analysis* **169** 107419. https://doi.org/10.1016/j.csda.2021.107419

Luo, L., Zhou, L. and Song, P. (2022). Real-time regression analysis of streaming clustered data with possible abnormal data batches. *Journal of the American Statistical Association*. https://doi.org/10.1080/01621459.2022.2026778

Luo, L., Han, R., Lin, Y. and Huang, J. (2021). Statistical inference in high-dimensional generalized linear models with streaming data. arXiv:2018.04437.

Ma, X., Lin, L. and Gai, Y. (2023). A general framework of online updating variable selection for generalized linear models with streaming datasets. *Journal of Statistical Computation and Simulation* **93** 325–340.

Pan, X., Sun, Q. and Zhou, W. (2021). Iteratively reweighted l_1-penalized robust regression. *Electronic Journal of Statistics* **15** 3287–3348.

Quan, M. and Lin, Z. (2022). Optimal one-pass nonparametric estimation under memory constraint. *Journal of the American Statistical Association*.

https://doi.org/10.1080/01621459.2022.2115374

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics* **22** 400–407.

Schifano, E., Wu, J., Wang, C., Yan, J. and Chen, M. (2016). Online updating of statistical inference in the big data setting. *Technometrics* **58** 393–403. https://doi.org/10.1080/00401706.2016.1142900

Shi, C., Song, R., Lu, W. and Li, R. (2021). Statistical inference for high-dimensional models via recursive online-score estimation. *Journal of the American Statistical Association* **116** 1307–1318. https://doi.org/10.1080/01621459.2019.1710154

Sun, Q., Zhou, W. and Fan, J. (2020). Adaptive Huber regression. *Journal of the American Statistical Association* **115** 254–265. https://doi.org/10.1080/01621459.2018.1543124

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B* **58** 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Wang, H., Li, R. and Tsai, C.-L. (2007). Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika* **94** 553–568. https://doi.org/10.1093/biomet/asm053

Western, B. (1995). Concepts and suggestions for robust regression analysis. *American Journal of Political Science* **39** 758–764. https://doi.org/10.2307/2111654

Yang, Y. and Yao, F. (2022). Online estimation for functional data. *Journal of the American Statistical Association*. https://doi.org/10.1080/01621459.2021.2002158

Yohai, V. and Maronna, R. (1979). Asymptotic behavior of M-estimators for the linear model. *Annals of Statistics* **7** 258–268. https://doi.org/10.1214/aos/1176344610

Yu, B. (2020). p-Huber loss functions and its robustness. *Advances in Applied Mathematics* **9** 2283–2291. https://doi.org/10.12677/aam.2020.912267

Zheng, C. (2021). A new principle for tuning-free Huber regression. *Statistica Sinica* **31** 2153–2177. https://doi.org/10.5705/ss.202019.0045

Zhou, W., Bose, K., Fan, J. and Liu, H. (2018). A new perspective on robust M-estimation: finite sample theory and applications to dependence-adjusted multiple testing. *Annals of Statistics* **46** 1904–1931. https://doi.org/10.1214/17-AOS1606

Zou, H. (2006). The adaptive LASSO and its oracle properties. *Journal of the American Statistical Association* **101** 1418–1429. https://doi.org/10.1198/016214506000000735