Neurocomputing 508 (2022) 208–224

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Renewable quantile regression for streaming data sets

Rong Jiang [a], Keming Yu [b,c,*]

[a] Donghua University, Shanghai 201620, People's Republic of China
[b] Anqing Normal University, People's Republic of China
[c] Brunel University London, United Kingdom

ARTICLE INFO

Article history:
Received 1 October 2021
Revised 28 May 2022
Accepted 3 August 2022
Available online 13 August 2022
Communicated by Zidong Wang

2010 MSC::
60G08
62G20

Keywords:
Quantile regression
Streaming data
Variable selection
Online updating
Optimisation algorithm

ABSTRACT

Online updating is an important statistical method for the analysis of big data arriving in streams due to its ability to break the storage barrier and the computational barrier under certain circumstances. The quantile regression, as a widely used regression model in many fields, faces challenges in model fitting and variable selection with big data arriving in streams. Chen et al. (2019, Annals of Statistics) has proposed a quantile regression method for streaming data, but a strong additional condition is required. In this paper, renewable optimized objective functions for regression parameter estimation and variable selection in a quantile regression are proposed. The proposed methods are illustrated using current data and the summary statistics of historical data. Theoretically, the proposed statistics are shown to have the same asymptotic distributions as the standard version computed on an entire data stream with the data batches pooled into one data set, without additional condition. Both simulations and data analysis are conducted to illustrate the finite sample performance of the proposed methods.

1. Introduction

The concept of "big data" may have different meanings to people from different fields and has since become a dominant topic in nearly all academic disciplines and in applied fields. In a broad sense, big data are data on a massive scale in terms of volume, variety, velocity, variability and veracity [14]. Applying statistical models and methods to such big data can cause excessive computational burden not only in terms of strains on computer memory due to the large volume but also strains in terms of computational efficiency since even seemingly very simple tasks can take an inordinate amount of time to compute. In a recent review, [31] grouped the statistical and computational methodologies into three categories: subsampling-based approaches (e.g., [35,40; 34]), divide and conquer approaches (e.g., [23]; [18]; [1]; [16]; [17]), and online updating approaches (e.g., [27]; [24]; [28]; [26]). Online updating approaches are distinct from the other two because they target problem where data arrive in streams or large chunks and address statistical problems in an updating framework without storage requirements for previous data. Because the demand for stream processing is increasing ([4; 6] and among others), which makes online updating particularly appealing due to its ability which processes huge volume of data at speed so that organisations or businesses can react to changing conditions in real-time.

Our focus in this paper is a streaming data set that arrives in streams. Assume we have the streaming data set $\{D_1, \ldots, D_b\}$ 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 $N_b = \sum_{j=1}^{b} n_j$. In the era of big data, streaming data sets from various areas, such as bioinformatics, medical imaging and computer vision, are rapidly increasing in volume and velocity. This presents challenges to learning efficient statistical models and inferences. How to make statistical inferences without storage requirements for previous raw data is the key in the streaming data environment. When large amounts of data arrive in streams, online updating is an important method to alleviate both computational and data storage issues. In this framework for regression-type analyses, [27] developed online-updating algorithms for linear models and estimating equations. The estimation consistency of these methods has been established based on a strong regularity condition: the total number of streaming data sets b, needs to satisfy the order of $b = O\left(n_j^c\right)$, with

* Corresponding author.
E-mail addresses: jrtrying@dhu.edu.cn (R. Jiang), keming.yu@brunel.ac.uk (K. Yu).

https://doi.org/10.1016/j.neucom.2022.08.019
0925-2312/

$c < 1/3$ for all js, where n_j is the size of the j-th data batch. 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 $b \to \infty$. [24] proposed a renewable estimation for the generalized linear model, which overcame the above unnatural restriction. [22] introduced a general framework of renewable weighted sums for various online updating estimations. Regarding other references, [32] expanded the scope of the online updating method by accommodating the arrival of new predictor variables mid-way along the data stream. [37] developed an online updating method for survival analysis under Cox proportional hazards models, and [39] proposed an online updating-based test to evaluate the proportional hazards assumption. [21] developed an update estimator for a linear errors-in-variables model.

The quantile regression (QR) [19], which analyzes the conditional distribution of outcomes given a set of covariates and has been widely used in many fields, faces challenges in model fitting and variable selection with big data arriving in streams. However, the above methods for streaming data based on least squares and estimating equations will no longer be applicable. In this paper, we consider the following quantile regression:

$$q_\tau(\mathbf{Y}|\mathbf{X}=\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\beta}_0, \tag{1.1}$$

where $q_\tau(\mathbf{Y}|\mathbf{X}=\mathbf{x}) = \inf\{y : P(\mathbf{Y} \leqslant y|\mathbf{X}=\mathbf{x}) \geqslant \tau\}$, $\mathbf{X}$ is a random vector of p-dimensional covariates, and β_0 is a vector of unknown parameters of interest. Note that β_0 should truly be $\beta_0(\tau)$, and we omit the subscript τ for notational convenience.

If the data 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 the independent and identically distributed (i.i.d.) samples from $(\mathbf{Y}, \mathbf{X}) \in R \times R^p$. The standard quantile regression estimator [20] solves the following minimization problem:

$$\min_{\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^{n} \rho_\tau(Y_i - \mathbf{X}_i^\top \boldsymbol{\beta}), \tag{1.2}$$

where $\rho_\tau(r) = \tau r - rI(r<0)$ is the check loss function, and $I(\cdot)$ is the indicator function. Note that by using (1.2), we find that the above methods for streaming data based on the least squares and estimating equations are not suitable for the QR because the quantile regression estimator has no display expression like the least squares estimator and the loss function of the quantile regression is not differentiable, even though loss function needs to be second-order differentiable in the estimation equation.

To circumvent the nondifferentiability of the QR loss function, [15] proposed smoothing the indicator part of the check function via a kernel smoothing survival function. Recently, [2] applied Horowitz's smoothing quantile regression for streaming data. This is interesting; however, their method requires that the sample size of the j-th batch $n_j \to \infty$ and n_j be approximately n_{j-2}^2. This implies a very strong restriction. For example, $n_1 = 100, n_3 = 10^4, n_5 = 10^8$, $n_7 = 10^{16}, n_9 = 10^{32}, \cdots$. [36] also considered QR for streaming data. However, their method requires that $b/\sqrt{N_b} \to 0$, which means the number of batches b can not very large, and for achieving the same asymptotic covariance matrix as that of the estimator with full data, the covariates of each batch are homogeneous. The smoothing method proposed by [15] is smoother at the cost of convexity, which inevitably raises optimization issues. Generally, computing a global minimum of a nonconvex function is intractable, but convolution-type smoothing can be obtained [11], which yields a convex and twice differentiable loss function, a lower mean squared error than that of the estimator in [15] and a more accurate Bahadur-Kiefer representation than the standard QR estimator. In this paper, in contrast to [2], we propose a renewable method for quantile regression via a smoothing objective function. The proposed method does not require any specific condition as that in [11] as long as each batch sample size is sufficiently large ($n_j \to \infty, j = 1, \ldots, b$).

Variable selection plays an important role in the model building process. In practice, it is common to have a large number of candidate predictor variables available. A major challenge in regression analysis is to decide which predictors, among many potential predictors, are to be included in the model. Several methods, including the least absolute shrinkage and selection operator (LASSO) [30], smoothly clipped absolute deviation (SCAD) [8], adaptive LASSO [42], and minimax convex penalty (MCP) [41], have been proposed to select variables and estimate their regression coefficients simultaneously. Several algorithms were developed for variable selection in models with streaming data sets. [7] applied the truncated stochastic gradient descent (SGD) to a linear model. [29] introduced a novel framework, which combines updated statistics and truncation techniques, for variable selection in a linear model. These SGD algorithms and truncation techniques, however, are sensitive to the learning rate or step size and tend to select the set with larger cardinality to include all important variables. [5] considered a class of online estimators in a high-dimensional autoregressive model. [28] proposed an inference procedure for high-dimensional linear models via recursive online-score estimation. In both works, it is assumed that the entire data set is available at the initial stage for computing an initial estimator and that the information in the streaming data is used to reduce the bias of the initial estimator. However, the assumption that the full data set is available at the initial stage is not realistic for analyzing streaming data sets. [12] proposed an online debiased LASSO method for high-dimensional linear models with streaming data sets based on the least squares method. [25] studied a general framework for online updating variable selection in a generalized linear model with streaming data sets. In this paper, we also study a renewable variable selection method for quantile regression.

To summarize, we make the following important contributions to the existing literature. (1) We develop a renewable estimation and algorithm for the quantile regression that only requires the availability of the current data batch in the data stream and sufficient statistics of the historical data at each stage of the analysis. Theoretically, the proposed estimator achieves optimal efficiency and its asymptotic covariance matrix is the same as that of the estimator with full data without additional condition. (2) We study a renewable optimized objective function for variable selection in a quantile regression. The proposed method only requires the availability of the current data batch in the data stream and sufficient statistics of the historical data at each stage of the analysis. In order to realize a numerical solution, we introduce an efficient algorithm for this optimization problem. Moreover, the proposed method can choose tuning parameters via a data-driven and online updating BIC criterion. Theoretically, the proposed estimator achieves the same consistency and oracle properties as the estimator based on the entire data set under some general conditions. (3) The proposed renewable methods are all free of the constraint on the number of batches, which means that the new methods are adaptive to

Table 1
The differences between our proposed methods and the previous methods in the references.

Reference	Method	Additional conditions
[27]	Estimating equations	$b = O\left(n_j^c\right), c < 1/3$
[24]	Estimating equations	None
[2]	Quantile regression	n_j is approximately n_{j-2}^2
[36]	Quantile regression	$b/\sqrt{N_b} \to 0$
This paper	Quantile regression	None

the situation where streaming data sets arrive fast and perpetually. Finally, Table 1 shows the differences between our proposed methods and the previous methods in the references for linear model.

The remainder of this paper is organized as follows. In Section 2, the renewable smoothing QR estimator is proposed. The renewable variable selection 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 discussion in Section 5. All technical proofs are provided in the Appendix.

2. Renewable parameter estimation

2.1. Standard smoothing quantile regression

Suppose that the batches up to the b-th batch of streaming data can be pooled into one data set. Note that for a quantile regression, the loss function $\rho_\tau(r) = \tau r - rI(r < 0)$ is nondifferentiable. Therefore, the QR estimator has no display expression, so it is impossible to construct a renewable estimator for streaming data. To circumvent the nondifferentiability of the QR loss function, the QR estimator of β_0 in the model (1.1) can be solved by minimizing the following smoothing quantile regression (SQR) objective function [11]: $\hat{\boldsymbol{\beta}}^* = \arg\min_{\boldsymbol{\beta}} S_h(\boldsymbol{\beta})$ with

$$S_h(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\int_{-\infty}^{+\infty}\rho_\tau(t)K_h(t - Y_i + \mathbf{X}_i^\top\beta)dt, \tag{2.1}$$

where $K_h(\cdot) = K(\cdot/h)/h, K(\cdot)$ is a smooth kernel function and h is a bandwidth. Now, $S_h(\boldsymbol{\beta})$ is twice continuously differentiable with the gradient and Hessian matrix

$$\begin{aligned} S_h^{(1)}(\boldsymbol{\beta}) &\equiv \tfrac{\partial S_h(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}} = n^{-1}\sum_{i=1}^{n}\mathbf{X}_i\left\{\widetilde{K}\left((\mathbf{X}_i^\top\beta - Y_i)/h\right) - \tau\right\}, \\ S_h^{(2)}(\boldsymbol{\beta}) &\equiv \tfrac{\partial^2 S_h(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}^2} = n^{-1}\sum_{i=1}^{n}\mathbf{X}_i\mathbf{X}_i^\top K_h\left(\mathbf{X}_i^\top\beta - Y_i\right), \end{aligned} \tag{2.2}$$

respectively, where $\widetilde{K}(t) = \int_{-\infty}^{t}K(u)du$.

2.2. Smoothing quantile regression for streaming data sets

For model (1.1), $D_j = \{\mathbf{X}_j, \mathbf{Y}_j\}$ is the j-th batch data set, where $\mathbf{Y}_j = \left(Y_{1j}, \ldots, Y_{n_j j}\right)^\top$ and $\mathbf{X}_j = \left(\mathbf{X}_{1j}, \ldots, \mathbf{X}_{n_j j}\right)^\top$. We suppose that the $(\mathbf{X}_{ij}, Y_{ij})$ for all is and js are i.i.d. samples from $(\mathbf{Y}, \mathbf{X})$. 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 SQR $\hat{\boldsymbol{\beta}}_1$ (or $\hat{\boldsymbol{\beta}}_1^*$) by minimizing (2.1) to a renewed SQR $\hat{\boldsymbol{\beta}}_2^*$ without using any subject-level data but only some summary statistics from D_1. By (2.1) and (2.2), the SQR $\hat{\boldsymbol{\beta}}_1$ satisfies,

$$\frac{1}{N_1}U\left(D_1; \hat{\boldsymbol{\beta}}_1; h_1\right) = \mathbf{0}, \tag{2.3}$$

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

$$\frac{1}{N_2}U\left(D_1; \hat{\boldsymbol{\beta}}_2^*; h_2\right) + \frac{1}{N_2}U\left(D_2; \hat{\boldsymbol{\beta}}_2^*; h_2\right) = \mathbf{0}. \tag{2.4}$$

Solving Eq. (2.4) 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 derive a renewable estimate, by the (A.10) in the Appendix, we can obtain

$$U\left(D_1; \hat{\boldsymbol{\beta}}_2^*; h_2\right) = U\left(D_1; \hat{\boldsymbol{\beta}}_2^*; h_1\right) + O_p\left(n_1h_1^2 + n_1h_2^2\right), \tag{2.5}$$

where $O_p(\cdot)$ means bounded with probability. We take the first-order Taylor series expansion of the $U\left(D_1; \hat{\boldsymbol{\beta}}_2^*; h_1\right)$ around $\hat{\boldsymbol{\beta}}_1$,

$$U\left(D_1; \hat{\boldsymbol{\beta}}_2^*; h_1\right) = U\left(D_1; \hat{\boldsymbol{\beta}}_1; h_1\right) + J\left(D_1; \hat{\boldsymbol{\beta}}_1; h_1\right)\left(\hat{\boldsymbol{\beta}}_2^* - \hat{\boldsymbol{\beta}}_1\right) + O_p\left(n_1\|\hat{\boldsymbol{\beta}}_2^* - \hat{\boldsymbol{\beta}}_1\|_2^2\right), \tag{2.6}$$

where $J(D_j; \boldsymbol{\beta}; h) = \partial U(D_j; \boldsymbol{\beta}; h)/\partial\boldsymbol{\beta} = \sum_{i\in D_j}\mathbf{X}_i\mathbf{X}_i^\top K_h(\mathbf{X}_i^\top\beta - Y_i)$. By (2.3), (2.5) and (2.6), we have

$$\begin{aligned} U\left(D_1; \hat{\boldsymbol{\beta}}_2^*; h_2\right) =& J\left(D_1; \hat{\boldsymbol{\beta}}_1; h_1\right)\left(\hat{\boldsymbol{\beta}}_2^* - \hat{\boldsymbol{\beta}}_1\right) \\ &+ O_p\left(n_1\|\hat{\boldsymbol{\beta}}_2^* - \hat{\boldsymbol{\beta}}_1\|_2^2 + n_1h_1^2 + n_1h_2^2\right). \end{aligned} \tag{2.7}$$

By placing (2.7) into (2.4), we obtain

$$\begin{aligned} &\tfrac{1}{N_2}J\left(D_1; \hat{\boldsymbol{\beta}}_1; h_1\right)\left(\hat{\boldsymbol{\beta}}_2^* - \hat{\boldsymbol{\beta}}_1\right) + \tfrac{1}{N_2}U\left(D_2; \hat{\boldsymbol{\beta}}_2^*; h_2\right) \\ &+O_p\left(\tfrac{n_1}{N_2}\left\{\|\hat{\boldsymbol{\beta}}_2^* - \hat{\boldsymbol{\beta}}_1\|_2^2 + h_1^2 + h_2^2\right\}\right) = \mathbf{0}. \end{aligned} \tag{2.8}$$

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 bandwidths h_1 and h_2, the error term $O_p\left(\frac{n_1}{N_2}\left\{\|\hat{\boldsymbol{\beta}}_2^* - \hat{\boldsymbol{\beta}}_1\|_2^2 + h_1^2 + h_2^2\right\}\right)$ 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

$$\frac{1}{N_2}J\left(D_1; \hat{\boldsymbol{\beta}}_1; h_1\right)\left(\hat{\boldsymbol{\beta}}_2 - \hat{\boldsymbol{\beta}}_1\right) + \frac{1}{N_2}U\left(D_2; \hat{\boldsymbol{\beta}}_2; h_2\right) = \mathbf{0}. \tag{2.9}$$

Through Eq. (2.9), the initial $\hat{\boldsymbol{\beta}}_1$ is renewed by $\hat{\boldsymbol{\beta}}_2$ only using the historical summary statistics, including sample variance matrix $J\left(D_1; \hat{\boldsymbol{\beta}}_1; h_1\right)$ and estimate $\hat{\boldsymbol{\beta}}_1$, instead of the subject-level raw data D_1.

Generalizing the Eq. (2.9) 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:

$$\frac{1}{N_b}\sum_{j=1}^{b-1}J\left(D_j; \hat{\boldsymbol{\beta}}_j; h_j\right)\left(\hat{\boldsymbol{\beta}}_b - \hat{\boldsymbol{\beta}}_{b-1}\right) + \frac{1}{N_b}U\left(D_b; \hat{\boldsymbol{\beta}}_b; h_b\right) = \mathbf{0}. \tag{2.10}$$

2.3. Large sample properties

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

C1. The kernel function $K(\cdot)$ is even, integrable, and twice differentiable with bounded first and second derivatives such that $\int K(u)du = 1$ and $0 < \int_0^\infty K(u)\{1 - K(u)\}du < \infty$. In addition, $\int|u^2K(u)|du < \infty$, $\int uK(u)du = 0$ and $\int u^2K(u)du \neq 0$.

C2. The conditional density function of $\mathbf{Y}$ given $\mathbf{X} = \mathbf{x}, f(y|\mathbf{x})$ is bounded, continuous, strictly positive and $\lim_{y\to\pm\infty}f(y|\mathbf{x}) = 0$. The derivative $f'(\cdot|\cdot)$ is uniformly continuous in the sense that $\lim_{\delta\to 0}\sup_{(\mathbf{x},y)\in R^{p+1}}\sup_{t:|t|\leqslant\delta}|f'(y + t|\mathbf{x}) - f'(y|\mathbf{x})| = 0$,and such that $\sup_{(\mathbf{x},y)\in R^{p+1}}|f'(y|\mathbf{x})| < \infty$ and $\lim_{y\to\pm\infty}f'(y|\mathbf{x}) = 0$.

C3. The components of $\mathbf{X}$ are positive, bounded random variables, and $\boldsymbol{\Sigma} = E(\mathbf{X}\mathbf{X}^\top)$ is a positive definite matrix.

Remark 2.1. *Condition* **C1** *is a mild condition on $K(\cdot)$ for smoothing approximation. For example, for the Gaussian kernel $K(u) = (2\pi)^{-1/2}\exp(-u^2/2)$, it satisfies condition* **C1**. *Condition* **C2** *is a regular condition on the smoothness of the conditional density function $f(y|\mathbf{x})$. The conditions* **C2** *and* **C3** *ensure that Ω in Theorem*

2.1 *is positive definite, which means that* Ω^{-1} *exists. Conditions* **C1**-**C3** *are standard conditions, which are commonly used in smoothing quantile regression, as shown in [11].*

Theorem 2.1. *Assume that conditions* **C1**-**C3** *are satisfied. If* $h_j = o\left(N_j^{-1/4}\right)$, $h_j(N_j/\ln N_j)^{1/3} \to \infty$ *with* $N_j = \sum_{i=1}^{j} n_i$ *and* $n_i \to \infty, i = 1, \ldots, b$, *we have*

$$\sqrt{N_b}\left(\hat{\boldsymbol{\beta}}_b - \beta_0\right) \xrightarrow{L} \mathcal{N}\left(0, \tau(1-\tau)\Omega^{-1}\Sigma\Omega^{-1}\right),$$

where $\xrightarrow{L}$ *represents the convergence in the distribution and* $\Omega = E\{f(\mathbf{X}^\top\boldsymbol{\beta}_0|\mathbf{X})\mathbf{X}\mathbf{X}^\top\}$.

Through the result of Theorem 2.1, it is interesting to notice that the renewable estimator $\hat{\boldsymbol{\beta}}_b$ achieves optimal efficiency and its asymptotic covariance matrix is the same as that of the SQR estimator $\hat{\boldsymbol{\beta}}_b^*$ which is computed directly on all the samples, as shown in Theorem 5 in [11]. This implies that the proposed renewable estimator achieves the same asymptotic distribution as the SQR estimator.

2.4. Algorithm

Numerically, it is quite straightforward to find $\hat{\boldsymbol{\beta}}_b$ from (2.10) using the Newton–Raphson method at the $(r+1)$-th iteration:

$$\hat{\boldsymbol{\beta}}_b^{(r+1)} = \hat{\boldsymbol{\beta}}_b^{(r)} - \left\{\widehat{J}_{b-1} + J\left(D_b; \hat{\boldsymbol{\beta}}_b^{(r)}; h_b\right)\right\}^{-1} \widehat{U}_b^{(r)}, \tag{2.11}$$

where $\widehat{J}_{b-1} = \sum_{j=1}^{b-1} J\left(D_j; \hat{\boldsymbol{\beta}}_j; h_j\right)$ and $\widehat{U}_b^{(r)} = \widehat{J}_{b-1}\left(\hat{\boldsymbol{\beta}}_b^{(r)} - \hat{\boldsymbol{\beta}}_{b-1}\right) + U\left(D_b; \hat{\boldsymbol{\beta}}_b^{(r)}; h_b\right)$. When p is large, to speed up the calculation of (2.11), we may avoid updating $J\left(D_b; \hat{\boldsymbol{\beta}}_b^{(r)}; h_b\right)$ at each iteration. Replacing $\hat{\boldsymbol{\beta}}_b^{(r)}$ with $\hat{\boldsymbol{\beta}}_{b-1}$ leads to the following updating algorithm:

$$\hat{\boldsymbol{\beta}}_b^{(r+1)} = \hat{\boldsymbol{\beta}}_b^{(r)} - \left\{\widehat{J}_{b-1} + J\left(D_b; \hat{\boldsymbol{\beta}}_{b-1}; h_b\right)\right\}^{-1} \widehat{U}_b^{(r)}. \tag{2.12}$$

In algorithm (2.12), clearly we only use the subject-level data of current data D_b and summary statistics $\widehat{J}_{b-1}$ and $\hat{\boldsymbol{\beta}}_{b-1}$ from historical data batches up to $b-1$ rather than subject-level raw data of $\{D_1, \ldots, D_{b-1}\}$. Thus, our proposed renewable method is indeed an online estimation procedure. We summarize the general algorithm for the proposed renewable SQR method by (2.12) as follows.

Algorithm 1 Renewable SQR estimation for streaming data sets.

1: **Input:** streaming data sets $D_1, \ldots, D_b, \ldots$, the quantile level τ, kernel function $K(\cdot)$ and bandwidths h_b with $b = 1, 2 \ldots$
2: **Initialize:** calculate $\hat{\boldsymbol{\beta}}_1$ by minimizing (2.1) with D_1, and compute $J\left(D_1; \hat{\boldsymbol{\beta}}_1; h_1\right)$;
3: **for:** $b = 2, 3, \ldots$ **do**
4: read in data set D_b and compute $\widehat{J}_{b-1} + J\left(D_b; \hat{\boldsymbol{\beta}}_{b-1}; h_b\right)$;
5: select the initial estimator $\hat{\boldsymbol{\beta}}_b^{(0)} = \hat{\boldsymbol{\beta}}_{b-1}$ and run the following iterations until convergence:
$\hat{\boldsymbol{\beta}}_b^{(r+1)} = \hat{\boldsymbol{\beta}}_b^{(r)} - \left\{\widehat{J}_{b-1} + J\left(D_b; \hat{\boldsymbol{\beta}}_{b-1}; h_b\right)\right\}^{-1} \widehat{U}_b^{(r)}$; 6: update $\widehat{J}_b = \widehat{J}_{b-1} + J\left(D_b; \hat{\boldsymbol{\beta}}_b; h_b\right)$;
7: save $\hat{\boldsymbol{\beta}}_b$ and $\widehat{J}_b$ and release data set D_b from the memory;
8: **end**
9: **Output:** $\hat{\boldsymbol{\beta}}_b$ for $b = 2, 3, \ldots$

Note that in step 7 in Algorithm 1, we only need to save $\hat{\boldsymbol{\beta}}_b$ and $\widehat{J}_b$, 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 is assumed to be a fixed number in this paper, our method greatly reduces the amount of data storage.

3. Renewable variable selection

3.1. Variable selection based on all data

To avoid overfitting and improve the generalization ability, we first consider the penalized SQR (PSQR) based on all data (the batches up to the b-th batch of streaming data can be pooled into one data set):

$$\tilde{\boldsymbol{\beta}}^* = \arg\min_{\boldsymbol{\beta}} \{S_h(\boldsymbol{\beta}) + p_\lambda(|\boldsymbol{\beta}|)\}, \tag{3.1}$$

where $p_\lambda(\cdot)$ is a penalty function with regularization parameter λ. Among the various penalty functions, we consider the SCAD because of its properties of unbiasedness, sparsity and continuity. The SCAD penalty function is defined through its first-order derivative and symmetry around the origin. For any $\theta > 0$,

$$p'_\lambda(\theta) = \lambda\left\{I(\theta \leqslant \lambda) + \frac{(a\lambda - \theta)_+}{(a-1)\lambda} I(\theta > \lambda)\right\},$$

where $a > 2$. $a = 3.7$ was suggested by [8] from a Bayesian perspective and is commonly used in the variable selection literature.

3.2. Variable selection based on streaming data sets

We begin with two batches of data D_1 and D_2. We want to update the initial PSQR $\tilde{\boldsymbol{\beta}}_1$ (or $\tilde{\boldsymbol{\beta}}_1^*$) to a renewed PSQR $\tilde{\boldsymbol{\beta}}_2^*$ without using any subject-level data and using only some summary statistics from D_1.

$$\tilde{\boldsymbol{\beta}}_1 = \arg\min_{\boldsymbol{\beta}}\left\{S_{h_1}(D_1, \boldsymbol{\beta}) + p_{\lambda_1}(|\boldsymbol{\beta}|)\right\},$$

where $S_h(D_1, \boldsymbol{\beta})$ is the check function $S_h(\boldsymbol{\beta})$ in (2.1) with data D_1. Here, the PSQR $\tilde{\boldsymbol{\beta}}_1$ also satisfies

$$\frac{1}{N_1} U(D_1; \tilde{\boldsymbol{\beta}}_1; h_1) + p'_{\lambda_1}(|\tilde{\boldsymbol{\beta}}_1|) sign(\tilde{\boldsymbol{\beta}}_1) = 0, \tag{3.2}$$

where $sign(\cdot)$ is the sign function. Then, $\tilde{\boldsymbol{\beta}}_2^*$ satisfies the following aggregated score equation:

$$\tfrac{1}{N_2} U(D_1; \tilde{\boldsymbol{\beta}}_2^*; h_2) + \tfrac{1}{N_2} U(D_2; \tilde{\boldsymbol{\beta}}_2^*; h_2) + p'_{\lambda_2}(|\tilde{\boldsymbol{\beta}}_2^*|) sign(\tilde{\boldsymbol{\beta}}_2^*) = 0. \tag{3.3}$$

By similar analysis in Section 2.2 and (3.2), we can obtain

$$\begin{aligned} \tfrac{1}{N_2} U(D_1; \tilde{\boldsymbol{\beta}}_2^*; h_2) &= \tfrac{1}{N_2} U(D_1; \tilde{\boldsymbol{\beta}}_1; h_1) + \tfrac{1}{N_2} J(D_1; \tilde{\boldsymbol{\beta}}_1; h_1)(\tilde{\boldsymbol{\beta}}_2^* - \tilde{\boldsymbol{\beta}}_1) + \Re \\ &= -\tfrac{N_1}{N_2} p'_{\lambda_1}(|\tilde{\boldsymbol{\beta}}_1|) sign(\tilde{\boldsymbol{\beta}}_1) + \tfrac{1}{N_2} J(D_1; \tilde{\boldsymbol{\beta}}_1; h_1)(\tilde{\boldsymbol{\beta}}_2^* - \tilde{\boldsymbol{\beta}}_1) + \Re, \end{aligned} \tag{3.4}$$

where $\Re$ is an asymptotically ignored error term. By substituting (3.4) into (3.3), we have

$$\begin{aligned} &\tfrac{1}{N_2} J(D_1; \tilde{\boldsymbol{\beta}}_1; h_1)(\tilde{\boldsymbol{\beta}}_2^* - \tilde{\boldsymbol{\beta}}_1) + \tfrac{1}{N_2} U(D_2; \tilde{\boldsymbol{\beta}}_2^*; h_2) \\ &+ p'_{\lambda_2}(|\tilde{\boldsymbol{\beta}}_2^*|) sign(\tilde{\boldsymbol{\beta}}_2^*) - \tfrac{N_1}{N_2} p'_{\lambda_1}(|\tilde{\boldsymbol{\beta}}_1|) sign(\tilde{\boldsymbol{\beta}}_1) + \Re_3 = 0. \end{aligned}$$

Removing the asymptotically ignored term $\Re$, we propose a new estimator $\tilde{\boldsymbol{\beta}}_2$ as a solution to the equation of the form

$$\begin{aligned} &\tfrac{1}{N_2} J(D_1; \tilde{\boldsymbol{\beta}}_1; h_1)(\tilde{\boldsymbol{\beta}}_2 - \tilde{\boldsymbol{\beta}}_1) + \tfrac{1}{N_2} U(D_2; \tilde{\boldsymbol{\beta}}_2; h_2) \\ &+ p'_{\lambda_2}(|\tilde{\boldsymbol{\beta}}_2|) sign(\tilde{\boldsymbol{\beta}}_2) - \tfrac{N_1}{N_2} p'_{\lambda_1}(|\tilde{\boldsymbol{\beta}}_1|) sign(\tilde{\boldsymbol{\beta}}_1) = 0. \end{aligned} \tag{3.5}$$

Through Eq. (3.5), the initial $\tilde{\boldsymbol{\beta}}_1$ is renewed by $\tilde{\boldsymbol{\beta}}_2$ using statistics $J(D_1;\tilde{\boldsymbol{\beta}}_1;h_1),\tilde{\boldsymbol{\beta}}_1$ 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:

$$\begin{aligned}&\tfrac{1}{N_b}\sum_{j=1}^{b-1}J(D_j;\tilde{\boldsymbol{\beta}}_j;h_j)(\tilde{\boldsymbol{\beta}}_b-\tilde{\boldsymbol{\beta}}_{b-1})+\tfrac{1}{N_b}U(D_b;\tilde{\boldsymbol{\beta}}_b;h_b)\\&+p'_{\lambda_b}(|\tilde{\boldsymbol{\beta}}_b|)sign(\tilde{\boldsymbol{\beta}}_b)-\tfrac{N_{b-1}}{N_b}p'_{\lambda_{b-1}}(|\tilde{\boldsymbol{\beta}}_{b-1}|)sign(\tilde{\boldsymbol{\beta}}_{b-1})=0.\end{aligned}\tag{3.6}$$

Note that (3.6) is equal to

$$\begin{aligned}\tilde{\boldsymbol{\beta}}_b=\arg\min_{\boldsymbol{\beta}}\Big\{&\tfrac{1}{2N_b}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_{b-1})^\top\widetilde{J}_{b-1}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_{b-1})+\tfrac{1}{N_b}S_{h_b}(D_b;\boldsymbol{\beta})\\&-\tfrac{N_{b-1}}{N_b}p'_{\lambda_{b-1}}(|\tilde{\boldsymbol{\beta}}_{b-1}|)sign(\tilde{\boldsymbol{\beta}}_{b-1})(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_{b-1})+p_{\lambda_b}(|\boldsymbol{\beta}|)\Big\},\end{aligned}\tag{3.7}$$

where $\widetilde{J}_{b-1}=\sum_{j=1}^{b-1}J(D_j;\tilde{\boldsymbol{\beta}}_j;h_j)$ and $S_h(D_b;\boldsymbol{\beta})$ is the check function $S_h(\boldsymbol{\beta})$ in (2.1) with data D_b.

The following theorems show the consistency and oracle properties of the estimator $\tilde{\boldsymbol{\beta}}_b$ in (3.6) or (3.7).

Theorem 3.1. *(Consistency). Suppose that conditions in Theorem 2.1 hold and $\lambda_j\to 0$ with $j=1,\ldots,b$. Then,*

$$\|\tilde{\boldsymbol{\beta}}_b-\beta_0\|_2=O_p\left(N_b^{-1/2}\right).$$

Theorem 3.1 demonstrates that our estimator $\tilde{\boldsymbol{\beta}}_b$ is root-N_b consistent. The following theorem shows that $\tilde{\boldsymbol{\beta}}_b$ has the oracle property proposed in [8]. Without any loss of generality, we assume that the first s elements of β_0 are nonzero and the last $p-s$ elements are zero. That is, β_0 can be written as $\beta_0=(\beta_{01}^\top,\beta_{02}^\top)^\top$, where β_{01} is an s-dimensional vector of nonzero elements and $\beta_{02}=\mathbf{0}$ is a $(p-s)$-dimensional vector of zeros.

Theorem 3.2. *(Oracle property). Suppose that all conditions in Theorem 3.1 hold. If $\sqrt{N_b}\lambda_b\to\infty$, then with a probability tending to one, the root-N_b consistent local minimizer $\tilde{\boldsymbol{\beta}}_b=\left(\tilde{\boldsymbol{\beta}}_{(b1)}^\top,\tilde{\boldsymbol{\beta}}_{(b2)}^\top\right)^\top$ in Theorem 3.1 satisfies the following:*

(i) *Sparsity: $\tilde{\boldsymbol{\beta}}_{(b2)}=\mathbf{0}$, and*

(ii) *Asymptotic normality:*

$$\sqrt{N_b}(\tilde{\boldsymbol{\beta}}_{(b1)}-\beta_{01})\xrightarrow{L}\mathcal{N}\left(0,\tau(1-\tau)\Omega_{(11)}^{-1}\Sigma_{(11)}\Omega_{(11)}^{-1}\right),$$

where $\tilde{\boldsymbol{\beta}}_{(b1)}$ and $\tilde{\boldsymbol{\beta}}_{(b2)}$ are the first s and the last $p-s$ elements of $\tilde{\boldsymbol{\beta}}_b$, respectively. $\Omega_{(11)}$ and $\Sigma_{(11)}$ are the top-left s-by-s submatrix of Ω and Σ, respectively.

Theorems 3.1 and 3.2 show that the renewable estimator $\tilde{\boldsymbol{\beta}}_b$ in (3.7) achieves the same consistency and oracle property as the estimator $\tilde{\boldsymbol{\beta}}^*$ in (3.1), which is directly computed using all the samples ($n=N_b$), as shown in Lemmas 1 and 2 in the Appendix.

Remark 3.1. *The proposed renewable variable selection in (3.6) is different from the method in [25] for generalized linear models. Specifically, the theoretical derivations of the two methods are different, and there is no the term $N_{b-1}/N_bp'_{\lambda_{b-1}}(|\tilde{\boldsymbol{\beta}}_{b-1}|)sign(\tilde{\boldsymbol{\beta}}_{b-1})$ in [25], which cannot be ignored.*

3.3. Selection of regularization parameter

It is well known that the regularization parameter plays an important role in the penalized method. [33] verified that the SCAD penalized method with the regularization parameter selected by the Bayesian information criterion (BIC) can consistently identify the true model. Following [33], we use the BIC to choose the optimal value of the regularization parameters λ in (3.1) and λ_b in (3.7). Specifically, the BIC statistics are defined as

$$\begin{aligned}BIC(\lambda)=\;&\ln S_h(\tilde{\boldsymbol{\beta}}_\lambda^*)+df_\lambda\ln(n)/n;\\BIC(\lambda_b)=\;&\ln\Big\{\tfrac{1}{2N_b}(\tilde{\boldsymbol{\beta}}_{b,\lambda_b}-\tilde{\boldsymbol{\beta}}_{b-1})^\top\widetilde{J}_{b-1}(\tilde{\boldsymbol{\beta}}_{b,\lambda_b}-\tilde{\boldsymbol{\beta}}_{b-1})+\tfrac{1}{N_b}S_{h_b}(D_b;\tilde{\boldsymbol{\beta}}_{b,\lambda_b})\\&-\tfrac{N_{b-1}}{N_b}p'_{\lambda_{b-1}}(|\tilde{\boldsymbol{\beta}}_{b-1}|)sign(\tilde{\boldsymbol{\beta}}_{b-1})(\tilde{\boldsymbol{\beta}}_{b,\lambda_b}-\tilde{\boldsymbol{\beta}}_{b-1})\Big\}+df_{\lambda_b}\ln(N_b)/N_b,\end{aligned}\tag{3.8}$$

where $\tilde{\boldsymbol{\beta}}_\lambda^*$ and $\tilde{\boldsymbol{\beta}}_{b,\lambda_b}$ are the penalized estimators of β_0 by (3.1) and (3.7) given λ and λ_b, respectively; and df_λ and df_{λ_b} are the number of nonzero coefficients in $\tilde{\boldsymbol{\beta}}_\lambda^*$ and $\tilde{\boldsymbol{\beta}}_{b,\lambda_b}$, respectively.

3.4. Algorithm

This section is devoted to computational algorithm and numerical implementation. We focus on the algorithm for (3.7), and the algorithm for (3.1) is similar to that of (3.7). We describe a fast and easily implementable method using the local adaptive majorization-minimization (LAMM) principle [9].

As discussed in [8], the penalized function of SCAD is folded concavely with respect to $\boldsymbol{\beta}$, making it difficult to maximize. We propose applying the adaptive local linear approximation to the penalty function of SCAD [10] and approximately solve

$$\min_{\boldsymbol{\beta}}\left\{H_b(\boldsymbol{\beta})+p'_{\lambda_b}\left(|\tilde{\boldsymbol{\beta}}_b^{(r)}|\right)|\boldsymbol{\beta}|\right\},\tag{3.9}$$

where

$$\begin{aligned}H_b(\boldsymbol{\beta})=\Big\{&\tfrac{1}{2N_b}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_{b-1})^\top\widetilde{J}_{b-1}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_{b-1})+\tfrac{1}{N_b}S_{h_b}(D_b;\boldsymbol{\beta})\\&-\tfrac{N_{b-1}}{N_b}p'_{\lambda_{b-1}}(|\tilde{\boldsymbol{\beta}}_{b-1}|)sign(\tilde{\boldsymbol{\beta}}_{b-1})(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_{b-1})\Big\},\end{aligned}$$

and $\tilde{\boldsymbol{\beta}}_b^{(0)}$ is the initial estimator. We can take $\tilde{\boldsymbol{\beta}}_{b-1}$ as an initial estimator. As stated by [9], the majorization requirement only needs to hold locally at $\tilde{\boldsymbol{\beta}}_b^{(r+1)}$ when starting from $\tilde{\boldsymbol{\beta}}_b^{(r)}$. We therefore locally majorize $H_b(\boldsymbol{\beta})$ in (3.9) at $\tilde{\boldsymbol{\beta}}_b^{(r)}$ using an isotropic quadratic function

$$g_b\left(\boldsymbol{\beta}|\tilde{\boldsymbol{\beta}}_b^{(r)}\right)=H_b\left(\tilde{\boldsymbol{\beta}}_b^{(r)}\right)+\left(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_b^{(r)}\right)^\top H_b^{(1)}\left(\tilde{\boldsymbol{\beta}}_b^{(r)}\right)+\tfrac{\phi}{2}\|\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_b^{(r)}\|_2^2,$$

where

$$H_b^{(1)}(\boldsymbol{\beta})=\tfrac{1}{N_b}\widetilde{J}_{b-1}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_{b-1})+\tfrac{1}{N_b}U(D_b;\boldsymbol{\beta};h_b)-\tfrac{N_{b-1}}{N_b}p'_{\lambda_{b-1}}(|\tilde{\boldsymbol{\beta}}_{b-1}|)sign(\tilde{\boldsymbol{\beta}}_{b-1}),$$

and ϕ is a quadratic parameter such that $g_b\left(\tilde{\boldsymbol{\beta}}_b^{(r+1)}|\tilde{\boldsymbol{\beta}}_b^{(r)}\right)\geqslant H_b\left(\tilde{\boldsymbol{\beta}}_b^{(r+1)}\right)$. To find the smallest ϕ such that $g_b\left(\tilde{\boldsymbol{\beta}}_b^{(r+1)}|\tilde{\boldsymbol{\beta}}_b^{(r)}\right)\geqslant H_b\left(\tilde{\boldsymbol{\beta}}_b^{(r+1)}\right)$, the basic idea of LAMM is to start from a relatively small isotropic parameter $\phi=\phi_0=10^{-6}$ [9], 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:

$$\min_{\boldsymbol{\beta}}\left\{\left(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_b^{(r)}\right)^\top H_b^{(1)}\left(\tilde{\boldsymbol{\beta}}_b^{(r)}\right)+\frac{\phi}{2}\|\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_b^{(r)}\|_2^2+p'_{\lambda_b}\left(|\tilde{\boldsymbol{\beta}}_b^{(r)}|\right)|\boldsymbol{\beta}|\right\}.\tag{3.10}$$

It can be shown that (3.10) is minimized at

$$\tilde{\boldsymbol{\beta}}_b^{(r+1)} = Soft\Big(\tilde{\boldsymbol{\beta}}_b^{(r)} - \phi^{-1}H_b^{(1)}\Big(\tilde{\boldsymbol{\beta}}_b^{(r)}\Big), \phi^{-1}p'_{\lambda_b}\Big(|\tilde{\boldsymbol{\beta}}_b^{(r)}|\Big)\Big), \tag{3.11}$$

where $Soft(\boldsymbol{\mu}, \boldsymbol{v})$ is the soft-thresholding operator, defined by $Soft(\boldsymbol{\mu}, \boldsymbol{v}) = sign(\boldsymbol{\mu})\max(|\boldsymbol{\mu}| - \boldsymbol{v}, 0)$. The simplicity of this updating rule is because (3.10) is an unconstrained optimization problem. A simple stopping criterion for (3.11) is $\|\tilde{\boldsymbol{\beta}}_b^{(r+1)} - \tilde{\boldsymbol{\beta}}_b^{(r)}\|_2 \leqslant \epsilon$ for a sufficiently small ϵ, say 10^{-4}.

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 two real data analyses. All programs are written in R code. As mentioned in [13], the SQR method is insensitive to the choice of bandwidth h. In view of Theorem 2.1, we take $h_j = (N_j \ln N_j)^{-1/4}$ for simplicity, $j = 1, \ldots, b$, and the Gaussian kernel $K(u) = \left(\sqrt{2\pi}\right)^{-1}\exp(-u^2/2)$ in all of the numerical experiments.

Some settings in the simulation are described below: τ is the quantile level, p is the dimension of covariance, b is the number of batches, N_b is the sample size of all data, Cases 1 and 2 are the two different errors settings, RLS, SQR, OLEQR and RSQR are the estimation methods, PLS, PQR, PSQR and RPSQR are the variable selection methods, MSE is the mean squared error of coefficient estimation, t is the computation time of estimation method, **C** is the average proportion of nonzero coefficients correctly estimated to be nonzero, **IC** is the average proportion of zero coefficients incorrectly estimated to be nonzero, MAFE is the mean absolute fitting error and MAPE is the mean absolute prediction error.

Algorithm 2 The PSQR estimation for all data.

Input: all data sets $\{\mathbf{X}_i, Y_i\}_{i=1}^n$, the quantile level τ, kernel function $K(\cdot)$ and bandwidth h.
Initialize: select the initial estimator $\tilde{\boldsymbol{\beta}}^{*(0)}$ as the SQR estimator;
choose λ via (3.8);
for: $r = 0, 1, \ldots$, until $\|\tilde{\boldsymbol{\beta}}^{*(r+1)} - \tilde{\boldsymbol{\beta}}^{*(r)}\|_2 \leqslant \epsilon$ **do**
  Repeat
    $\tilde{\boldsymbol{\beta}}^{*(r+1)} = Soft\Big(\tilde{\boldsymbol{\beta}}^{*(r)} - \phi^{-1}S_h^{(1)}(\tilde{\boldsymbol{\beta}}^{*(r)}), \phi^{-1}p'_\lambda(|\tilde{\boldsymbol{\beta}}^{*(r)}|)\Big)$;
    If $g(\tilde{\boldsymbol{\beta}}^{*(r+1)}|\tilde{\boldsymbol{\beta}}^{*(r)}) < S_h^{(1)}(\tilde{\boldsymbol{\beta}}^{*(r+1)})$, where
    $g(\tilde{\boldsymbol{\beta}}^{*(r+1)}|\tilde{\boldsymbol{\beta}}^{*(r)}) = S_h(\tilde{\boldsymbol{\beta}}^{*(r)}) + \Big(\tilde{\boldsymbol{\beta}}^{*(r+1)} - \tilde{\boldsymbol{\beta}}_b^{(r)}\Big)^\top S_h^{(1)}(\tilde{\boldsymbol{\beta}}^{*(r)}) + \frac{\phi}{2}\|\tilde{\boldsymbol{\beta}}^{*(r+1)} - \tilde{\boldsymbol{\beta}}_b^{(r)}\|_2^2$; **then** $\phi \leftarrow 10\phi$;
  Until $g(\tilde{\boldsymbol{\beta}}^{*(r+1)}|\tilde{\boldsymbol{\beta}}^{*(r)}) \geqslant S_h^{(1)}(\tilde{\boldsymbol{\beta}}^{*(r+1)})$;
  Return $\tilde{\boldsymbol{\beta}}_b^{(r+1)}$ and $\phi \leftarrow \max\Big\{10^{-6}, \phi/10\Big\}$;
end
Output: $\tilde{\boldsymbol{\beta}}^* = \tilde{\boldsymbol{\beta}}^{*(r+1)}$.

Algorithm 3 The renewable PSQR estimation for streaming data sets.

Input: streaming data sets $D_1, \ldots, D_b, \ldots$, the quantile level τ, kernel function $K(\cdot)$ and bandwidths h_b with $b = 1, 2 \ldots$
Initialize: calculate $\tilde{\boldsymbol{\beta}}_1$ and λ_1 by Algorithm 2 with D_1, and compute $J(D_1; \tilde{\boldsymbol{\beta}}_1; h_1)$;
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 (3.8);
  for: $r = 0, 1, \ldots$, until $\|\tilde{\boldsymbol{\beta}}_b^{(r+1)} - \tilde{\boldsymbol{\beta}}_b^{(r)}\|_2 \leqslant \epsilon$ **do**
    Repeat
      $\tilde{\boldsymbol{\beta}}_b^{(r+1)} = Soft\Big(\tilde{\boldsymbol{\beta}}_b^{(r)} - \phi^{-1}H_b^{(1)}\Big(\tilde{\boldsymbol{\beta}}_b^{(r)}\Big), \phi^{-1}p'_{\lambda_b}\Big(|\tilde{\boldsymbol{\beta}}_b^{(r)}|\Big)\Big)$;
      If $g_b\Big(\tilde{\boldsymbol{\beta}}_b^{(r+1)}|\tilde{\boldsymbol{\beta}}_b^{(r)}\Big) < H_b\Big(\tilde{\boldsymbol{\beta}}_b^{(r+1)}\Big)$ **then** $\phi \leftarrow 10\phi$;
    Until $g_b\Big(\tilde{\boldsymbol{\beta}}_b^{(r+1)}|\tilde{\boldsymbol{\beta}}_b^{(r)}\Big) \geqslant H_b\Big(\tilde{\boldsymbol{\beta}}_b^{(r+1)}\Big)$;
    Return $\tilde{\boldsymbol{\beta}}_b^{(r+1)}$ and $\phi \leftarrow \max\Big\{10^{-6}, \phi/10\Big\}$;
  end
  update $\widetilde{J}_b = \widetilde{J}_{b-1} + J(D_b; \tilde{\boldsymbol{\beta}}_b; h_b)$, where $\tilde{\boldsymbol{\beta}}_b = \tilde{\boldsymbol{\beta}}_b^{(r+1)}$;
  save $\tilde{\boldsymbol{\beta}}_b, \widetilde{J}_b$ and λ_b, release data set D_b from the memory;
end
Output: $\tilde{\boldsymbol{\beta}}_b$ for $b = 2, 3, \ldots$

Note that in step 14 in Algorithm 3, we only need to save $\tilde{\boldsymbol{\beta}}_b, \widetilde{J}_b$ 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).

4.1. Simulation Example 1: renewable parameter estimation

In this section, we study the performance of the renewable SQR (RSQR) estimator proposed in Section 2. Furthermore, we include the following three competitors in our comparison: (1) the SQR estimator with full data, which can be obtained by the conquer algorithm in [13]; (2) the renewable least squares estimator (RLS) for the streaming data set, as given in [24]; and (3) the online linear estimator for the QR (OLEQR) for the streaming data set, as given in [2].

We generate data from the following linear model:

$$\mathbf{Y} = \mathbf{X}^\top\beta_0 + \sigma(\mathbf{X})\Big\{\varepsilon - F_\varepsilon^{-1}(\tau)\Big\} \tag{4.1}$$

where $\mathbf{X} = (1, \mathbf{X}_1, \ldots, \mathbf{X}_p)^\top$ is a $(p+1)$-dimensional covariate vector and $(\mathbf{X}_1, \ldots, \mathbf{X}_p)$ is drawn from a multivariate normal distribution $N(0, \bar{\boldsymbol{\Sigma}})$. The covariance matrix $\bar{\boldsymbol{\Sigma}}$ is constructed by $\bar{\Sigma}_{ij} = 0.5^{i-j}$ for $1 \leqslant i, j \leqslant p$ with $p = 10$ and 100. The true value of the parameter

is $\beta_0 = \mathbf{1}_{(p+1)\times 1}$, where $\mathbf{1}_{(p+1)\times 1}$ is the $(p+1)$-dimensional vector with all elements being one. $F_{\varepsilon}^{-1}(\tau)$ is the τ-quantile of ε, which is used to eliminate the influence of quantiles. Two error distributions of ε are considered: a standard normal distribution ($N(0,1)$) and a t distribution with 3 degrees of freedom ($t(3)$). In this section, we consider that the sample size of each batch is $\bar{n}$. Then, the full data are $N_b = \bar{n}b$, and we consider the following two cases:

Case 1 (Normal errors): $\sigma(\mathbf{X}) = \mathbf{1}_{N_b\times 1}$ and $\varepsilon \sim N(0,1)$, and.

Case 2 (Heteroscedastic errors): $\sigma(\mathbf{X}) = \mathbf{1}_{N_b\times 1} + 0.5\cos(\mathbf{X}^{\top}\beta_0)$ and $\varepsilon \sim t(3)$.

To evaluate the performance of the four methods, we calculate the mean squared error (MSE): $\|\hat{\beta} - \beta_0\|_2$ and computation time (in seconds). Simulation results are based on 100 simulation replications.

4.1.1. Fixed $\bar{n}$ with varying b

In this section, we fix the sample size of each batch as $\bar{n} = 200$ and vary the number of batches $b = 100, \ldots, 1000$ for $\tau = 0.3, 0.5$ and 0.7. From Figs. 1–3, the following conclusions can be drawn:

(1) In terms of the MSEs in Figs. 1 and 2, we note that (i) all the estimators are close to the true value because the MSEs are very small; and (ii) for any given number of batches b, p, errors and quantiles τ, the figures show that the MSEs of the proposed estimator (RSQR) are very close to those of the SQR and better than those of the OLEQR.
(2) We only present the results of $\tau = 0.5$ for the QR because the computation times of different quantiles are similar. In terms of the computation time in Fig. 3, we note that (i) the computation time of all estimation methods is very small because the results of the computation times are very small; and (ii) the computation times of the three methods are close at $p = 10$, and the computation time of the SQR is less than those of the RSQR and OLEQR.

4.1.2. Fixed N_b with varying b

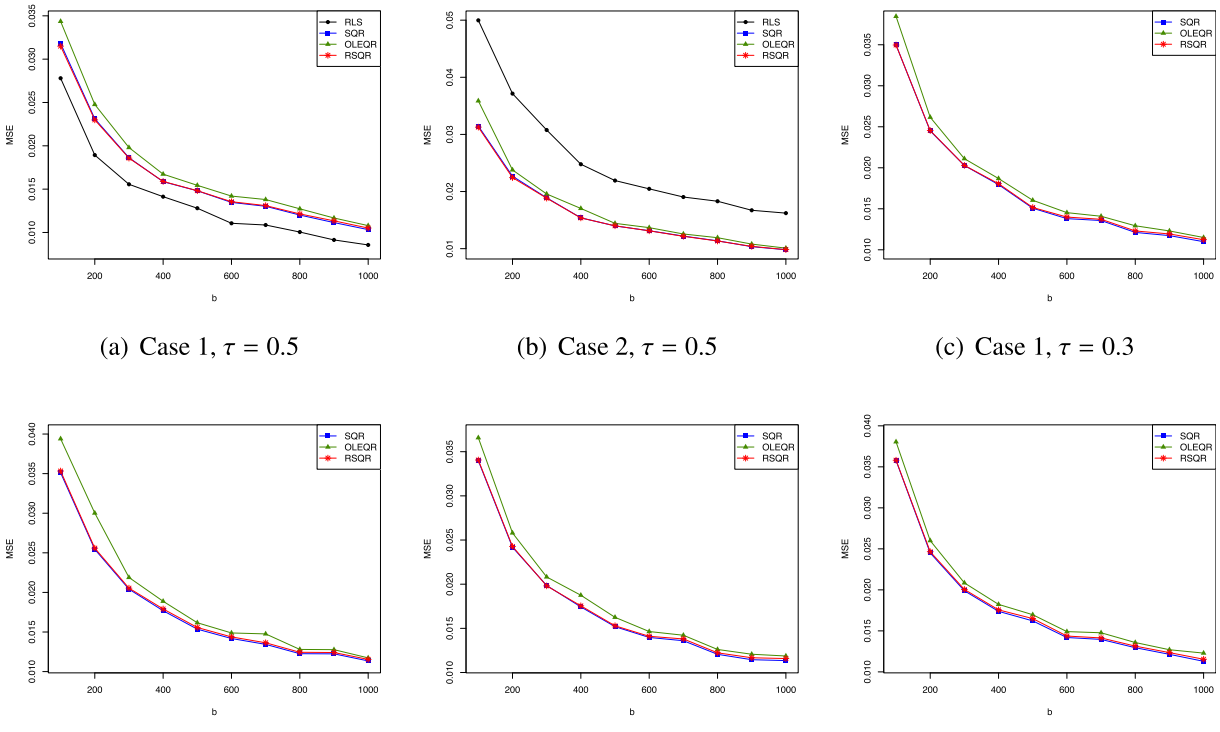
In this section, we fix the sample size of the full data $N_b = 10^6$ and vary the number of batches $b = 10, 50, 100, 200, 500, 1000, 2000$ for $\tau = 0.1, 0.5$ and 0.9. The simulation results are presented in Tables 2–5, and the following conclusions can be drawn:

(1) In terms of the MSEs in Tables 2–4, we note that (i) all the estimators are close to the true value because the MSEs are very small; and (ii) for any given number of batches b, p and quantiles τ, the tables show that the MSEs of the RSQR are very close to those of the SQR and better than those of the OLEQR.
(2) In terms of the computation times in Table 5, we note that (i) all estimation methods are very fast. (ii) When the data volume is not particularly large, there is little difference in the calculation/estimation time. Because OLEQR, as a one-round smoothing quantile regression method, is faster than our proposed method RSQR which is a multi-round smoothing quantile regression method. However, the computation times of RSQR are very close to those of OLEQR because our method RSQR needs fewer iterations to achieve convergence. (iii) Once the data size is large, RSQR obviously saves time.

4.2. Simulation Example 2: renewable variable selection

In this section, we study the performances of the PSQR estimator method and the renewable penalized SQR estimator (RPSQR) method proposed in Section 3. Furthermore, we include the following two competitors in our comparison:

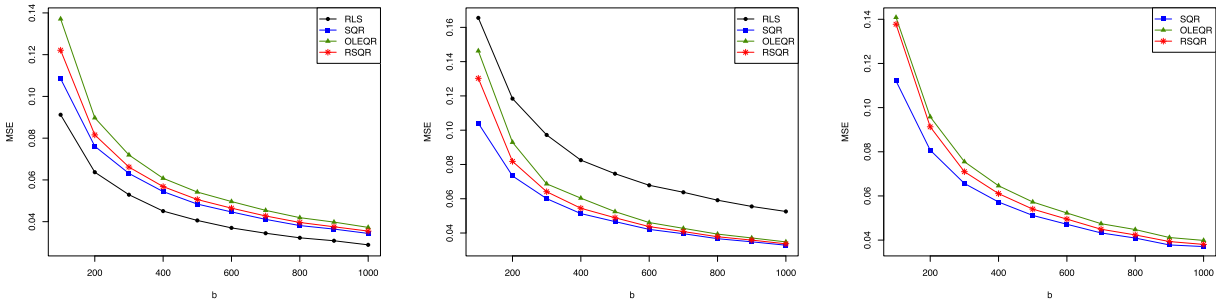
(1) the penalized least squares estimator (PLS) estimator with full data, which can be obtained by the "ncvreg" function with the "SCAD" method in the R package "ncvreg"; and



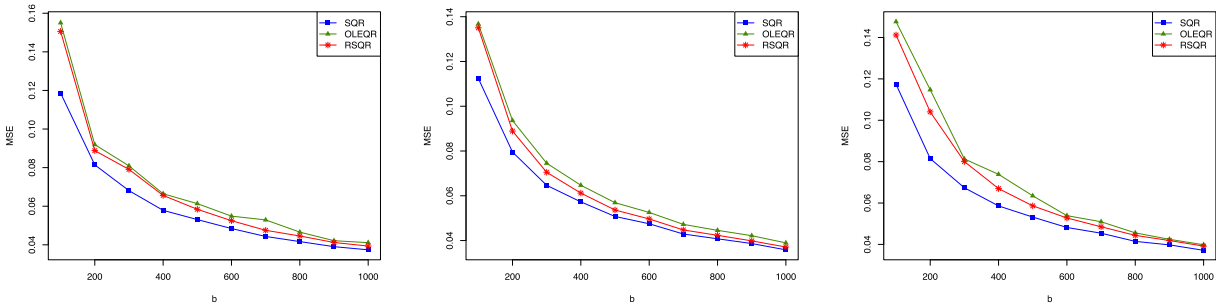
(a) Case 1, $\tau = 0.5$ (b) Case 2, $\tau = 0.5$ (c) Case 1, $\tau = 0.3$

(d) Case 2, $\tau = 0.3$ (e) Case 1, $\tau = 0.7$ (f) Case 2, $\tau = 0.7$

Fig. 1. The mean MSEs under different batches b, quantiles τ, methods and errors for simulation Example 1 with a fixed $\bar{n}$ and p = 10.

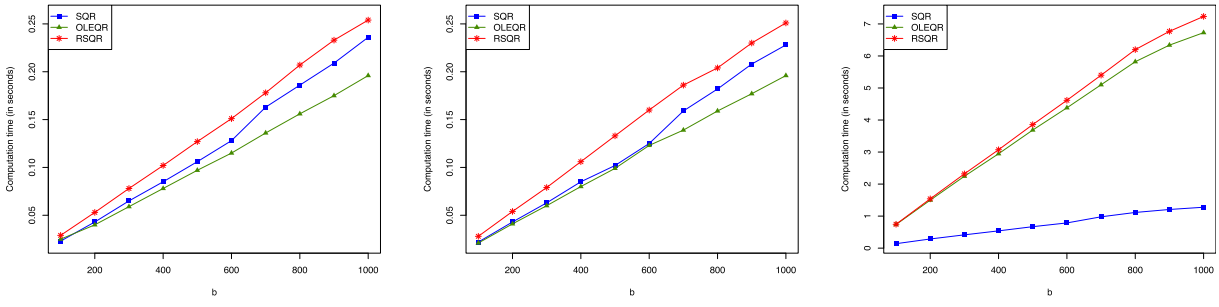


(a) Case 1, $\tau = 0.5$ (b) Case 2, $\tau = 0.5$ (c) Case 1, $\tau = 0.3$

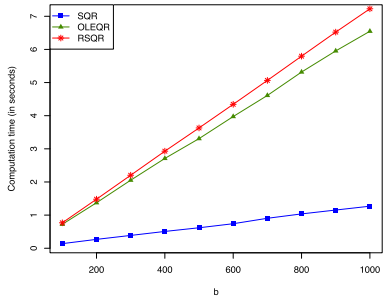


(d) Case 2, $\tau = 0.3$ (e) Case 1, $\tau = 0.7$ (f) Case 2, $\tau = 0.7$

Fig. 2. The mean MSEs under different batches b, quantiles τ, methods and errors for simulation Example 1 with a fixed $\bar{n}$ and $p = 100$.



(a) Case 1, p=10 (b) Case 2, p=10 (c) Case 1, p=100



(d) Case 2, p=100

Fig. 3. The computation times (in seconds) under different batches b, p, methods and errors for simulation Example 1 with a fixed $\bar{n}$ and $\tau = 0.5$.

Table 2
The means and standard deviations (in parentheses) of the MSEs (×100) under different batches b, methods and errors for simulation Example1 with a fixed N_b and $\tau = 0.5$.

Errors	p	b	RLS	SQR	OLEQR	RSQR
Casse 1	10	10	0.383 (0.096)	0.445 (0.124)	0.487 (0.138)	0.485 (0.131)
		50	0.383 (0.096)	0.445 (0.124)	0.488 (0.138)	0.485 (0.131)
		100	0.383 (0.096)	0.445 (0.124)	0.489 (0.136)	0.485 (0.131)
		200	0.383 (0.096)	0.445 (0.124)	0.490 (0.136)	0.485 (0.131)
		500	0.383 (0.096)	0.445 (0.124)	0.490 (0.137)	0.485 (0.131)
		1000	0.383 (0.096)	0.445 (0.124)	0.491 (0.137)	0.485 (0.131)
		2000	0.383 (0.096)	0.445 (0.124)	0.489 (0.138)	0.485 (0.131)
	100	10	1.287 (0.104)	1.529 (0.126)	1.628 (0.138)	1.583 (0.131)
		50	1.287 (0.104)	1.529 (0.126)	1.633 (0.137)	1.585 (0.131)
		100	1.287 (0.104)	1.529 (0.126)	1.632 (0.139)	1.585 (0.132)
		200	1.287 (0.104)	1.529 (0.126)	1.634 (0.139)	1.585 (0.131)
		500	1.287 (0.104)	1.529 (0.126)	1.633 (0.135)	1.585 (0.131)
		1000	1.287 (0.104)	1.529 (0.126)	1.635 (0.137)	1.586 (0.130)
		2000	1.287 (0.104)	1.529 (0.126)	1.631 (0.135)	1.587 (0.130)
Casse 2	10	10	0.695 (0.179)	0.433 (0.110)	0.450 (0.110)	0.447 (0.110)
		50	0.695 (0.179)	0.433 (0.110)	0.450 (0.111)	0.448 (0.109)
		100	0.695 (0.179)	0.433 (0.110)	0.450 (0.111)	0.449 (0.110)
		200	0.695 (0.179)	0.433 (0.110)	0.450 (0.109)	0.448 (0.109)
		500	0.695 (0.179)	0.433 (0.110)	0.448 (0.109)	0.448 (0.111)
		1000	0.695 (0.179)	0.433 (0.110)	0.449 (0.110)	0.447 (0.110)
		2000	0.695 (0.179)	0.433 (0.110)	0.450 (0.110)	0.449 (0.111)
	100	10	2.368 (0.202)	1.453 (0.127)	1.517 (0.131)	1.473 (0.128)
		50	2.368 (0.202)	1.453 (0.127)	1.515 (0.128)	1.473 (0.128)
		100	2.368 (0.202)	1.453 (0.127)	1.517 (0.128)	1.473 (0.128)
		200	2.368 (0.202)	1.453 (0.127)	1.519 (0.127)	1.473 (0.128)
		500	2.368 (0.202)	1.453 (0.127)	1.515 (0.125)	1.473 (0.128)
		1000	2.368 (0.202)	1.453 (0.127)	1.517 (0.125)	1.474 (0.128)
		2000	2.368 (0.202)	1.453 (0.127)	1.517 (0.127)	1.474 (0.127)

Table 3
The means and standard deviations (in parentheses) of the MSEs (×100) under different batches b, methods and errors for simulation Example1 with a fixed N_b and $\tau = 0.1$.

Errors	p	b	SQR	OLEQR	RSQR
Casse 1	10	10	0.569 (0.138)	0.654 (0.164)	0.645 (0.158)
		50	0.569 (0.138)	0.652 (0.164)	0.649 (0.160)
		100	0.569 (0.138)	0.651 (0.164)	0.650 (0.156)
		200	0.569 (0.138)	0.656 (0.157)	0.653 (0.158)
		500	0.569 (0.138)	0.657 (0.162)	0.654 (0.160)
		1000	0.569 (0.138)	0.658 (0.161)	0.656 (0.161)
		2000	0.569 (0.138)	0.658 (0.161)	0.657 (0.164)
	100	10	1.949 (0.164)	2.240 (0.186)	2.141 (0.176)
		50	1.949 (0.164)	2.255 (0.192)	2.145 (0.175)
		100	1.949 (0.164)	2.267 (0.186)	2.146 (0.175)
		200	1.949 (0.164)	2.269 (0.192)	2.146 (0.174)
		500	1.949 (0.164)	2.265 (0.195)	2.150 (0.176)
		1000	1.949 (0.164)	2.268 (0.188)	2.161 (0.174)
		2000	1.949 (0.164)	2.271 (0.188)	2.191 (0.175)
Casse 2	10	10	0.844 (0.214)	1.011 (0.255)	0.998 (0.246)
		50	0.844 (0.214)	1.001 (0.254)	1.003 (0.251)
		100	0.844 (0.214)	1.019 (0.254)	1.003 (0.252)
		200	0.844 (0.214)	1.016 (0.255)	1.004 (0.250)
		500	0.844 (0.214)	1.020 (0.247)	1.005 (0.249)
		1000	0.844 (0.214)	1.010 (0.244)	1.005 (0.250)
		2000	0.844 (0.214)	1.007 (0.256)	1.003 (0.248)
	100	10	2.878 (0.253)	3.351 (0.263)	3.178 (0.251)
		50	2.878 (0.253)	3.418 (0.274)	3.195 (0.251)
		100	2.878 (0.253)	3.429 (0.275)	3.202 (0.253)
		200	2.878 (0.253)	3.420 (0.266)	3.202 (0.254)
		500	2.878 (0.253)	3.418 (0.282)	3.243 (0.253)
		1000	2.878 (0.253)	3.432 (0.274)	3.273 (0.246)
		2000	2.878 (0.253)	3.471 (0.282)	3.361 (0.245)

Table 4
The means and standard deviations (in parentheses) of the MSEs ($\times 100$) under different batches b, methods and errors for simulation Example 1 with a fixed N_b and $\tau = 0.9$.

Errors	p	b	SQR	OLEQR	RSQR
Casse 1	10	10	0.559 (0.149)	0.701 (0.185)	0.642 (0.178)
		50	0.559 (0.149)	0.700 (0.181)	0.644 (0.181)
		100	0.559 (0.149)	0.703 (0.181)	0.644 (0.180)
		200	0.559 (0.149)	0.703 (0.182)	0.644 (0.180)
		500	0.559 (0.149)	0.706 (0.178)	0.644 (0.180)
		1000	0.559 (0.149)	0.705 (0.181)	0.644 (0.180)
		2000	0.559 (0.149)	0.707 (0.179)	0.643 (0.181)
	100	10	1.968 (0.172)	2.278 (0.181)	2.177 (0.176)
		50	1.968 (0.172)	2.289 (0.185)	2.181 (0.175)
		100	1.968 (0.172)	2.307 (0.182)	2.181 (0.176)
		200	1.968 (0.172)	2.307 (0.186)	2.181 (0.177)
		500	1.968 (0.172)	2.300 (0.182)	2.183 (0.176)
		1000	1.968 (0.172)	2.300 (0.182)	2.192 (0.183)
		2000	1.968 (0.172)	2.303 (0.177)	2.226 (0.181)
Casse 2	10	10	0.791 (0.213)	0.950 (0.238)	0.943 (0.232)
		50	0.791 (0.213)	0.946 (0.246)	0.945 (0.235)
		100	0.791 (0.213)	0.951 (0.242)	0.945 (0.236)
		200	0.791 (0.213)	0.956 (0.243)	0.945 (0.236)
		500	0.791 (0.213)	0.967 (0.247)	0.946 (0.238)
		1000	0.791 (0.213)	0.975 (0.248)	0.947 (0.236)
		2000	0.791 (0.213)	0.960 (0.253)	0.947 (0.236)
	100	10	2.863 (0.219)	3.352 (0.248)	3.182 (0.227)
		50	2.863 (0.219)	3.419 (0.255)	3.196 (0.230)
		100	2.863 (0.219)	3.434 (0.258)	3.197 (0.229)
		200	2.863 (0.219)	3.422 (0.249)	3.201 (0.233)
		500	2.863 (0.219)	3.409 (0.260)	3.243 (0.235)
		1000	2.863 (0.219)	3.431 (0.276)	3.292 (0.237)
		2000	2.863 (0.219)	3.469 (0.268)	3.441 (0.254)

Table 5
The mean computation times (in seconds) under different full data sample sizes N_b, batches b, methods and errors for simulation Example 1 with a fixed $p = 10$ and $\tau = 0.5$.

		Case 1				**Case 2**			
N_b	b	RLS	SQR	OLEQR	RSQR	RLS	SQR	OLEQR	RSQR
10^6	10	0.20	1.33	0.54	0.78	0.18	1.22	0.50	0.70
	50	0.18	1.33	0.45	0.71	0.17	1.22	0.46	0.66
	100	0.17	1.33	0.43	0.66	0.17	1.22	0.45	0.68
	200	0.16	1.33	0.44	0.66	0.17	1.22	0.42	0.60
	500	0.17	1.33	0.46	0.70	0.17	1.22	0.45	0.65
	1000	0.17	1.33	0.61	0.75	0.18	1.22	0.50	0.69
	2000	0.19	1.33	0.64	0.86	0.19	1.22	0.60	0.82
10^7	10	2.53	12.00	6.97	8.75	2.93	11.45	7.00	9.19
	50	2.55	12.00	6.40	7.45	2.73	11.45	5.84	7.24
	100	2.48	12.00	6.04	7.32	2.39	11.45	5.51	6.88
	200	2.68	12.00	5.89	7.17	2.39	11.45	5.62	6.75
	500	2.31	12.00	4.92	6.46	2.68	11.45	4.77	6.04
	1000	1.78	12.00	4.36	5.61	2.05	11.45	4.50	5.36
	2000	1.81	12.00	4.24	5.16	1.87	11.45	4.41	5.02
2×10^7	10	50.16	49.54	21.38	24.93	90.10	66.60	20.87	30.57
	50	5.17	49.54	12.30	14.71	7.28	66.60	9.37	14.62
	100	5.39	49.54	12.11	14.35	4.10	66.60	8.78	11.34
	200	5.07	49.54	11.92	13.78	4.31	66.60	8.65	12.47
	500	5.23	49.54	11.53	12.67	4.52	66.60	8.47	9.98
	1000	5.10	49.54	9.80	11.39	4.34	66.60	8.28	9.55
	2000	3.84	49.54	8.60	9.94	4.04	66.60	8.19	9.28

(2) the penalized QR (PQR) estimator with full data, which can be obtained by the "rq" function with the "SCAD" method in the R package "quantreg".

The data are generated from model (4.1) with $\beta_0 = (1, 1, 2, 3, 4, 5, 0, \ldots, 0)$ and $p = 100$. We fix the sample size of each batch as $\bar{n} = 400$ and vary the number of batches $b = 100, \ldots, 1000$. The other settings are the same as in simulation Example 1.

To evaluate the performance of the four methods, we calculate the MSE in simulation Example 1, the average proportion of nonzero coefficients correctly estimated to be nonzero (denoted as **C**), and the average proportion of zero coefficients incorrectly estimated to be nonzero (denoted as **IC**). Note that **C** = 1 and **IC** = 0 imply perfect recovery. We further study the computational efficiency of our proposed estimator using the computation time (in seconds). The simulation results for $\tau = 0.2, 0.5$ and 0.8 are presented in Tables 6–10, respectively, based on 100 simulation replications. From Tables 6–10, the following conclusions can be drawn:

(1) In terms of the MSEs in Tables 6–8, we note that (i) all the estimators are close to the true value because the MSEs are very small; and (ii) for any given number of batches b, p, errors and

Table 6
The means and standard deviations (in parentheses) of the MSEs (×100) under different batches b, methods and errors for simulation Example 2 with a fixed $\bar{n} = 400$ and $\tau = 0.5$.

Errors	b	PLS	PQR	PSQR	RPSQR
Casse 1	100	1.346 (0.467)	7.943 (0.722)	1.633 (0.508)	1.701 (0.578)
	200	0.941 (0.285)	5.682 (0.437)	1.112 (0.391)	1.176 (0.436)
	300	0.772 (0.241)	4.601 (0.399)	0.895 (0.322)	0.931 (0.311)
	400	0.660 (0.225)	4.033 (0.350)	0.742 (0.299)	0.777 (0.315)
	500	0.641 (0.200)	3.566 (0.331)	0.730 (0.250)	0.775 (0.281)
	600	0.563 (0.192)	3.318 (0.267)	0.698 (0.249)	0.726 (0.269)
	700	0.521 (0.187)	3.036 (0.238)	0.602 (0.213)	0.642 (0.247)
	800	0.484 (0.170)	2.870 (0.236)	0.557 (0.211)	0.582 (0.230)
	900	0.443 (0.136)	2.665 (0.205)	0.510 (0.177)	0.525 (0.187)
	1000	0.431 (0.157)	2.506 (0.228)	0.510 (0.181)	0.525 (0.204)
Casse 2	100	2.541 (0.775)	7.544 (0.555)	1.509 (0.622)	1.644 (0.639)
	200	1.668 (0.599)	5.317 (0.408)	1.076 (0.385)	1.180 (0.407)
	300	1.491 (0.502)	4.387 (0.355)	0.856 (0.329)	0.902 (0.338)
	400	1.257 (0.356)	3.743 (0.298)	0.772 (0.263)	0.829 (0.272)
	500	1.170 (0.348)	3.372 (0.271)	0.688 (0.262)	0.718 (0.265)
	600	0.983 (0.327)	3.080 (0.242)	0.601 (0.192)	0.622 (0.202)
	700	0.962 (0.323)	2.868 (0.240)	0.592 (0.191)	0.615 (0.199)
	800	0.893 (0.313)	2.701 (0.238)	0.542 (0.184)	0.562 (0.189)
	900	0.815 (0.282)	2.430 (0.216)	0.501 (0.175)	0.520 (0.168)
	1000	0.802 (0.256)	2.378 (0.202)	0.481 (0.165)	0.491 (0.164)

Table 7
The means and standard deviations (in parentheses) of the MSEs (×100) under different batches b, methods and errors for simulation Example 2 with a fixed $\bar{n} = 400$ and $\tau = 0.2$.

Errors	b	PQR	PSQR	RPSQR
Casse 1	100	8.823 (0.928)	1.785 (0.681)	1.839 (0.758)
	200	6.109 (0.335)	1.244 (0.450)	1.272 (0.497)
	300	5.227 (0.176)	1.054 (0.334)	1.070 (0.352)
	400	4.680 (0.154)	0.943 (0.289)	0.958 (0.297)
	500	4.077 (0.412)	0.823 (0.283)	0.852 (0.288)
	600	3.839 (0.170)	0.718 (0.255)	0.734 (0.268)
	700	3.299 (0.176)	0.685 (0.238)	0.701 (0.240)
	800	3.272 (0.321)	0.652 (0.219)	0.675 (0.234)
	900	3.102 (0.214)	0.634 (0.195)	0.652 (0.200)
	1000	2.977 (0.167)	0.549 (0.179)	0.557 (0.189)
Casse 2	100	9.918 (1.010)	2.160 (0.698)	2.541 (0.931)
	200	7.628 (0.371)	1.523 (0.526)	1.774 (0.578)
	300	5.975 (0.637)	1.202 (0.451)	1.393 (0.538)
	400	5.335 (0.543)	1.054 (0.368)	1.167 (0.422)
	500	4.640 (0.533)	0.885 (0.277)	1.073 (0.388)
	600	4.498 (0.172)	0.812 (0.295)	0.889 (0.345)
	700	4.241 (0.395)	0.825 (0.269)	0.860 (0.263)
	800	3.930 (0.428)	0.701 (0.258)	0.799 (0.271)
	900	3.389 (0.140)	0.763 (0.242)	0.819 (0.254)
	1000	3.319 (0.183)	0.676 (0.226)	0.711 (0.249)

quantiles τ, the tables show that the MSEs of RPSQR are very close to those of the PSQR and better than those of the PQR.
(2) In terms of the **IC**s in Table 9, the performance of the proposed PSQR and RPSQR methods are very good with **IC**s close to 0 under different batches, quantiles and errors. The **IC**s of the PLS and PQR are all zero and one, respectively, under different batches, quantiles and errors. The performance of the PQR is bad, which may be because the PQR by "rq" with the "scad" method should probably be regarded as experimental, as mentioned in the report of the "quantreg" R package. Therefore, our proposed variable selection method PSQR is a good method for quantile regression, because it runs fast and can accurately select important variables.
(3) The four methods can select all true predictors in all settings and thus we do not report **C** in the tables.
(4) In terms of computation time in Table 10, we note that (i) for any given number of machines K, quantiles τ and error terms, the computation time of the PQR is the longest, as expected. Moreover, the proposed PSQR and RPSQR methods are much faster to compute than the PQR. (ii) The computation time of PSQR and RPSQR is very close. Therefore, the renewable method does not add much computational complexity.

4.3. Real data Example 1: Beijing multisite air-quality data set

We apply the proposed RSQR method in Section 2 to the analysis of the Beijing multisite air-quality dataset. The dataset includes 420768 hourly air pollutant data points from 12 nationally-controlled air-quality monitoring sites. The air-quality data are from the Beijing Municipal Environmental Monitoring Center. The meteorological data at each air-quality site were matched with the nearest weather station from the China Meteorological Administration. The time period is from March 1st, 2013 to February 28th, 2017. The dataset was obtained from the following online website: https://archive.ics.uci.edu/ml/datasets/Beijing + Multi-Site + Air-Quality + Data.

In this study, we use model (1.1) to explore the relationship between the $PM_{2.5}$ concentration (ug/m^3) and seven variables in Table 11. Because the data are from 12 nationally-controlled air-quality monitoring sites, we set the number of batches $b = 12$.

Table 8
The means and standard deviations (in parentheses) of the MSEs (×100) under different batches b, methods and errors for simulation Example 2 with a fixed $\bar{n} = 400$ and $\tau = 0.8$.

Errors	b	PQR	PSQR	RPSQR
Casse 1	100	9.696 (0.458)	1.861 (0.632)	1.894 (0.640)
	200	6.274 (0.436)	1.270 (0.450)	1.294 (0.439)
	300	5.687 (0.464)	1.064 (0.342)	1.100 (0.370)
	400	4.456 (0.443)	0.868 (0.300)	0.899 (0.308)
	500	4.238 (0.280)	0.850 (0.300)	0.863 (0.310)
	600	3.598 (0.147)	0.778 (0.264)	0.799 (0.273)
	700	3.477 (0.249)	0.707 (0.221)	0.723 (0.241)
	800	3.198 (0.210)	0.600 (0.191)	0.610 (0.198)
	900	2.882 (0.233)	0.595 (0.189)	0.605 (0.191)
	1000	2.671 (0.215)	0.568 (0.177)	0.581 (0.185)
Casse 2	100	10.602 (0.564)	2.220 (0.753)	2.567 (0.752)
	200	7.486 (0.475)	1.500 (0.512)	1.758 (0.550)
	300	5.984 (0.612)	1.199 (0.433)	1.417 (0.535)
	400	5.404 (0.374)	1.017 (0.396)	1.127 (0.446)
	500	4.854 (0.333)	0.880 (0.296)	1.007 (0.363)
	600	4.087 (0.350)	0.871 (0.295)	0.993 (0.350)
	700	3.929 (0.322)	0.798 (0.278)	0.851 (0.298)
	800	3.538 (0.308)	0.773 (0.253)	0.831 (0.265)
	900	3.479 (0.279)	0.695 (0.230)	0.772 (0.241)
	1000	3.433 (0.236)	0.694 (0.225)	0.750 (0.240)

Table 9
The mean C (×100) of the PSQR, and RPSQR estimators under different τ, b and errors for simulation Example 2.

Errors	τ	Methods	b = 100	200	300	400	500	600	700	800	900	1000
Case 1	0.2	PSQR	0.0	0.0	0.1	0.1	0.1	0.0	0.1	0.1	0.0	0.0
		RPSQR	0.3	0.2	0.1	0.1	0.1	0.1	0.1	0.2	0.1	0.0
	0.5	PSQR	0.2	0.3	0.5	0.3	0.1	0.4	0.1	0.2	0.2	0.3
		RPSQR	0.5	0.5	0.2	0.3	0.4	0.2	0.5	0.3	0.2	0.3
	0.8	PSQR	0.1	0.3	0.2	0.1	0.0	0.1	0.0	0.1	0.0	0.0
		RPSQR	0.3	0.3	0.3	0.1	0.1	0.1	0.1	0.1	0.2	0.1
Case 2	0.2	PSQR	0.2	0.1	0.1	0.0	0.1	0.1	0.1	0.1	0.1	0.1
		RPSQR	1.4	1.3	1.1	0.8	0.6	0.7	0.6	0.8	0.5	0.5
	0.5	PSQR	0.9	0.3	0.2	0.3	0.5	0.2	0.2	0.4	0.2	0.4
		RPSQR	1.1	1.1	0.6	0.7	0.5	0.4	0.6	0.4	0.3	0.4
	0.8	PSQR	0.4	0.2	0.3	0.1	0.1	0.1	0.1	0.0	0.1	0.1
		RPSQR	1.8	1.5	1.5	1.0	1.1	1.0	0.8	0.9	0.9	0.7

Table 10
The mean computation times (in seconds) of the PQR, PSQR, and RPSQR estimators with $\tau = 0.5$ under different b and errors for simulation Example 2.

Errors	Methods	b = 100	200	300	400	500	600	700	800	900	1000
Case 1	PQR	19.88	42.68	66.24	88.74	119.88	142.98	180.12	217.88	219.72	247.01
	PSQR	1.79	3.46	4.88	7.16	8.58	9.75	11.73	12.76	13.85	16.05
	RPSQR	2.02	3.79	5.62	7.51	9.27	11.09	12.71	14.43	16.21	17.97
Case 2	PQR	19.21	40.02	71.58	100.38	123.42	148.32	172.98	189.72	234.36	282.72
	PSQR	1.79	3.43	5.01	6.58	8.00	9.66	11.19	12.36	13.73	14.34
	RPSQR	2.04	3.69	5.28	6.94	8.64	10.27	11.95	13.63	15.25	16.99

Fig. 4 depicts the changes in the estimated coefficients for the Beijing multisite air-quality data using our proposed RSQR method with quantiles $\tau = 0.1$ to 0.9. From Fig. 4, it is easy to see that the estimated coefficients of TEMP and PRES decrease as quantile τ increases, and the other estimated coefficients increase as quantile τ increases. Furthermore, we evaluate the performance of the proposed RSQR estimator compared with the SQR, RLS and OLEQR, based on the mean absolute fitting error (MAFE):

$$MAFE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \widehat{Y}_i|,$$

where n is the total sample size of 388817 (the number of data points after deleting missing data), Y_i is the value of $PM_{2.5}$, and $\widehat{Y}_i$ is the fitted value of Y_i at quantile $\tau = 0.5$. The results are present in Table 12. In terms of the MAFE, we find that the performance of the SQR is the best and that the performance of the RSQR is very close to that of the SQR. The computation time of the RSQR is less than that of the SQR.

4.4. Real data Example2: Year Prediction MSD data set

As an illustration, we now apply the proposed PSQR and RPSQR methodologies in Section 3 to the Year Prediction MSD dataset. The dataset is collected from the public database of the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/YearPredictionMSD). The dataset is a freely-available collection of audio features for contemporary popular music tracks ranging from 1922 to 2011. Approximately 515345 observations were recorded with 91 variables: the year of the song and 12 average timbre and 78 timbre covariance variables. The research problem is to predict the release year of songs from the audio features.

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 covari-

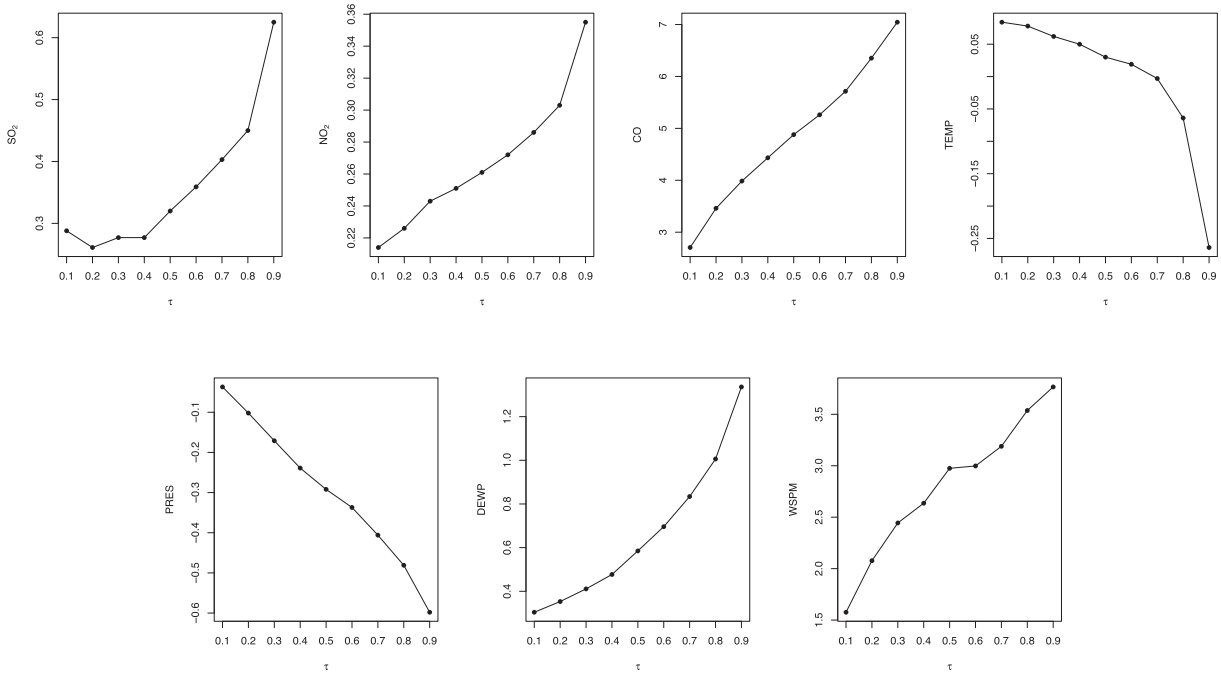


Fig. 4. The estimated coefficients of RSQR under different quantiles τ for real data Example 1.

Table 11
Covariates and their descriptions for real data Example 1.

Name	Description
SO_2	SO_2 concentration (ug/m^3)
NO_2	NO_2 concentration (ug/m^3)
CO	CO concentration (ug/m^3)
TEMP	temperature (degrees Celsius)
PRES	pressure (hPa)
DEWP	dew point temperature (degrees Celsius)
WSPM	wind speed (m/s)

Table 12
The MAFEs and computation times (in seconds) with $\tau = 0.5$ for the RLS, SQR, OLEQR and RSQR estimators for real data Example 1.

	RLS	SQR	OLEQR	RSQR
MAFE	30.39	29.23	29.26	29.23
t	0.06	1.37	0.28	0.31

ance variables are the covariate variables, is used to fit the data. To evaluate the performances of our proposed methods (PSQR and RPSQR) in Section 3, we first calculate the mean absolute prediction error (MAPE) of the predictions under quantile $\tau = 0.5$. The first 500000 data points are used for the estimation, and the remaining 15345 data points are used for the prediction. Therefore,

$$MAPE = \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\left|Y_i - \widehat{Y}_i\right|,$$

where $\widehat{Y}_i$ is the fitted value of Y_i and $\tilde{n} = 15345$. The results of MAPEs and their standard deviations by bootstrap method [3], are presented in Table 13. The Table 13 clearly shows that the penalized QR estimators (PQR, PSQR and RPSQR) are better than the penalized LS estimator because of the smaller MAPEs. The performances of PQR, PSQR and RPSQ are very close, because their asymptotic properties are the same, see Theorems 1 and 2 in [38] (PQR), Lemmas 1 and 2 (PSQR) and Theorems 3.1 and 3.2 (RPSQR). Methods PQR and PSQR are a little better than RPSQR because they directly use full data. Moreover, from Theorems 1 and 2, our proposed method RPSQR is not influenced by the number of batches b, so the results (MAPE) of RPSQR for different b are very close.

Furthermore, to illustrate the computational advantage of the proposed methods (PSQR and RPSQR), we also list the running time of our method under different b and quantiles τ in Table 14. The results show that PSQR costs much less time than the PQR, and its computation time is close to that of the RPSQR. In addition, we study the number of variables selected by our proposed methods (PSQR and RPSQR). The results are presented in Table 14, which shows that the SCAD produces a small model because the numbers of selected variables under different bs and quantiles levels τ are all smaller than the case with $p = 90$ variables. Moreover, for any given b, the number of selected variables decreases as the quantile level τ increases. The performances (MAPE, t and NSV) of the RPSQR under different bs are close to those of the PSQR.

5. Discussion

In this article, we considered renewable parameter estimation and variable selection for a quantile regression with streaming data sets. The method requires only the availability of the current

Table 13
The MAPEs and standard deviations (in parentheses) of the PLS, PQR, PSQR, and RPSQR estimators with $\tau = 0.5$ under different bs for real data Example 2.

Method	MAPE
PLS	6.906 (0.004)
PQR	6.694 (0.002)
PSQR	6.694 (0.002)
RPSQR (b = 100)	6.707 (0.003)
RPSQR (b = 200)	6.718 (0.004)
RPSQR (b = 300)	6.767 (0.004)
RPSQR (b = 400)	6.796 (0.006)
RPSQR (b = 500)	6.821 (0.007)

Table 14
The computation times (t) and number of selected variables (NSV) of the PLS, PQR, PSQR, and RPSQR estimators with $\tau = 0.2$, 0.5 and 0.8 under different bs for real data Example2.

	t			NSV		
Method	$\tau = 0.2$	$\tau = 0.5$	$\tau = 0.8$	$\tau = 0.2$	$\tau = 0.5$	$\tau = 0.8$
PQR	270.97	276.39	313.85	65	51	33
PSQR	22.53	21.93	28.75	65	50	33
RPSQR (b = 100)	15.72	14.08	14.43	60	52	33
RPSQR (b = 200)	14.16	14.28	14.24	65	52	32
RPSQR (b = 300)	14.97	14.52	14.82	64	53	30
RPSQR (b = 400)	15.09	14.99	15.18	65	50	31
RPSQR (b = 500)	15.97	15.29	15.57	75	49	30

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. The scale of the data to be stored is $(p+1)p$ (or p^2+p+1 for variable selection) instead of $N_b p$, which is the sample size of streaming data sets up to b batches. Because p is assumed to be a fixed number in this paper, our method greatly reduces the amount of data storage. Theoretically, the proposed estimators achieve optimal efficiency, and their asymptotic covariance matrixes are the same as those of the estimators with full data. Moreover, 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. As the proposed methods are all based on a convolution-type smoothing approach of the objective function, algorithms 1–3 are all fast and scalable.

From the numerical studies in Section 4, we can see that our proposed methods are very close to the estimators directly using all data, and better than other methods in existing reference by smaller MSE. The variable selection method can effectively select important variables. The proposed methods run fast and are faster than the full data estimators for large sample size and large dimension.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research is supported by the National Social Science Foundation of China (Series number: 21BTJ040).

Appendix A. Proof of main results

Proof of Theorem 2.1. Define a function

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

According to Eq. (2.10), the renewable estimator $\hat{\boldsymbol{\beta}}_b$ satisfies $G_b\left(\hat{\boldsymbol{\beta}}_b\right) = \mathbf{0}$.Under condition $n_1 \to \infty, \hat{\boldsymbol{\beta}}_1$ is $\sqrt{N_1}$-consistent (see the Eq. (2.11) in [11]). If $\left\{\hat{\boldsymbol{\beta}}_j\right\}_{j=1}^{b-1}$ are $\sqrt{N_j}$-consistent, we have $G_b(\boldsymbol{\beta}_0) = o_p(1)$.Thus, by the Lemma 4 in [11], we have

$$\begin{aligned} G_b\left(\hat{\boldsymbol{\beta}}_b\right) - G_b(\boldsymbol{\beta}_0) = & \frac{1}{N_b}\sum_{j=1}^{b-1} J\left(D_j;\hat{\boldsymbol{\beta}}_j;h_j\right)\left(\hat{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\right) \\ & + \frac{1}{N_b}\left\{U\left(D_b;\hat{\boldsymbol{\beta}}_b;h_b\right) - U(D_b;\boldsymbol{\beta}_0;h_b)\right\} \\ = & \frac{1}{N_b}\left\{\sum_{j=1}^{b-1} J\left(D_j;\hat{\boldsymbol{\beta}}_j;h_j\right) + J(D_b;\boldsymbol{\beta}_0;h_b)\right\} \\ & \left(\hat{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\right) + O_p\left(\frac{n_b}{N_b}\|\hat{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|_2^2\right) = o_p(1). \end{aligned} \tag{A.2}$$

From equations (A.1), (A.2) and $G_b\left(\hat{\boldsymbol{\beta}}_b\right) = \mathbf{0}$, we know that

$$G_b(\boldsymbol{\beta}_0) = \frac{1}{N_b}\left\{\sum_{j=1}^{b-1} J\left(D_j;\hat{\boldsymbol{\beta}}_j;h_j\right) + J(D_b;\boldsymbol{\beta}_0;h_b)\right\}\left(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_b\right) + O_p\left(\frac{n_b}{N_b}\|\hat{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|_2^2\right).$$

It follows that

$$\begin{aligned} & -\frac{1}{N_b}\left\{\sum_{j=1}^{b-1} J\left(D_j;\hat{\boldsymbol{\beta}}_j;h_j\right) + J(D_b;\boldsymbol{\beta}_0;h_b)\right\}\left(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_b\right) \\ & + \frac{1}{N_b}\sum_{j=1}^{b-1} J\left(D_j;\hat{\boldsymbol{\beta}}_j;h_j\right)\left(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_{b-1}\right) \\ & + \frac{1}{N_b}U(D_b;\boldsymbol{\beta}_0;h_b) + O_p\left(\frac{n_b}{N_b}\|\hat{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|_2^2\right) = \mathbf{0}. \end{aligned} \tag{A.3}$$

By $U\left(D_1;\hat{\beta}_1;h_1\right) = \mathbf{0}$, we have

$$\begin{aligned} U(D_1;\boldsymbol{\beta}_0;h_1) = & U\left(D_1;\hat{\beta}_1;h_1\right) + J\left(D_1;\hat{\beta}_1;h_1\right)\left(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_1\right) \\ & + O_p\left(n_1\|\hat{\beta}_1-\beta_0\|_2^2\right), \\ = & J\left(D_1;\hat{\beta}_1;h_1\right)\left(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_1\right) + O_p\left(n_1\|\hat{\beta}_1-\beta_0\|_2^2\right). \end{aligned} \tag{A.4}$$

By (2.9), we can obtain

$$\begin{aligned} U(D_2;\boldsymbol{\beta}_0;h_2) = & U\left(D_2;\hat{\beta}_2;h_2\right) + J\left(D_2;\hat{\beta}_2;h_2\right)\left(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_2\right) \\ & + O_p\left(n_2\|\hat{\beta}_2-\beta_0\|_2^2\right) \\ = & -J\left(D_1;\hat{\beta}_1;h_1\right)\left(\hat{\beta}_2-\hat{\beta}_1\right) + J\left(D_2;\hat{\beta}_2;h_2\right)\left(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_2\right) \\ & + O_p\left(n_2\|\hat{\beta}_2-\beta_0\|_2^2\right) \end{aligned} \tag{A.5}$$

Thus, combining (A.4) and (A.5),

$$\begin{aligned} U(D_1;\boldsymbol{\beta}_0;h_1) + U(D_2;\boldsymbol{\beta}_0;h_2) = & \left\{J\left(D_1;\hat{\beta}_1;h_1\right) + J\left(D_2;\hat{\beta}_2;h_2\right)\right\}\left(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_2\right) \\ & + O_p\left(n_1\|\hat{\beta}_1-\beta_0\|_2^2 + n_2\|\hat{\beta}_2-\beta_0\|_2^2\right). \end{aligned} \tag{A.6}$$

Similarly to equation (A.6), at the $(b-1)$-th data batch, it is easy to shown that

$$\begin{aligned} \sum_{j=1}^{b-1} U(D_j;\boldsymbol{\beta}_0;h_j) = & \sum_{j=1}^{b-1} J\left(D_j;\hat{\beta}_j;h_j\right)\left(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_{b-1}\right) \\ & + O_p\left(\sum_{j=1}^{b-1} n_j\|\hat{\beta}_j-\beta_0\|_2^2\right). \end{aligned} \tag{A.7}$$

Plugging equation (A.7) into equation (A.3), we get

$$-\frac{1}{N_b}\left\{\sum_{j=1}^{b-1}J\left(D_j;\hat{\boldsymbol{\beta}}_j;h_j\right)+J(D_b;\boldsymbol{\beta}_0;h_b)\right\}\left(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_b\right)+\frac{1}{N_b}\sum_{j=1}^{b}U(D_j;\boldsymbol{\beta}_0;h_j)$$
$$+O_p\left(\sum_{j=1}^{b}\frac{n_j}{N_b}\|\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_0\|_2^2\right)=\mathbf{0}. \tag{A.8}$$

By the Lemmas 1 and 4 in [11], under condition $n_j\to\infty, j=1,\dots,b$, we can obtain

$$\|J\left(D_j;\hat{\boldsymbol{\beta}};h_j\right)-J(D_j;\beta_0;h_j)\|=O_p\left(n_j\|\hat{\beta}_j-\beta_0\|_2\right),$$
$$\|J(D_j;\beta_0;h_j)-E\{J(D_j;\beta_0;h_j)\}\|=o_p\left(\sqrt{n_jh_j^{-1}\ln n_j}\right),$$
$$E\{J(D_j;\beta_0;h_j)\}=\sum_{i\in D_j}E\{f(\mathbf{X}_i^\top\boldsymbol{\beta}_0|\mathbf{X}_i)\mathbf{X}_i\mathbf{X}_i^\top\}+o_p(n_jh_j).$$

Since $\left\{\hat{\boldsymbol{\beta}}_j\right\}_{j=1}^{b-1}$ are consistent, and conditions $h_j(N_j/\ln N_j)^{1/3}\to\infty$ and $n_j\to\infty, j=1,\dots,b$, we have

$$\frac{1}{N_b}\left\{\sum_{j=1}^{b-1}J\left(D_j;\hat{\boldsymbol{\beta}}_j;h_j\right)+J(D_b;\boldsymbol{\beta}_0;h_b)\right\}=\Omega+o_p(1). \tag{A.9}$$

By the Lemma 1 and Theorem 5 in [11], we have

$$U(D_j;\beta_0;h_j)=U(D_j;\beta_0;h_b)+O_p\left(n_jh_j^2+n_jh_b^2\right). \tag{A.10}$$

Plugging (A.9) and (A.10) into equation (A.8), we can obtain

$$(\Omega+o_p(1))\left(\hat{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\right)+\frac{1}{N_b}\sum_{j=1}^{b}U(D_j;\boldsymbol{\beta}_0;h_b)+\frac{n_b}{N_b}O_p\left(\|\hat{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|_2^2\right)$$
$$+\frac{1}{N_b}\sum_{j=1}^{b-1}O_p\left(n_jh_j^2+n_jh_b^2+n_j\|\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_0\|_2^2\right)=\mathbf{0}.$$

By Lemma 3 in [12], we have

$$\sum_{j=1}^{b}\frac{n_j}{N_j}\leqslant 1+\log(N_b/N_1),\quad \text{and}\ \sum_{j=1}^{b}\frac{n_j}{\sqrt{N_j}}\leqslant 2\sqrt{N_b}.$$

Thus, by condition $h_j=o\left(N_j^{-1/4}\right)$ the central limit theorem, we can proof the theorem.

Lemma 1. *Suppose that conditions* **C1-C3** *hold and* $h(n/\ln n)^{1/3}\to\infty, h\leqslant O(n^{-1/4})$ *and* $\lambda\to 0$ *as the sample size* $n\to\infty$. *Then*

$$\|\tilde{\boldsymbol{\beta}}^*-\beta_0\|_2=O_p(n^{-1/2}).$$

Proof of Lemma 1. Denote $Q(\boldsymbol{\beta})=nS_h(\boldsymbol{\beta})+np_\lambda(|\boldsymbol{\beta}|)$. To prove Lemma 1, it is sufficient to show that for any given $\delta>0$, there exists a large enough constant C such that

$$P\left\{\inf_{\|\theta\|_2=C}Q(\beta_0+\theta/\sqrt{n})>Q(\beta_0)\right\}\geqslant 1-\delta, \tag{A.11}$$

which implies that with probability at least $1-\delta$ there exists a local minimum in the ball $\{\beta_0+\theta/\sqrt{n}:\|\theta\|_2\leqslant C\}$. This in turn implies that there exists a local minimizer such that $\|\tilde{\boldsymbol{\beta}}^*-\beta_0\|_2=O_p(n^{-1/2})$.Note that

$$Q(\beta_0+\boldsymbol{\theta}/\sqrt{n})-Q(\beta_0)\geqslant nS_h(\beta_0+\theta/\sqrt{n})-nS_h(\beta_0)$$
$$+n\sum_{j=1}^{s}\{p_\lambda(|\beta_{0j}+\theta_j/\sqrt{n}|)-p_\lambda(|\beta_{0j}|)\}, \tag{A.12}$$

where β_{0j} and θ_j denote the j-th component of β_0 and θ, respectively. Given any fixed θ, by the Lemma 1 in [11] and condition $h\to 0$, then

$$nS_h(\beta_0+\theta/\sqrt{n})-nS_h(\beta_0)=\sqrt{n}\left\{S_h^{(1)}(\beta_0)\right\}^\top\theta+\frac{1}{2}\theta^\top\Omega\theta+o_p(1). \tag{A.13}$$

Note that $\sqrt{n}S_h^{(1)}(\beta_0)=O_p(1)$. By choosing a sufficiently large C, the second term of (A.13) dominates $nS_h(\beta_0+\theta/\sqrt{n})-nS_h(\beta_0)$ uniformly in $\|\boldsymbol{\theta}\|=C$. Note that SCAD penalty is flat for coefficient of magnitude larger than $a\lambda$. Thus, by condition $\lambda\to 0$, we can obtain that

$$n\sum_{j=1}^{s}\{p_\lambda(|\beta_{0j}+\theta_j/\sqrt{n}|)-p_\lambda(|\beta_{0j}|)\}=0 \tag{A.14}$$

uniformly in any compact subset of R^p.

Based on (A.12)-(A.14), $Q(\beta_0+\theta/\sqrt{n})-Q(\beta_0)$ is dominated by the quadratic term $\boldsymbol{\theta}^\top\Omega\boldsymbol{\theta}/2$ for $\|\boldsymbol{\theta}\|_2$ equal to sufficiently large C. Hence equation (A.11) is satisfied.

Lemma 2. *Suppose that all conditions in Lemma 1 hold. If* $\sqrt{n}\lambda\to\infty$, *then with probability tending to one, the root-n consistent local minimizer* $\tilde{\boldsymbol{\beta}}^*=\left(\tilde{\boldsymbol{\beta}}_{(1)}^{*\top},\tilde{\boldsymbol{\beta}}_{(2)}^{*\top}\right)^\top$ *satisfies:*

(i) *Sparsity:* $\tilde{\boldsymbol{\beta}}_{(2)}^*=\mathbf{0}$, *and*
(ii) *Asymptotic normality:*

$$\sqrt{n}\left(\tilde{\boldsymbol{\beta}}_{(1)}^*-\beta_{01}\right)\xrightarrow{L}\mathcal{N}\left(0,\tau(1-\tau)\Omega_{(11)}^{-1}\boldsymbol{\Sigma}_{(11)}\Omega_{(11)}^{-1}\right).$$

Proof of Lemma 2. (i) The sparsity result comes from this claim: if $\lambda\to 0$ and $\sqrt{n}\lambda\to\infty$ as $n\to\infty$, then with probability tending to one, any given $\beta_{(1)}$ satisfying $\|\beta_{(1)}-\beta_{01}\|_2=O_p(n^{-1/2})$ and any constant C_1,

$$Q\left(\left(\boldsymbol{\beta}_{(1)}^\top,\mathbf{0}^\top\right)^\top\right)=\min_{\|\boldsymbol{\beta}_{(2)}\|_2\leqslant C_1n^{-1/2}}Q\left(\left(\boldsymbol{\beta}_{(1)}^\top,\boldsymbol{\beta}_{(2)}^\top\right)^\top\right),$$

that is to say for any given $\delta_1>0$,

$$P\left(\inf_{\|\boldsymbol{\beta}_{(2)}\|_2\leqslant C_1n^{-1/2}}Q\left(\left(\boldsymbol{\beta}_{(1)}^\top,\boldsymbol{\beta}_{(2)}^\top\right)^\top\right)>Q\left(\left(\boldsymbol{\beta}_{(1)}^\top,\mathbf{0}^\top\right)^\top\right)\right)\geqslant 1-\delta_1.$$

In fact, based on (A.13), for any $\|\beta_{(1)}-\beta_{01}\|_2=O_p(n^{-1/2})$ and $\|\beta_{(2)}\|_2\leqslant C_1n^{-1/2}$, we have

$$\begin{aligned}
&Q\left(\left(\boldsymbol{\beta}_{(1)}^\top,\mathbf{0}^\top\right)^\top\right)-Q\left(\left(\boldsymbol{\beta}_{(1)}^\top,\boldsymbol{\beta}_{(2)}^\top\right)^\top\right)\\
=&\left\{Q\left(\left(\boldsymbol{\beta}_{(1)}^\top,\mathbf{0}^\top\right)^\top\right)-Q\left((\boldsymbol{\beta}_{01}^\top,\mathbf{0}^\top)^\top\right)\right\}\\
&-\left\{Q\left(\left(\boldsymbol{\beta}_{(1)}^\top,\boldsymbol{\beta}_{(2)}^\top\right)^\top\right)-Q\left((\boldsymbol{\beta}_{01}^\top,\mathbf{0}^\top)^\top\right)\right\}\\
=&n\left\{S_h\left(\left(\boldsymbol{\beta}_{(1)}^\top,\mathbf{0}^\top\right)^\top\right)-S_h\left((\boldsymbol{\beta}_{01}^\top,\mathbf{0}^\top)^\top\right)\right\}\\
&-n\left\{S_h\left(\left(\boldsymbol{\beta}_{(1)}^\top,\boldsymbol{\beta}_{(2)}^\top\right)^\top\right)-S_h\left((\boldsymbol{\beta}_{01}^\top,\mathbf{0}^\top)^\top\right)\right\}-n\sum_{j=s+1}^{p}p_\lambda(|\beta_j|)\\
=&-n\sum_{j=s+1}^{p}p_\lambda(|\beta_j|)+O_p(1).
\end{aligned}$$

Note that

$$
\begin{aligned}
n\sum_{j=s+1}^{p} p_\lambda(|\beta_j|) \geqslant\ & n\sum_{j=s+1}^{p}\left\{\lambda \liminf_{\lambda\to 0}\liminf_{\mu\to 0^+}\tfrac{p'_\lambda(\mu)}{\lambda}\beta_j sign(\beta_j)+o(|\beta_j|)\right\}\\
=\ & n\lambda\left\{\liminf_{\lambda\to 0}\liminf_{\mu\to 0^+}\tfrac{p'_\lambda(\mu)}{\lambda}\right\}\sum_{j=s+1}^{p}|\beta_j|\{1+o(1)\} \qquad (A.15)\\
=\ & n\lambda\sum_{j=s+1}^{p}|\beta_j|\{1+o(1)\},
\end{aligned}
$$

where the last step follows based on the fact $\liminf_{\lambda\to 0}\liminf_{\mu\to 0^+}p'_\lambda(\mu)/\lambda=1$. Since, $\sqrt{n}\lambda\to\infty$ and $\|\beta_{(2)}\|_2\leqslant C_1 n^{-1/2}, Q\left(\begin{pmatrix}\beta_{(1)}\\ \mathbf{0}\end{pmatrix}\right)-Q\left(\begin{pmatrix}\beta_{(1)}\\ \beta_{(2)}\end{pmatrix}\right)$ is dominated by $-n\sum_{j=s+1}^{p}p_\lambda(|\beta_j|)$, as a result, $Q\left(\begin{pmatrix}\beta_{(1)}\\ \mathbf{0}\end{pmatrix}\right)-Q\left(\begin{pmatrix}\beta_{(1)}\\ \beta_{(2)}\end{pmatrix}\right)<0$ for large n. This completes the proof of part (i) of the theorem.

(ii) From Lemma 1 and part (i), we know that $\tilde{\boldsymbol{\beta}}^*_{(1)}$ is a root-n consistent local minimizer of $Q\left(\begin{pmatrix}\boldsymbol{\beta}_{(1)}\\ \mathbf{0}\end{pmatrix}\right)$, which is regarded as a function of $\boldsymbol{\beta}_{(1)}$, and that satisfies

$$
\partial Q(\boldsymbol{\beta})/\partial\boldsymbol{\beta}_j\big|_{\boldsymbol{\beta}=\left(\tilde{\boldsymbol{\beta}}^{*\top}_{(1)},\mathbf{0}^\top\right)^\top}=0,\ j=1,\ldots,s. \qquad (A.16)
$$

Note that $\tilde{\boldsymbol{\beta}}^*_{(1)}$ is also the minimizer of

$$
\begin{aligned}
& Q\left(\left(\boldsymbol{\beta}^\top_{(1)},\mathbf{0}^\top\right)^\top\right)-Q\left(\left(\boldsymbol{\beta}^\top_{01},\mathbf{0}^\top\right)^\top\right)\\
=\ & nS_h\left(\left(\boldsymbol{\beta}^\top_{(1)},\mathbf{0}^\top\right)^\top\right)-nS_h\left(\left(\boldsymbol{\beta}^\top_{01},\mathbf{0}^\top\right)^\top\right)\\
& +n\sum_{j=1}^{s}\{p_\lambda(|\beta_j|)-p_\lambda(|\beta_{0j}|)\}\\
=\ & n\left(\left(\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\right)^\top,\mathbf{0}^\top\right)S_h^{(1)}(\beta_0)\\
& +\tfrac{n}{2}\left(\left(\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\right)^\top,\mathbf{0}^\top\right)\Omega\left(\left(\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\right)^\top,\mathbf{0}^\top\right)^\top+o_p(1)\\
& +n\sum_{j=1}^{s}\left\{p'_\lambda(|\beta_{0j}|)sign(\beta_{0j})(\beta_j-\beta_{0j})+\tfrac{1}{2}p''_\lambda(|\beta_{0j}|)(\beta_j-\beta_{0j})^2(1+o_p(1))\right\}\\
=\ & n\left(\left(\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\right)^\top,\mathbf{0}^\top\right)S_h^{(1)}(\beta_0)\\
& +\tfrac{n}{2}\left(\left(\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\right)^\top,\mathbf{0}^\top\right)\Omega\left(\left(\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\right)^\top,\mathbf{0}^\top\right)^\top\\
& +o_p(1)+o_p\left(n\|\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\|_2^2\right),
\end{aligned} \qquad (A.17)
$$

where the last equation is because $p'_\lambda(|\boldsymbol{\beta}_{0j}|)sign(\boldsymbol{\beta}_{0j})=0$ and $p''_\lambda(|\boldsymbol{\beta}_{0j}|)\to 0$, $j=1,\ldots,s$, by condition $\lambda\to 0$. Thus, based on (A.16) and (A.17), by Slutsky's theorem and the central limit theorem, we can prove the part (ii).

Proof of Theorem 3.1. The renewable penalized estimator $\tilde{\boldsymbol{\beta}}_b$ also satisfies

$$
\tilde{\boldsymbol{\beta}}_b=\arg\min_{\boldsymbol{\beta}}\widetilde{Q}_b(\boldsymbol{\beta}),
$$

where $\widetilde{Q}_b(\boldsymbol{\beta})=\frac{1}{2}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_{b-1})^\top\widetilde{J}_{b-1}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_{b-1})+S_{h_b}(D_b;\boldsymbol{\beta})-N_{b-1}p'_{\lambda_{b-1}}(|\tilde{\boldsymbol{\beta}}_{b-1}|)sign(\tilde{\boldsymbol{\beta}}_{b-1})(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_{b-1})+N_b p_{\lambda_b}(|\boldsymbol{\beta}|)$.Similar to the proof of Lemma 1, it is sufficient to show that for any given $\tilde{\delta}>0$, there exists a large enough constat $\widetilde{C}$ such that

$$
P\left\{\inf_{\|\tilde{\theta}\|_2=\widetilde{C}}\widetilde{Q}_b\left(\beta_0+\tilde{\theta}/\sqrt{N_b}\right)>\widetilde{Q}_b(\beta_0)\right\}\geqslant 1-\tilde{\delta}. \qquad (A.18)
$$

It implies that with probability at least $1-\tilde{\delta}$ there exists a local minimizer satisfying $\|\tilde{\boldsymbol{\beta}}_b-\beta_0\|_2=O_p\left(N_b^{-1/2}\right)$.By the Lemma 1, $\|\tilde{\boldsymbol{\beta}}_1-\beta_0\|_2=O_p\left(n_1^{-1/2}\right)$, if $\|\tilde{\boldsymbol{\beta}}_j-\beta_0\|_2=O_p\left(N_j^{-1/2}\right)$ with $j=1,\ldots,b-1$, and by $\|\tilde{\theta}\|_2=\widetilde{C}$, we have

$$
\begin{aligned}
& \widetilde{Q}_b\left(\beta_0+\tilde{\theta}/\sqrt{N_b}\right)-\widetilde{Q}_b(\beta_0)\\
=\ & (\boldsymbol{\beta}_0-\tilde{\boldsymbol{\beta}}_{b-1})^\top\widetilde{J}_{b-1}\tilde{\theta}/\sqrt{N_b}+\tfrac{1}{2}\tilde{\theta}^\top\widetilde{J}_{b-1}\tilde{\theta}/N_b+S_{h_b}\left(D_b;\beta_0+\tilde{\theta}/\sqrt{N_b}\right)\\
& -S_{h_b}(D_b;\boldsymbol{\beta}_0)\\
& +N_b\left\{p_{\lambda_b}\left(|\beta_0+\tilde{\theta}/\sqrt{N_b}|\right)-p_{\lambda_b}(|\boldsymbol{\beta}_0|)\right\}\\
& -N_{b-1}p'_{\lambda_{b-1}}(|\tilde{\boldsymbol{\beta}}_{b-1}|)sign(\tilde{\boldsymbol{\beta}}_{b-1})\tilde{\theta}/\sqrt{N_b}\\
=\ & \tfrac{1}{2}\tilde{\theta}^\top\Omega\tilde{\theta}+O_p(1)+N_b\left\{p_{\lambda_b}\left(|\beta_0+\tilde{\theta}/\sqrt{N_b}|\right)-p_{\lambda_b}(|\boldsymbol{\beta}_0|)\right\}\\
& -N_{b-1}p'_{\lambda_{b-1}}(|\tilde{\boldsymbol{\beta}}_{b-1}|)sign(\tilde{\boldsymbol{\beta}}_{b-1})\tilde{\theta}/\sqrt{N_b}\\
\geqslant\ & \tfrac{1}{2}\tilde{\theta}^\top\Omega\tilde{\theta}+O_p(1).
\end{aligned} \qquad (A.19)
$$

Thus $\widetilde{Q}_b\left(\beta_0+\tilde{\theta}/\sqrt{N_b}\right)-\widetilde{Q}_b(\beta_0)$ is dominated by the quadratic term $\tilde{\theta}^\top\Omega\tilde{\theta}/2$ for $\|\tilde{\boldsymbol{\theta}}\|$ equal to sufficiently large $\widetilde{C}$. Hence equation (A.18) is satisfied.

Proof of Theorem 3.2. (i) For any $\|\beta_{(1)}-\beta_{01}\|_2=O_p\left(N_b^{-1/2}\right)$ and $\|\beta_{(2)}\|_2\leqslant\widetilde{C}_1N_b^{-1/2}$, by (A.19), we have

$$
\begin{aligned}
& \widetilde{Q}_b\left(\left(\boldsymbol{\beta}^\top_{(1)},\mathbf{0}^\top\right)^\top\right)-\widetilde{Q}_b\left(\left(\boldsymbol{\beta}^\top_{(1)},\boldsymbol{\beta}^\top_{(2)}\right)^\top\right)\\
=\ & \left\{\widetilde{Q}_b\left(\left(\boldsymbol{\beta}^\top_{(1)},\mathbf{0}^\top\right)^\top\right)-\widetilde{Q}_b\left(\left(\boldsymbol{\beta}^\top_{01},\mathbf{0}^\top\right)^\top\right)\right\}\\
& -\left\{\widetilde{Q}_b\left(\left(\boldsymbol{\beta}^\top_{(1)},\boldsymbol{\beta}^\top_{(2)}\right)^\top\right)-\widetilde{Q}_b\left(\left(\boldsymbol{\beta}^\top_{01},\mathbf{0}^\top\right)^\top\right)\right\}\\
=\ & -N_b\sum_{k=s+1}^{p}p_{\lambda_b}(|\boldsymbol{\beta}_k|)+N_{b-1}\sum_{k=s+1}^{p}p'_{\lambda_{b-1}}(|\tilde{\boldsymbol{\beta}}_{b-1,k}|)sign(\tilde{\boldsymbol{\beta}}_{b-1,k})\boldsymbol{\beta}_k+O_p(1).
\end{aligned}
$$

By condition $\|\beta_{(2)}\|_2\leqslant\widetilde{C}_1N_b^{-1/2}$, we have $N_{b-1}\sum_{k=s+1}^{p}p'_{\lambda_{b-1}}(|\tilde{\boldsymbol{\beta}}_{b-1,k}|)sign(\tilde{\boldsymbol{\beta}}_{b-1,k})\boldsymbol{\beta}_k=o_p\left(\sqrt{N_{b-1}/N_b}\right)$. And similar to (A.15), we can obtain $N_b\sum_{k=s+1}^{p}p_{\lambda_b}(|\beta_k|)\geqslant N_b\lambda_b\sum_{k=q+1}^{p}|\beta_k|\{1+o(1)\}$. Since, $\sqrt{N_b}\lambda_b\to\infty$ and $\|\beta_{(2)}\|_2\leqslant\widetilde{C}_1N_b^{-1/2}$, $\widetilde{Q}_b\left(\left(\boldsymbol{\beta}^\top_{(1)},\mathbf{0}^\top\right)^\top\right)-\widetilde{Q}_b\left(\left(\boldsymbol{\beta}^\top_{(1)},\boldsymbol{\beta}^\top_{(2)}\right)^\top\right)$ is dominated by $-N_b\sum_{k=s+1}^{p}p_{\lambda_b}(|\beta_k|)$, as a result, $\widetilde{Q}_b\left(\left(\boldsymbol{\beta}^\top_{(1)},\mathbf{0}^\top\right)^\top\right)-\widetilde{Q}_b\left(\left(\boldsymbol{\beta}^\top_{(1)},\boldsymbol{\beta}^\top_{(2)}\right)^\top\right)<0$ for large N_b. This completes the proof of part (i) of the theorem.

(ii) From Theorem 3.1 and part (i), we know that $\tilde{\boldsymbol{\beta}}_{(b1)}$ is a root-N_b consistent local minimizer of $\widetilde{Q}_b\left(\left(\boldsymbol{\beta}^\top_{(1)},\mathbf{0}^\top\right)^\top\right)$, which is regarded as a function of $\boldsymbol{\beta}_{(1)}$, and that satisfies

$$
\partial\widetilde{Q}_b(\boldsymbol{\beta})/\partial\beta_k\big|_{\boldsymbol{\beta}=\left(\tilde{\boldsymbol{\beta}}^\top_{(b1)},\mathbf{0}^\top\right)^\top}=0,\ k=1,\ldots,s. \qquad (A.20)
$$

Note that $\tilde{\boldsymbol{\beta}}_{(b1)}$ is also the minimizer of $\widetilde{Q}_b\left(\left(\boldsymbol{\beta}^\top_{(1)},\mathbf{0}^\top\right)^\top\right)-\widetilde{Q}_b\left(\left(\boldsymbol{\beta}^\top_{01},\mathbf{0}^\top\right)^\top\right)$. By the Theorem 3.1, $\|\tilde{\boldsymbol{\beta}}_j-\beta_0\|_2=O_p\left(N_j^{-1/2}\right)$ and conditions $h_j=o\left(N_j^{-1/4}\right),j=1,\ldots,b$. Then, by (A.7) and (A.10), we have

$$
\begin{aligned}
&\widetilde{Q}_b\left(\left(\boldsymbol{\beta}_{(1)}^{\top},\mathbf{0}^{\top}\right)^{\top}\right)-\widetilde{Q}_b\left(\left(\boldsymbol{\beta}_{01}^{\top},\mathbf{0}^{\top}\right)^{\top}\right)\\
=\;&\frac{1}{2}\left(\left(\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\right)^{\top},\mathbf{0}^{\top}\right)\left\{\widetilde{J}_{b-1}+J\left(D_b;\left(\boldsymbol{\beta}_{01}^{\top},\mathbf{0}^{\top}\right)^{\top};h_b\right)\right\}\left(\left(\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\right)^{\top},\mathbf{0}^{\top}\right)^{\top}\\
&+\left\{\sum_{j=1}^{b}U\left(D_b;\left(\boldsymbol{\beta}_{01}^{\top},\mathbf{0}^{\top}\right)^{\top};h_b\right)+o_p\left(\sqrt{N_b}\right)\right\}\left(\left(\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\right)^{\top},\mathbf{0}^{\top}\right)\\
&-N_{b-1}\sum_{k=1}^{s}p\prime_{\lambda_{b-1}}\left(|\tilde{\boldsymbol{\beta}}_{b-1,k}|\right)sign\left(\tilde{\boldsymbol{\beta}}_{b-1,k}\right)\left(\boldsymbol{\beta}_k-\boldsymbol{\beta}_{0,k}\right)\\
&+o_p\left(N_b\|\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\|_2^2\right)+O_p\left(n_b\|\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\|_2^2\right)+C_2\\
=\;&\frac{1}{2}\left(\sqrt{N_b}\left(\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\right)^{\top},\mathbf{0}^{\top}\right)\Omega\left(\sqrt{N_b}\left(\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\right)^{\top},\mathbf{0}^{\top}\right)^{\top}\\
&+\frac{1}{\sqrt{N_b}}\sum_{j=1}^{b}U\left(D_b;\left(\boldsymbol{\beta}_{01}^{\top},\mathbf{0}^{\top}\right)^{\top};h_b\right)\left(\sqrt{N_b}\left(\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\right)^{\top},\mathbf{0}^{\top}\right)\\
&+\left(\sqrt{N_b}\left(\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\right)^{\top},\mathbf{0}^{\top}\right)o_p(1)+o_p\left(N_b\|\boldsymbol{\beta}_{(1)}-\boldsymbol{\beta}_{01}\|_2^2\right)+C_2
\end{aligned}
\tag{A.21}
$$

Thus, based on (A.20) and (A.21), by Slutsky's theorem and the central limit theorem, we can prove the part (ii).

References

[1] X. Chen, W. Liu, X. Mao, Z. Yang, Distributed high-dimensional regression under a quantile loss function, J. Mach. Learn. Res. 21 (2020) 1–43.
[2] X. Chen, W. Liu, Y. Zhang, Quantile regression under memory constraint, Ann. Stat. 47 (2019) 3244–3273.
[3] A. Davison, D. Hinkley, Bootstrap methods and their application, Cambridge University Press, Cambridge, 1997.
[4] G. De Francisci Morales, A. Bifet, Samoa: Scalable advanced massive online analysis, J. Mach. Learn. Res. 16 (2015) 149–153.
[5] Deshpande, Y., Javanmard, A., Mehrabi, M., 2020. Online debiasing for adaptively collected high-dimensional data with applications to time series analysis. arXiv:1911.01040v3.
[6] A. Eftekhari, G. Ongie, L. Balzano, M.B. Wakin, Streaming principal component analysis from incomplete data, J. Mach. Learn. Res. 20 (2019) 1–62.
[7] Fan, J., Gong, W., Li, C.J., Sun, Q., 2018a. Statistical sparse online regression: a diffusion approximation perspective. Proceedings of the 21st International Conference on Artificial Intelligence and Statistics 84, 1017–1026.
[8] J. Fan, R. Li, Variable selection via nonconcave penalized likelihood and its oracle properties, J. Am. Stat. Assoc. 96 (2001) 1348–1360.
[9] J. Fan, H. Liu, Q. Sun, T. Zhang, I-lamm for sparse learning: Simultaneous control of algorithmic complexity and statistical error, Ann. Stat. 96 (2018) 1348–1360.
[10] J. Fan, L. Xue, H. Zou, Strong oracle optimality of folded concave penalized estimation, Ann. Stat. 42 (2014) 819–849.
[11] M. Fernandes, E. Guerre, E. Horta, Smoothing quantile regressions, J. Bus. Econ. Stat. 39 (2021) 338–357.
[12] Han, R., Luo, L., Lin, Y., Huang, J., 2021. Online debiased lasso. arXiv:2106.05925v1.
[13] He, X., Pan, X., Tan, K.M., Zhou, W., 2020. Smoothed quantile regression with large scale inference. arXiv: Statistics Theory.
[14] M. Hilbert, Big data for development: a review of promises and challenges, Development Policy Review 34 (2016) 135–174.
[15] J. Horowitz, Bootstrap methods for median regression models, Econometrica 66 (1998) 1327–1352.
[16] A. Hu, Y. Jiao, Y. Liu, Y. Shi, Y. Wu, Distributed quantile regression for massive heterogeneous data, Neurocomputing 448 (2021) 249–262.
[17] R. Jiang, K. Yu, Smoothing quantile regression for a distributed system, Neurocomputing 466 (2021) 311–326.
[18] M. Jordan, J. Lee, Y. Yang, Communication-efficient distributed statistical learning, J. Am. Stat. Assoc. 14 (2019) 668–681.
[19] R. Koenker, Quantile regression, Cambridge University Press, Cambridge, 2005.
[20] R. Koenker, G. Bassett, Regression quantile, Econometrica 46 (1978) 33–50.
[21] J. Lee, H. Wang, E. Schifano, Online updating method to correct for measurement error in big data streams, Comput. Stat. Data Anal. 149 (2020) 106976.
[22] Lin, L., Li, W., Lu, J., 2020. Unified rules of renewable weighted sums for various online updating estimations.
[23] N. Lin, R. Xi, Aggregated estimating equation estimation, Statistics and Its Interface 4 (2011) 73–83.
[24] L. Luo, P. Song, Renewable estimation and incremental inference in generalized linear models with streaming data sets, J. Roy. Stat. Soc. B 82 (2020) 69–97.
[25] Ma, X., Lin, L., Gai, Y., 2021. A general framework of online updating variable selection for generalized linear models with streaming datasets. arXiv:2101.08639v1
[26] S. Mohamad, A. Bouchachia, Deep online hierarchical dynamic unsupervised learning for pattern mining from utility usage data, Neurocomputing 390 (2020) 359–373.
[27] E. Schifano, J. Wu, C. Wang, J. Yan, M.-H. Chen, Online updating of statistical inference in the big data setting, Technometrics 58 (2016) 393–403.
[28] C. Shi, R. Song, W. Lu, R. Li, Statistical inference for high-dimensional models via recursive online-score estimation, J. Am. Stat. Assoc. 116 (2021) 1307–1318.
[29] Sun, L., Wang, M., Guo, Y., Barbu, A., 2020. A novel framework for online supervised learning with feature selection. arXiv:1803.11521v7.
[30] R. Tibshirani, Regression shrinkage and selection via the lasso, J. Roy. Stat. Soc. B 58 (1996) 267–288.
[31] C. Wang, M.-H. Chen, E. Schifano, J. Wu, J. Yan, Statistical methods and computing for big data, Statistics and Its Interface 9 (2016) 399–414.
[32] C. Wang, M.-H. Chen, J. Wu, J. Yan, Y. Zhang, E. Schifano, Online updating method with new variables for big data streams, Can. J. Stat. 46 (2018) 123–146.
[33] H. Wang, R. Li, C.-L. Tsai, Tuning parameter selectors for the smoothly clipped absolute deviation method, Biometrika 94 (2007) 553–568.
[34] H. Wang, Y. Ma, Optimal subsampling for quantile regression in big data, Biometrika 108 (2021) 99–112.
[35] H. Wang, R. Zhu, P. Ma, Optimal subsampling for large sample logistic regression, J. Am. Stat. Assoc. 113 (2018) 829–844.
[36] K. Wang, H. Wang, S. Li, Renewable quantile regression for streaming datasets, Knowl.-Based Syst. 235 (2021) 107675.
[37] J. Wu, M.H. Chen, E. Schifano, J. Yan, Online updating of survival analysis, J. Comput. Graph. Stat. (2021) 1–35.
[38] Y. Wu, Y. Liu, Variable selection in quantile regression, Statistica Sinica 19 (2009) 801–817.
[39] Y. Xue, H. Wang, J. Yan, E. Schifano, An online updating approach for testing the proportional hazards assumption with streams of survival data, Biometrics 76 (2020) 171–182.
[40] J. Yu, H. Wang, M. Ai, H. Zhang, Optimal distributed subsampling for maximum quasi-likelihood estimators with massive data, J. Am. Stat. Assoc. (2020) 1–12.
[41] C.-H. Zhang, Nearly unbiased variable selection under minimax concave penalty, Ann. Stat. 38 (2010) 894–942.
[42] H. Zou, The adaptive lasso ad its oracle properties, J. Am. Stat. Assoc. 101 (2006) 1418–1429.

Dr. Rong Jiang received the B.S and Ph.D. degrees from Tongji University, China, in 2009 and 2014, respectively. He is currently an Associate Professor with the School of Science, Donghua University, China. His research interests include high-dimensional data analysis, distributed algorithm for large-scale data, and statistical computing.

Professor Keming Yu received his first degree in Mathematics and MSc in Statistics from universities in China and then his PhD in Statistics from The Open University, Milton Keynes, UK. Currently he is the Chair in Statistics in Brunel University London. His research interests include regression models for big and small data with applications, nonparametric smoothing and applied Bayesian analysis, machines learning and statistical reliability with applications.