Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 9(9): 26114–26138.
DOI: 10.3934/math.20241276
Received: 07 June 2024
Revised: 21 August 2024
Accepted: 29 August 2024
Published: 10 September 2024

Research article

Bayesian quantile regression for streaming data

Zixuan Tian[1,2,#], Xiaoyue Xie[3,#] and Jian Shi[1,2,*]

[1] Academy of Mathematics and Systems Science, Chinese Academy of Science, Beijing, China
[2] School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China
[3] Equipment Management and UAV Engineering, Air Force Engineering University, Xi'an, China
[#] These authors contributed equally to this work.

* **Correspondence:** Email: jshi@iss.ac.cn.

Abstract: Quantile regression has been widely used in many fields because of its robustness and comprehensiveness. However, it remains challenging to perform the quantile regression (QR) of streaming data by a conventional methods, as they are all based on the assumption that the memory can fit all the data. To address this issue, this paper proposes a Bayesian QR approach for streaming data, in which the posterior distribution was updated by utilizing the aggregated statistics of current and historical data. In addition, theoretical results are presented to confirm that the streaming posterior distribution is theoretically equivalent to the orcale posterior distribution calculated using the entire dataset together. Moreover, we provide an algorithmic procedure for the proposed method. The algorithm shows that our proposed method only needs to store the parameters of historical posterior distribution of streaming data. Thus, it is computationally simple and not storage-intensive. Both simulations and real data analysis are conducted to illustrate the good performance of the proposed method.

Keywords: Bayesian modeling; big data; quantile regression; streaming data
Mathematics Subject Classification: 62F15, 62G08, 68W27

1. Introduction

The term "big data" may have different meanings in different fields, so it has thus become a dominant topic in almost all academic and application fields. In a more general sense, big data encapsulates aspects such as quantity, diversity, speed, variability, and accuracy [1]. However, applying statistical models and methods to big data often entails a substantial computational burden, straining computer memory due to its large size and impacting computational efficiency, in that even seemingly simple tasks can demand considerable computation time. In order to solve the dilemma of big data analysis, some researchers have proposed three methods [2]: Subsampling-based approaches [3, 4], divide and conquer approaches [5–8], and online updating approaches [9–14]. Differing from the other two methods, online updating approaches focus on streaming data and address statistical problems in an updating framework without storage requirements for historical data. In the era of big data, streaming datasets have gradually appeared in various fields, including investment analysis, medical imaging, and computervision, making streaming data analysis increasingly useful and vital across various industries. For instance, financial institutions track the changes of the stock market in realtime, calculate the value at risk, and automatically rebalance the portfolio according to the stock price movements. Due to the increasing demand for stream processing, online updating became more particularly appealing with its ability to quickly process huge volumes of data so that organisa tions or businesses can react to changing conditions in realtime.

In the framework for regression-type analysis, Shi et al. [11] developed online updating algorithms for linear models and estimating equations. However, theoretical gurantees of these methods have been established based on some strong regularity conditions. Mohamad and Bouchachia [13] proposed a renewable estimation for the generalized linear model, which overcame the aforementioned limitations. Lin et al. [15] introduced a general framework to execute renewable weighted sum estimation in various online updates. Wang et al. [16] expanded the scope of online update methods by adapting to new predictive variables midway through the data flow. Wu and Chen [17] developed an online update method for survival analysis under the Cox proportional hazard model. Furthermore, Xue et al. [18] proposed a online updating approach to evaluate the proportional hazard hypothesis. Balakrishnan and Madigan [19] proposed a one-pass streaming algorithm for Bayesian regression. Based on sketch drawing technology and random projection, Geppert et al. [20] improved the streaming algorithm for Bayesian regression proposed in [19]. All of the above methods and algorithms are applicable in practice. However, they mainly focus on mean regression, or are built upon likelihood framework. Moreover, it is noteworthy that mean regression and likelihood methods still have limitations in robustness, and are very sensitive to outliers or heavy-tailed distributions.

As a natural upgrade solution, quantile regression (QR) [21] has been used as a feasible alternative to mean regression and has become a popular data analysis strategy. Quantile regression does not simply focus on the average value of the distribution but considers the entire conditional distribution, thus being more comprehensive. In addition, it is also robust to heavy-tailed distributions and outliers. Because of these advantages, quantile regression has a wide range of applications, namely in health research [22], longitudinal data analysis [23, 24], economics [25], and machine learning [26, 27]. However, because the datasets arrive continuously in an unbounded stream and the computer memory is usually limited, the traditional quantile regression method cannot retain them when processing stream data. Therefore, the traditional quantile regression strategy, which assumes that the computer memory can adapt to all of the datasets, is no longer applicable. In addition, the estimator obtained by quantile regression has no closed form, and the quantile regression loss function is also nondifferentiable. This also leads to the fact that streaming data analysis methods cannot be directly applied to quantile regression; see Chen et al. [28].

To circumvent the nondifferentiability of the QR loss function, Chen et al. [28] introduced Horowitz's smoothed QR, involving smoothing the indicator part of the check function through a kernel smoothing survival function. Additionally, Chen and Zhou [29] proposed online QR strategies, employing a smooth function to approximate the nondifferentiable check loss function. The method in [28] introduced a linear-type estimator for QR, and its calculation process needs to select bandwidth parameters through data-driven criteria. Furthermore, the approach in [29] relied on the Taylor expansion of the estimating equations, requiring the estimation of the unknown conditional density function of the response variable. More recently, Wang et al. [30] developed a renewable QR estimation strategy for streaming datasets, which renews the estimator with current data and summarized statistics of historical data rather than historical raw data. In summary, the above methods have effectively solved the dilemma of quantile regression in the case of streaming data, but these methods focus on the solution of frequentist rather than Bayesian. In the field of Bayesian streaming data, current research work is mainly focused on Bayesian flow networks (BFN) [48, 49]. Rezende and Mohamed [48] introduced the concept of normalizing flows, which is a key component of Bayesian flow networks. Müller and Quintana [49] provided a comprehensive introduction to the application of Bayesian nonparametric models in data analysis and inference, including streaming data. BFN is a generative model that integrates the advantages of Bayesian inference and deep learning. It has received extensive attention in recent years in areas such as data generation, image generation, and text generation. Bayesian flow networks particularly excel in handling discrete data generation problems, as they can generate continuous and differentiable samples, providing a new solution for discrete data generation. BFN combines the streaming models of deep learning with Bayesian inference, enabling the model to adapt to both continuous and discrete data, and enhancing the interpretability and flexibility of generative models. In tasks such as image generation and text generation, BFN has achieved good experimental results, demonstrating its tremendous potential in practical applications. Therefore, it is interesting to explore a more convenient Bayesian quantile regression strategy for streaming datasets. To the best of our knowledge, there is little related work on Bayesian quantile regression with streaming data.

In this paper, in contrast to the above methods, we develop a new idea for the analysis of streaming data using the Bayesian quantile model, which renews the posterior distribution with current data and summary statistics of historical data, instead of historical raw data. Specifically, we only need to retain the posterior distribution parameters of previous data and a likehood function for current data stream, and the historical data can be completely discarded. The Bayesian streaming quantile regression (BSQR) model is formulated for streaming data through a conjugate normal-inverse-gamma (NIG) prior distribution. Leveraging the posterior distribution of historical data as the prior for current data stream, a posterior distribution is established without any loss of information. Theoretically, we prove that the Bayesian streaming quantile regression estimator is equivalent to the global oracle estimator calculated based on the entire data. Numerical experiments on both synthetic and real data are also included to validate the theoretical results and illustrate the good performance of our new method.

The rest of the paper is organized as follows. Section 2 presents the introduction of BQR method. Section 3 gives the detailed expressions of NIG prior and posterior distributions for BQR using an informative g-prior. The BSQR method is developed in Section 4. Sections 5 and 6 validate the obtained theoretical results and illustrate the good performance of the proposed BSQR approach on synthetic and real data. Section 7 contains the discussion and conclusions. In addition, the proof of Theorem 1 is provided in the appendix.

2. Bayesian quantile regression model

2.1. Quantile regression based on AL distribution

We consider the following linear quantile regression model

$$y_i = \boldsymbol{x}_i^T \beta + e_i, \tag{2.1}$$

where y_i is a continuous response variable and $\boldsymbol{x}_i$ is an $k \times 1$ vector of predictors for the i-th observation, $i = 1, \cdots, n$. β is an $k \times 1$ vector of unknown parameter of interest, e_i is the error term, and its distribution is assumed to have zero τ-th quantile.

Now, consider a linear conditional quantile function as follows:

$$Q_\tau(y_i|\boldsymbol{x}_i) = \boldsymbol{x}_i^T \beta_\tau, \tag{2.2}$$

where $Q_\tau(\cdot) = inf\{y : F_{y_i}(y|x_i) \geq \tau\}$ denotes the inverse of the cumulative distribution function of the response conditional on x_i. In τ-th quantile regression, the estimation for β can be obtained by minimizing

$$\min_{\beta \in \boldsymbol{R}^p} \sum_{i=1}^{n} \rho_\tau(y_i - \boldsymbol{x}_i^T \beta), \tag{2.3}$$

where $\rho_\tau(u) = u(\tau - I(u \leq 0))$ denotes the check function.

Koenker and Machado [50] showed that $\rho_\tau(u)$ is exactly matched to the asymmetric Laplace distribution (ALD), whose density function is given by

$$f(y|\mu, \sigma, \tau) = \frac{\tau(1-\tau)}{\sigma} \exp\{-\frac{1}{\sigma}\rho_\tau(y - \mu)\}, \tag{2.4}$$

where μ is the location parameter and σ is the scale parameter.

Assume that errors $\{e_i\}_{i=1}^n$ are i.i.d $ALD(0, \sigma, \tau)$, the likehood of y_i is

$$f(y_i|\beta, \sigma, \tau) = \frac{\tau(1-\tau)}{\sigma} \exp\{-\frac{1}{\sigma}\rho_\tau(y_i - \boldsymbol{x}_i^T \beta)\}. \tag{2.5}$$

According to Koenker and Machado [50], minimizing (2.3) is equivalent to maximizing the above likelihood function (2.5). Yu and Moyeed [51] proposed a Bayesian method for QR by assuming an ALD on the response as a working likelihood. Over the past few years, this working likelihood function has been used in Geraci and Geraci and Bottai [52, 53]; among others. So, it is a natural choice to use the ALD error distribution in Bayesian quantile models; "we believe that the ALD approach is a valuable and relatively simple alternative to these other methods that does not need complex choices of prior distributions and prior parameters"; see Section 3 of Benoit et al. [54].

However, direct use of the likelihood function is rather inconvenient for Bayesian quantile regression. Specifically, the conditional distribution for the regression coefficients is not analytically tractable due to the complexity of the above likelihood function. By Chu et al. [31] and Lum et al. [32], the e_i can be denoted as a location-scale mixture of normals as follows:

$$e_i = \frac{1-2\tau}{\tau(1-\tau)} z_i + u_i \sqrt{\frac{2}{\tau(1-\tau)} \sigma z_i}, \tag{2.6}$$

where, $z_i \sim Exp(\sigma^{-1})$, $u_i \sim N(0,1)$, and z_i and u_i are independent. Let $v_i = \frac{1}{\tau(1-\tau)} z_i$, then

$$(e_i| v_i, \sigma) \sim N((1-2\tau)v_i, 2\sigma v_i), \ (v_i|\sigma) \sim Exp(\sigma^{-1}\tau(1-\tau)). \tag{2.7}$$

Then, we have

$$(Y_\tau|\beta, \sigma, \Sigma, X) \sim N_n(X\beta + (1-2\tau)V, 2\sigma\Sigma), \tag{2.8}$$

where, $Y = \{y_1, y_2, \cdots, y_n\}^T$ is an $n \times 1$ response vector of y_i, X is an $n \times k$ predictor matrix with i-th row $\boldsymbol{x}_i^T$, V denotes an $n \times 1$ vector of v_i, and Σ is the diagonal matrix of v_i.

Subsequently, let $Y_\tau^* = \frac{1}{\sqrt{2}} (Y - (1-2\tau) V)$ and $X^* = \frac{1}{\sqrt{2}} X$, we have

$$(Y_\tau^* \big| X^*, \beta, \sigma, \Sigma) \sim N_n (X^*\beta, \sigma\Sigma),$$

and

$$f(Y_\tau^*|X^*, \beta, \sigma, \Sigma) = (2\pi\sigma)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} exp\{-\frac{1}{2\sigma}[Y_\tau^* - X^*\beta]^T \Sigma^{-1} [Y_\tau^* - X^*\beta]\}. \tag{2.9}$$

2.2. Likelihood function and prior distribution

According to Chu et al. [31], a NIG distribution is given by the following definition.

Definition 1. *Let β be k-dimensional vector and $\sigma > 0$, if the joint distribution of (β, σ) has the following form*

$$f(\beta, \sigma) = C\sigma^{-\left(a+\frac{k}{2}+1\right)} \exp\{-\frac{1}{\sigma}[b + \frac{1}{2}(\beta - \mu)^T \Lambda (\beta - \mu)]\}, \tag{2.10}$$

where C is a proportionality constant, then we have that (β, σ) follows the k-dimensional distribution of $NIG_k(\mu, \Lambda, a, b)$, that is, $f(\sigma) \sim IG(a, b)$, $f(\beta|\sigma) \sim N_k\left(\mu, \sigma\Lambda^{-1}\right)$.

Let $\hat{\beta}_\tau = (X^{*T}\Sigma^{-1}X^*)^{-1} X^*\Sigma^{-1} Y_\tau^*$, the likelihood (2.8) can be reformulated as:

$$\begin{aligned} f(Y_\tau^*|X^*, \beta, \sigma, V, \Sigma) &\propto \sigma^{-\frac{n-k}{2}} exp\{-\frac{1}{2\sigma}[Y_\tau^* - X^*\hat{\beta}_\tau]^T \Sigma^{-1} [Y_\tau^* - X^*\hat{\beta}_\tau]\} \\ &\times \sigma^{-\frac{k}{2}} exp\{-\frac{1}{2\sigma}(\beta - \hat{\beta}_\tau)^T \Lambda (\beta - \hat{\beta}_\tau)\} \\ &= (\sigma)^{-(a+\frac{k}{2}+1)} exp\{-\frac{1}{\sigma}[b_\tau + \frac{1}{2}(\beta - \mu_\tau)^T \Lambda (\beta - \mu_\tau)]\}, \end{aligned} \tag{2.11}$$

where $\mu_\tau = \hat{\beta}_\tau$, $\Lambda = X^{*T}\Sigma^{-1}X^*$, $a = \frac{n-k-2}{2}$ and $b_\tau = \frac{1}{2}[Y^* - X^*\hat{\beta}_\tau]^T \Sigma^{-1} [Y^* - X^*\hat{\beta}_\tau]$.

According to Definition 1, the likelihood function (2.8) can be represented as the $NIG(\mu_\tau, \Lambda, a, b_\tau)$ distribution,

$$f(Y_\tau^* \big| X^*, \beta, \sigma, \Sigma) \propto NIG(\mu_\tau, \Lambda, a, b_\tau). \tag{2.12}$$

Now, for the sake of computational convenience, a conjugate NIG prior distribution is needed to construct our Bayesian model of streaming data. According to the study of Chu et al. [31], a conjugate prior for (β, σ) with a modification of Zellner's informative g-prior in quantile regression could be expressed as:

$$(\beta|\sigma, X^*, \Sigma) \propto N_k(0_k, g\sigma(X^{*T}\Sigma^{-1}X^*)^{-1}), \tag{2.13}$$

$$(\sigma) \propto \sigma^{-1}. \tag{2.14}$$

Based on the above prior distributions of β and σ, a joint g-NIG prior distribution can be obtained by

$$\pi(\beta,\sigma|X^*,\Sigma)\propto\sigma^{-(\frac{k}{2}+1)}\exp\{-\frac{1}{\sigma}[\frac{1}{2}\beta^T\frac{X^{*T}\Sigma^{-1}X^*}{g}\beta]\}\propto NIG_k(\mu_0,\Lambda_{g_0},a_0,b_0),\tag{2.15}$$

where $\mu_0=0_k$, $\Lambda_{g_0}=\frac{X^{*T}\Sigma^{-1}X^*}{g}$, $a_0=0$, $b_0=0$, $g>0$ is a known parameter and can be set freely.

By (2.11) and (2.14), we have the following hierarchical model.

$$y\,|\,\beta,\sigma\sim NIG(\mu_\tau,\Lambda,a,b_\tau),\tag{2.16}$$

$$\beta\,|\,\sigma,X^*,\Sigma\sim N_k(0_k,g\sigma(X^{*T}\Sigma^{-1}X^*)^{-1}),\tag{2.17}$$

$$v_i\,|\,\sigma\sim Exp(\sigma^{-1}\tau(1-\tau)),\tag{2.18}$$

$$\sigma\sim\sigma^{-1}.\tag{2.19}$$

2.3. The full conditional distributions and the Gibbs sampler

Given the informative g-prior distribution, the model becomes computationally easier to solve since the full conditional distributions now are

$$\begin{aligned}\pi(\beta,\sigma,V|Y_\tau^*,X^*)&\propto f(Y_\tau^*\big|X^*,\beta,\sigma,V)f(\beta|X^*,\sigma,V)f(V|\sigma)f(\sigma)\\&\propto\sigma^{-(\frac{3n+k+2}{2})}(\prod_{i=1}^n v_i^{-\frac{1}{2}})|X^{*T}\Sigma^{-1}X^*|^{\frac{1}{2}}\exp\{-\frac{1}{2\sigma}[(Y_\tau^*-X^*\beta)^T\Sigma^{-1}(Y_\tau^*-X^*\beta)\\&\quad+\beta^T\frac{X^{*T}\Sigma^{-1}X^*}{g}\beta+2\tau(1-\tau)\sum_{i=1}^n v_i]\}.\end{aligned}\tag{2.20}$$

(1) Sample v_i from

$$\pi(v_i|\beta,\sigma,Y_i,X_i)\propto GIG(0,\bar{\xi}_i,\bar{\zeta}_i),\tag{2.21}$$

where, $\bar{\xi}_i=\frac{(y_i-x_i^T\beta)^2+(\beta^Tx_ix_i^T\beta)/g}{2\sigma}$, $\bar{\zeta}_i=\frac{1}{2\sigma}$, $i=1,2,\cdots,n$.

(2) Sample β from

$$\begin{aligned}\pi(\beta|Y_\tau^*,X^*,\sigma,V)&\propto exp\{-\frac{1}{2\sigma}[(Y_\tau^*-X^*\beta)^T\Sigma^{-1}(Y_\tau^*-X^*\beta)+\beta^T\frac{X^{*T}\Sigma^{-1}X^*}{g}\beta]\}\\&\propto N_k(\bar{\mu}_\tau,\sigma\bar{\Lambda}^{-1}),\end{aligned}\tag{2.22}$$

where, $\bar{\mu}_\tau=[(1+\frac{1}{g})X^{*T}\Sigma^{-1}X^*]^{-1}X^{*T}\Sigma^{-1}Y_\tau^*$, $\bar{\Lambda}=(1+\frac{1}{g})X^{*T}\Sigma^{-1}X^*$.

(3) Sample σ from

$$\begin{aligned}\pi(\sigma|Y_\tau^*,X^*,\beta,V)&\propto\sigma^{\frac{3n+k}{2}+1}exp\{-\frac{1}{2\sigma}[(Y_\tau-X\beta)^T\Sigma^{-1}(Y_\tau-X\beta)+\beta^T\frac{X^{*T}\Sigma^{-1}X^*}{g}\beta]+2\tau(1-\tau)\sum_{i=1}^n v_i\}\\&\propto IG(a,b),\end{aligned}\tag{2.23}$$

where, $a=\frac{3n+k}{2}$, $b=\frac{1}{2}[(Y_\tau^*-X^*\beta)^T\Sigma^{-1}(Y_\tau^*-X^*\beta)+\beta^T\frac{X^{*T}\Sigma^{-1}X^*}{g}\beta+2\tau(1-\tau)\sum\limits_{i=1}^n v_i]$.

Here, $GIG(0, \bar{\xi}_i, \bar{\zeta}_i)$ is the generalized inverse Gaussian distribution with probability density function $x^{-1} exp\{\frac{1}{2}(\bar{\zeta}x^{-1} + \bar{\xi}x)\}$ for $x > 0$.

For the convenience of calculation and expression, based on the posterior distribution of β and σ, we have a joint posterior distribution of $\pi(\beta, \sigma | Y_\tau^*, X^*, V)$ as follows:

$$\pi(\beta, \sigma | Y_\tau^*, X^*, V) \propto NIG(\bar{\mu}_\tau, \bar{\Lambda}, \bar{a}, \bar{b}_\tau), \tag{2.24}$$

where, $\bar{\mu}_\tau = [(1 + \frac{1}{g})X^{*T}\Sigma^{-1}X^*]^{-1} X^{*T}\Sigma^{-1}Y_\tau^*$, $\bar{\Lambda} = (1 + \frac{1}{g})X^{*T}\Sigma^{-1}X^*$, $\bar{a} = \frac{3}{2}n$, $\bar{b}_\tau = \frac{1}{2}Y_\tau^{*T}\Sigma^{-1}Y_\tau^* - \frac{1}{2}\bar{\mu}_\tau^T\bar{\Lambda}\bar{\mu}_\tau + \tau(1-\tau)\sum_{i=1}^{n} v_i$.

3. Bayesian quantile regression for streaming data

Different from traditional regression datasets, our focus in this paper is on streaming data sets. Assume $\{D_m, m = 1, \cdots, M, \cdots\}$ are aggregated streaming data, $D_m = \{(y_{mi}, \boldsymbol{x}_{mi}), i = 1, \cdots, n_m\}$ are the streaming data of m-th batch with sample size n_m, and the total sample size up to M is $N_M = \sum_{i=1}^{M} n_i$. To construct the posterior distribution for streaming data, the NIG multiplier operator defined in Chu et al. [31] will be used, as presented in the following proposition.

Proposition 1. *The multiplication of H k-dimensional distributions $NIG_k(\mu_h, \Lambda_h, a_h, b_h)$ is also a NIG distribution, that is*

$$NIG_k(\mu, \Lambda, a, b) = \prod_{h=1}^{H} NIG_k(\mu_h, \Lambda_h, a_h, b_h), \tag{3.1}$$

where $\mu = (\sum_{h=1}^{H} \Lambda_h)^{-1}(\sum_{h=1}^{H} \Lambda_h\mu_h)$, $\Lambda = \sum_{h=1}^{H} \Lambda_h$, $a = \sum_{h=1}^{H} a_h + \frac{(H-1)(k+2)}{2}$, $b = \sum_{h=1}^{H} b_h + \frac{1}{2}\sum_{h=1}^{H}(\mu_h - \mu)^T\Lambda_h(\mu_n - \mu)$.

We begin with the m-th batch of streaming data D_m. Let $Y_{\tau,m}^*$ and X_m^* be the data in m-th batch D_m, Σ_m be an $n_m \times n_m$ diagonal matrix, $\hat{\beta}_{\tau,m} = \left(X_m^{*T}\Sigma_m^{-1}X_m^{*T}\right)^{-1} X_m^*\Sigma_m^{-1}Y_{\tau,m}^*$. From (2.10), the likelihood function can be derived for the m-th batch of streaming data as follows:

$$\begin{aligned} f(Y_{\tau,m}^*|X_m^*, \beta, \sigma, \Sigma) &\propto \sigma^{-\frac{n-k}{2}} exp\{-\frac{1}{2\sigma}[Y_{\tau,m}^* - X_m^*\hat{\beta}_{\tau,m}]^T\Sigma^{-1}[Y_{\tau,m}^* - X_m^*\hat{\beta}_{\tau,m}]\} \\ &\quad \times \sigma^{-\frac{k}{2}} exp\{-\frac{1}{2\sigma}(\beta - \hat{\beta}_{\tau,m})^T\Lambda_m(\beta - \hat{\beta}_{\tau,m})\} \\ &= (\sigma)^{-(a_m + \frac{k}{2} + 1)} exp\{-\frac{1}{\sigma}[b_{\tau,m} + \frac{1}{2}(\beta - \mu_{\tau,m})^T\Lambda_m(\beta - \mu_{\tau,m})]\}, \end{aligned} \tag{3.2}$$

where, $\mu_{\tau,m} = \hat{\beta}_{\tau,m} = (X_m^{*T}\Sigma_m^{-1}X_m^*)^{-1} X_m^*\Sigma_m^{-1}Y_{\tau,m}^*$, $\Lambda_m = X_m^{*T}\Sigma_m^{-1}X_m^*$, $a_m = \frac{n_m - k - 2}{2}$, $b_{\tau,m} = \frac{1}{2}[Y_{\tau,m}^* - X_m^{*T}\hat{\beta}_{\tau,m}]^T\Sigma_m^{-1}[Y_{\tau,m}^* - X_m^{*T}\hat{\beta}_{\tau,m}]$.

Subsequently, the posterior distribution of the streaming data can be obtained by multiplying the NIG prior of the stream data with the likelihood function of the m-th batch data. To implement our BSQR method, the prior distribution of the m-th batch of streaming data is introduced in the following definition.

Definition 2. *The prior distribution of the m-th batch is defined by* $NIG(\mu_{p,\tau,m}, \Lambda_{p,m}, a_{p,m}, b_{p,\tau,m})$*, which consists of two parts, the adjusted posterior distribution of historical data and the g-NIG prior of current data, that is*

$$NIG(\mu_{p,\tau,m}, \Lambda_{p,m}, a_{p,m}, b_{p,\tau,m}) = NIG_k^{(m-1)}(\bar{\mu}_{\tau,m-1}, \bar{\Lambda}_{m-1}, \bar{a}_{m-1} - \frac{k+2}{2}, \bar{b}_{\tau,m-1}) \times NIG(\mu_0, \Lambda_{m,0}, a_0, b_0),$$

where, $NIG_k^{(m-1)}$ *denotes the posterior distribution of the* $(m-1)$*-th batch for streaming data,*

$$\mu_{p,\tau,m} = (\bar{\Lambda}_{m-1} + \Lambda_{m,0})^{-1}(\bar{\Lambda}_{m-1}\bar{\mu}_{\tau,m-1} + \Lambda_{m,0}\mu_0),$$

$$\Lambda_{p,m} = \bar{\Lambda}_{m-1} + \Lambda_{m,0},$$

$$a_{p,m} = \bar{a}_{m-1} + a_0,$$

$$b_{p,\tau,m} = \bar{b}_{\tau,m-1} + b_0 + \frac{1}{2}(\bar{\mu}_{\tau,m-1} - \mu_{p,\tau,m})^T \Lambda_{m-1}(\bar{\mu}_{\tau,m-1} - \mu_{p,\tau,m}) + \frac{1}{2}\mu_{p,\tau,m}^T \Lambda_{m,0}\mu_{p,\tau,m}.$$

Remark 1. *For streaming data* $D_1, D_2, \cdots D_m, \cdots$*, if* $n_1 \geq \frac{k+2}{3}$*, where* n_1 *is the sample size of the first batch of streaming data, the prior distribution* $NIG(\mu_{p,\tau,m}, \Lambda_{p,m}, a_{p,m}, b_{p,\tau,m})$ *exists.*

Through the prior distribution of stream data $NIG(\mu_{p,\tau,m}, \Lambda_{p,m}, a_{p,m}, b_{p,\tau,m})$, we can get the posterior distribution of the streaming data. Theorem 1 describes the renewability of the posterior distribution of streaming data obtained through the NIG multiplier operator.

Theorem 1. *Consider a linear QR model with streaming data observations* $D_1, D_2, \cdots D_m, \cdots$*. Setting the prior distribution of the initial streaming data as the g-NIG prior* $NIG_k\left(\mu_0, \Lambda_{g_0}, a_0, b_0\right)$*. When the m-th batch of streaming data is received, the posterior distribution of the streaming data* $NIG_k^{(m)}(\bar{\mu}_{\tau,m}, \bar{\Lambda}_m, \bar{a}_m, \bar{b}_{\tau,m})$ *can be expressed as*

$$NIG_k^{(m)}(\bar{\mu}_{\tau,m}, \bar{\Lambda}_m, \bar{a}_m, \bar{b}_{\tau,m}) = NIG(\mu_{p,\tau,m}, \Lambda_{p,m}, a_{p,m}, b_{p,\tau,m}) \times NIG_k^{(L)}(\mu_{\tau,m}, \Lambda_m, a_m, b_{\tau,m}),$$

where, $NIG_k^{(L)}(\mu_{\tau,m}, \Lambda_m, a_m, b_{\tau,m})$ *is the likelihood function for the m-th block of data nodes,*

$$\begin{aligned}\bar{\mu}_{\tau,m} =&(\bar{\Lambda}_{m-1} + \Lambda_{m,0} + \Lambda_m)^{-1}(\mu_0\Lambda_{m,0} + \bar{\mu}_{\tau,m-1}\bar{\Lambda}_{m-1} + \mu_{\tau,m}\Lambda_m) \\ =& ((1 + \frac{1}{g})\sum_{i=1}^{m} X_i^{*T}\Sigma_i^{-1}X_i^*)^{-1}(\sum_{i=1}^{m} X_i^{*T}\Sigma_i^{-1}Y_i^*),\end{aligned}$$

$\bar{\Lambda}_m = \bar{\Lambda}_{m-1} + \Lambda_{m,0} + \Lambda_m = (1 + \frac{1}{g})\sum_{i=1}^m X_i^{*T}\Sigma_i^{-1}X_i^*,$

$\bar{a}_m = \bar{a}_{m-1} + a_0 + a_m + \frac{k+2}{2} = \frac{3}{2}n_m + a_0,$

$\bar{b}_{\tau,m} = \bar{b}_{\tau,m-1} + b_0 + b_{\tau,m} + \frac{1}{2}(\bar{\mu}_{\tau,m-1} - \bar{\mu}_{\tau,m})^T\bar{\Lambda}_{m-1}(\bar{\mu}_{\tau,m-1} - \bar{\mu}_{\tau,m}) + \frac{1}{2}\bar{\mu}_{\tau,m}^T\Lambda_{m,0}\bar{\mu}_{\tau,m} + \frac{1}{2}(\mu_{\tau,m} - \bar{\mu}_{\tau,m})^T\Lambda_m(\mu_{\tau,m} - \bar{\mu}_{\tau,m}) = \frac{1}{2}\sum_{i=1}^m Y_{\tau,i}^{*T}\Sigma_i Y_{\tau,i}^* - \frac{1}{2}\bar{\mu}_{\tau,m}^T\bar{\Lambda}_m\bar{\mu}_{\tau,m} + \tau(1-\tau)\sum_{i=1}^{n} v_i + b_0.$

According to Theorem 1, the update of parameters in the posterior distribution of the streaming data only requires the parameters of the posterior distribution of historical data. Specifically, after receiving the streaming data, we only need to calculate and store the parameters of the posterior distribution $\bar{\mu}_{\tau,m}$,

$\bar{\Lambda}_m$, $\bar{a}_m$ and $\bar{b}_{\tau,m}$. In addition, when given all the values of v_i, we can see from Theorem 1 that the posterior distribution parameters of streaming data are equal to the posterior distribution parameters of the entire data. The proof of Theorem 1 is provided in the appendix.

Under Theorem 1 and Proposition 1, an effective algorithm for the Bayesian streaming quantile regression (BSQR) is developed in Algorithm 1.

Algorithm 1: Bayesian streaming data quantile regression algorithm

Step 1. Receive streaming data for the m-th batch and calculate the NIG likelihood function

$$NIG_k(\mu_{\tau,m}, \Lambda_m, a_m, b_{\tau,m}),$$

where, $\mu_{\tau,m} = \hat{\beta}_{\tau,m} = (X_m^{*T}\Sigma_m^{-1}X_m^*)^{-1}X_m^*\Sigma_m{}^{-1}Y_{\tau,m}^*$, $\Lambda_m = X_m^{*T}\Sigma_m^{-1}X_m^*$, $a_m = \frac{n-k-2}{2}$, $b_{\tau,m} = \frac{1}{2}[Y_{\tau,m}^* - X_m^{*T}\hat{\beta}_{\tau,m}]^T\Sigma_m^{-1}[Y_{\tau,m}^* - X_m^{*T}\hat{\beta}_{\tau,m}]$.

Step 2. Calculate the prior distribution of the m-th batch of streaming data:

$$NIG(\mu_{p,\tau,m}, \Lambda_{p,m}, a_{p,m}, b_{p,\tau,m}) = NIG(\bar{\mu}_{\tau,m-1}, \bar{\Lambda}_{m-1}, \bar{a}_{m-1} - \frac{k+2}{2}, \bar{b}_{\tau,m-1}) \times NIG(\mu_0, \Lambda_{m,0}, 0, 0).$$

Step 3. Calculate the posterior distribution of the m-th batch of streaming data:

$$NIG_k(\bar{\mu}_{\tau,m}, \bar{\Lambda}_m, \bar{a}_m, \bar{b}_{\tau,m}) = NIG(\mu_{p,\tau,m}, \Lambda_{p,m}, a_{p,m}, b_{p,\tau,m}) \times NIG_k^{(L)}(\mu_{\tau,m}, \Lambda_m, a_m, b_{\tau,m}).$$

Step 4. Gibbs sampling:
(1) Denote j as the iteration count. Then set $j = 0$ and establish a set of initial values: $v^{(j)}$, $\sigma^{(j)}$, $\beta^{(j)}$.
(2) Follow the posterior distributions of β_m, σ_m and v_m.
(i) Sample $\boldsymbol{v}_m^{(j+1)}$ from $f\left(\boldsymbol{v}_m \mid \boldsymbol{\beta}_m^{(j)}, \sigma_m^{(j)}\right)$.
(ii) Sample $\sigma_m^{(j+1)}$ from $f\left(\sigma_m \mid \boldsymbol{\beta}_m^{(j)}, \boldsymbol{v}_m^{(j+1)}\right)$.
(iii) Sample $\boldsymbol{\beta}_m^{(j+1)}$ from $f\left(\boldsymbol{\beta}_m \mid \sigma_m^{(j+1)}, \boldsymbol{v}_m^{(j+1)}\right)$.
(3) Set $j = j + 1$ and return to (2) until $j = I$, where I is the number of iteration times.

4. Simulation study

In this section, the proposed BSQR will be illustrated by simulated data.

We consider datasets in different cases and generate the streaming datasets $D_1, D_2, \cdots, D_M$ up to batch M, where $D_m = \{(y_{mi}, \boldsymbol{x}_{mi}) : 1 \leq i \leq n_m\}$.

$$y_{mi} = \boldsymbol{x}_{mi}^T\boldsymbol{\beta_0} + e_{mi}, i = 1, \cdots, n_m, \tag{4.1}$$

where, the elements of $\boldsymbol{x}_{mi}$ are independently and identically distributed as the standard normal distribution, and the vector $\boldsymbol{\beta}_0 = (1, 3, 2, 10, 4, 3, -1, 4, 5, 0)^T$. For the random error e_{mi}, in order to illustrate the robustness, the following three cases are considered.

Case 1. e_{mi} follows the asymmetric Laplace distribution.
Case 2. e_{mi} follows the standard normal distribution.
Case 3. e_{mi} follows $t(3)$ distribution.

For a comprehensive illustration, we compare our method with four competing estimators in the current literature.

(1) The oracle Bayesian quantile regression (BQR) obtained by processing the entire data once.

(2) The renewable quantile regression (RQR) proposed in Wang et al. [30].

(3) The renewable SQR (RSQR) estimator proposed in Jiang et al. [37].

(4) The online linear estimator for the QR (OLEQR) proposed in Chen et al. [28].

The first competing estimator BQR is used to show that our method is statistically equivalent to the oracle one and verify the theoretical results of Theorem 1 for the situation of finite sample. The second to fourth estimator RQR is used to compare our method with the method of streaming data in the frequentist framework, which makes our simulation experiments more comprehensive.

For the selection of hyperparameter g in g-NIG prior, Smith and Kohn [33] found that choosing g between 10 and 1000 can achieve good results. Meanwhile, they set $g = 100$ as the final value. Dao and Wang [34] also used g-NIG priors and found that their numerical results for different choices of g within the range of $100 \leq g \leq 1000$ yielded very similar results. For our model, we also validated the selection of g using simulation experiments, and the results are shown in Figure 1. It can be observed that the choice of g has little impact on MSE. Therefore, we follow the suggestions of Smith and Kohn and set $g = 100$, a choice adopted by various researchers, such as Lee et al. [35] and Chen et al. [36].

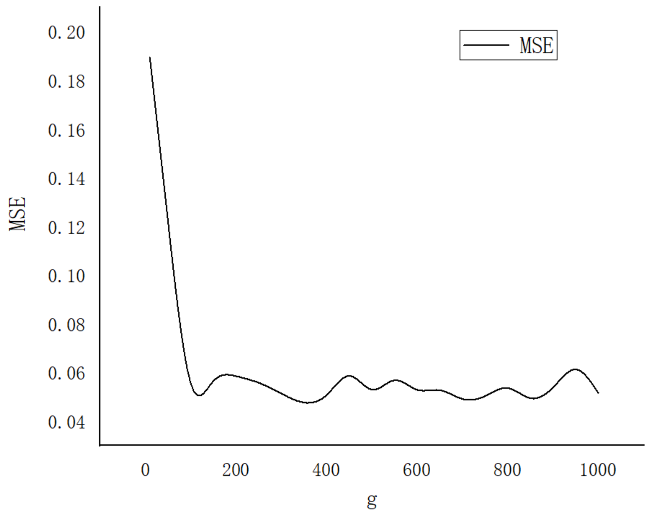

Figure 1. The choice of the hyperparameter g.

To examinate the perfomance of the BSQR method, we fix n with varying batch size n_m. Specifically, M streaming datasets with $n = 10^4$ observations are generated independently, where batch number M is between 50 and 100, with each batch having $\frac{10000}{M}$ observations. We repeat the experiment 500 times and define $\hat{\beta}_{j(s)}$ as the estimator of β_j in the s-th replication. In order to compare estimation accuracy, the mean square errors (MSE) $MSE(\hat{\beta}_j) = \frac{1}{500}\sum_{s=1}^{500}(\hat{\boldsymbol{\beta}}_{j(s)} - \boldsymbol{\beta}_{0j})^2$, $j = 1, 2, \cdots, 10$, are calculated for the three methods. The results of case 1 are shown in Tables 1 and 2.

Table 1. Comparison between Bayesian streaming data method and other methods in the case of ALD error distribution and m=50 batches.

τ	Method	MSE	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}
$\tau = 0.1$	BQR	0.109	1.98E-03	6.64E-03	5.78E-03	5.41E-02	8.93E-03	5.45E-03	2.12E-04	1.09E-03	1.04E-02	2.98E-03
	BSQR	0.107	1.47E-03	7.73E-03	3.13E-03	5.08E-02	7.75E-03	9.96E-03	2.82E-03	9.23E-03	1.29E-02	1.88E-03
	RQR	0.117	6.12E-03	1.33E-02	5.12E-03	8.30E-03	4.78E-03	9.79E-03	1.54E-03	2.22E-02	5.98E-03	4.43E-02
	RSQR	0.117	4.67E-03	4.25E-03	6.44E-03	9.21E-02	9.71E-03	9.79E-03	5.92E-03	6.59E-03	7.74E-03	9.41E-02
	OLEQR	0.121	5.05E-03	5.30E-02	5.77E-03	5.06E-03	8.42E-03	7.34E-03	3.77E-03	6.18E-02	8.04E-03	7.47E-02
$\tau = 0.3$	BQR	0.063	8.50E-04	3.61E-03	2.06E-03	2.85E-02	5.75E-03	5.81E-03	4.41E-03	6.05E-03	8.01E-03	1.96E-03
	BSQR	0.066	1.54E-03	3.50E-03	1.35E-03	3.14E-02	5.22E-03	6.49E-03	1.36E-03	6.07E-03	8.50E-03	1.15E-03
	RQR	0.068	9.59E-03	2.29E-03	7.05E-03	7.57E-03	5.79E-03	1.62E-03	1.76E-03	6.46E-03	2.11E-02	6.61E-03
	RSQR	0.067	3.67E-03	3.25E-02	5.44E-03	9.21E-03	9.74E-03	4.92E-03	6.59E-03	7.77E-02	7.31E-03	9.41E-03
	OLEQR	0.075	4.76E-03	2.33E-03	5.28E-03	6.32E-03	4.54E-03	7.79E-03	1.14E-03	2.32E-03	1.98E-03	5.43E-03
$\tau = 0.5$	BQR	0.057	3.33E-03	1.00E-03	8.50E-05	2.23E-02	9.09E-03	5.58E-04	4.93E-03	5.09E-03	1.04E-02	8.90E-04
	BSQR	0.057	1.11E-03	3.43E-03	1.83E-03	2.84E-02	5.32E-03	3.70E-03	8.20E-04	5.32E-03	6.97E-03	9.79E-04
	RQR	0.060	1.47E-03	6.15E-03	4.15E-03	1.46E-02	8.96E-03	5.96E-03	4.39E-03	3.44E-03	8.48E-02	7.64E-03
	RSQR	0.060	5.05E-03	5.30E-02	5.77E-03	5.06E-03	8.42E-03	7.43E-03	3.77E-03	1.18E-02	8.04E-03	7.41E-03
	OLEQR	0.071	1.12E-03	6.53E-03	2.12E-03	6.12E-03	1.78E-03	2.59E-03	2.46E-03	1.26E-03	9.98E-03	2.43E-03
$\tau = 0.7$	BQR	0.064	1.47E-03	4.19E-03	1.69E-03	3.23E-03	5.06E-03	3.53E-03	1.60E-03	5.74E-03	7.29E-03	1.60E-03
	BSQR	0.064	1.80E-03	4.10E-03	2.68E-03	3.20E-02	4.11E-03	3.83E-03	1.27E-03	7.01E-03	9.22E-03	1.36E-03
	RQR	0.068	5.08E-03	2.59E-03	5.11E-03	2.26E-03	3.55E-03	1.38E-02	2.26E-03	7.64E-03	3.34E-02	1.01E-02
	RSQR	0.068	2.89E-03	2.89E-03	1.12E-03	5.30E-03	3.78E-03	5.79E-03	3.25E-03	5.02E-03	5.98E-03	2.63E-02
	OLEQR	0.076	1.48E-03	9.74E-02	4.28E-03	2.35E-03	2.63E-03	1.44E-03	1.84E-03	2.44E-02	1.84E-03	1.99E-02
$\tau = 0.9$	BQR	0.111	3.52E-02	6.70E-03	4.40E-03	5.24E-02	1.06E-02	5.68E-03	4.16E-03	8.57E-03	1.17E-02	3.46E-03
	BSQR	0.111	4.06E-03	6.60E-03	3.85E-03	5.04E-02	1.12E-02	6.47E-03	3.82E-03	1.24E-02	1.58E-02	3.55E-03
	RQR	0.119	1.60E-02	2.08E-03	2.58E-03	1.52E-02	1.97E-03	1.25E-02	4.43E-03	1.07E-02	1.19E-02	8.86E-03
	RSQR	0.117	7.64E-03	1.67E-02	8.70E-03	2.12E-02	7.01E-03	3.83E-03	5.22E-03	2.45E-02	7.58E-03	1.11E-02
	OLEQR	0.124	1.53E-02	1.65E-02	1.66E-03	2.48E-03	5.92E-03	1.04E-03	7.04E-03	3.88E-02	1.28E-02	8.55E-03

Table 2. Comparison between Bayesian streaming data method and other methods in the case of ALD error distribution and m=100 batches.

τ	Method	MSE	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}
$\tau = 0.1$	BQR	0.109	1.98E-03	6.64E-03	5.78E-03	5.41E-02	8.93E-03	5.45E-03	2.12E-04	1.09E-03	1.04E-02	2.98E-03
	BSQR	0.110	3.35E-03	5.54E-03	2.29E-03	5.08E-02	9.47E-03	9.36E-02	3.19E-02	9.40E-03	1.96E-02	1.16E-03
	RQR	0.119	2.04E-02	1.12E-02	4.16E-03	9.22E-03	8.87E-03	1.44E-02	1.08E-03	2.23E-02	9.69E-03	7.12E-03
	RSQR	0.117	1.82E-02	1.01E-02	4.12E-03	8.30E-03	6.78E-03	9.79E-03	1.28E-03	2.20E-02	5.98E-03	7.43E-03
	OLEQR	0.121	2.56E-02	1.33E-02	5.10E-03	9.30E-03	8.78E-03	1.79E-02	1.13E-03	9.22E-03	4.98E-03	1.43E-02
$\tau = 0.3$	BQR	0.063	2.01E-04	1.32E-02	4.80E-03	2.35E-02	6.33E-03	1.26E-03	8.03E-05	4.59E-03	1.39E-03	1.55E-03
	BSQR	0.066	1.48E-03	3.99E-03	2.30E-03	3.10E-02	5.49E-03	4.37E-03	1.44E-03	6.09E-03	8.88E-03	1.52E-03
	RQR	0.066	3.19E-03	6.94E-03	4.59E-03	5.50E-02	5.56E-03	2.20E-03	6.33E-03	1.43E-02	2.10E-02	1.80E-02
	RSQR	0.066	3.12E-03	5.94E-02	5.59E-03	8.30E-02	2.78E-03	2.79E-03	6.28E-03	1.22E-02	2.30E-03	1.79E-02
	OLEQR	0.079	3.21E-03	5.33E-02	5.98E-03	9.12E-03	4.78E-03	5.79E-03	6.54E-03	4.25E-02	2.98E-03	2.12E-02
$\tau = 0.5$	BQR	0.057	3.33E-03	1.00E-03	8.50E-05	2.23E-02	9.09E-03	5.58E-04	4.93E-03	5.09E-03	1.04E-02	8.90E-04
	BSQR	0.057	1.58E-03	2.49E-03	1.53E-03	2.75E-02	3.90E-03	5.51E-03	9.75E-04	5.32E-03	7.89E-03	1.05E-04
	RQR	0.060	4.76E-03	7.50E-04	6.99E-03	3.88E-03	7.35E-03	7.98E-03	1.18E-02	5.27E-03	6.75E-03	5.35E-03
	RSQR	0.060	4.12E-03	6.43E-03	6.12E-03	2.30E-03	6.28E-03	7.79E-03	9.54E-03	1.01E-02	5.98E-03	6.03E-03
	OLEQR	0.065	1.32E-03	5.33E-03	5.12E-03	2.30E-03	1.38E-03	2.78E-03	3.54E-03	6.12E-03	2.28E-03	6.23E-02
$\tau = 0.7$	BQR	0.064	1.47E-03	4.19E-03	1.69E-03	3.23E-03	5.06E-03	3.53E-03	1.60E-03	5.74E-03	7.29E-03	1.60E-03
	BSQR	0.065	1.72E-03	5.21E-03	2.42E-03	3.01E-02	6.50E-03	3.56E-03	1.50E-03	6.14E-03	7.30E-03	1.13E-03
	RQR	0.070	1.32E-03	1.01E-03	5.14E-03	1.16E-02	2.50E-03	9.91E-03	1.49E-03	4.27E-03	1.13E-02	4.60E-03
	RSQR	0.068	1.56E-03	6.33E-03	5.65E-03	5.30E-03	5.38E-03	8.79E-03	2.56E-03	1.02E-02	2.98E-03	1.56E-02
	OLEQR	0.075	1.98E-03	1.33E-02	5.89E-03	6.89E-03	6.78E-03	8.29E-03	2.54E-03	9.56E-03	3.98E-03	8.98E-02
$\tau = 0.9$	BQR	0.111	3.52E-02	6.70E-03	4.40E-03	5.24E-02	1.06E-02	5.68E-03	4.16E-03	8.57E-03	1.17E-02	3.46E-03
	BSQR	0.115	3.31E-03	7.17E-03	5.11E-03	5.20E-03	9.83E-03	6.41E-03	3.89E-03	1.02E-03	1.41E-03	1.02E-03
	RQR	0.118	2.21E-02	1.18E-02	7.33E-03	1.15E-02	5.19E-03	1.41E-03	1.65E-03	3.07E-03	5.07E-02	9.56E-03
	RSQR	0.117	1.12E-02	1.01E-02	9.89E-03	5.63E-03	4.78E-03	8.79E-03	4.54E-03	1.22E-02	5.98E-03	2.01E-02
	OLEQR	0.126	1.49E-02	1.50E-02	1.19E-02	9.27E-03	5.70E-03	5.47E-03	1.32E-02	1.60E-02	1.49E-03	2.30E-02

In practice, the error term e_{mi} may not be distributed as AL distribution. Therefore, Cases 2 and 3 are also simulated when the error term follows t-distribution with a degree of freedom 3, and standard normal distribution, respectively. The results are shown in Tables 3–6.

From the simulation results, we have several findings. First, our streaming data Bayesian quantile regression method can always estimate the parameters consistently under three error distributions. In the meantime, the estimation results are not affected by the data batch size, since the MSE values of our method are almost in the same scale as the batch number increases from 50 to 100. Second, under ALD error distribution, our method performs better than the other three methods, and also performs comparably with the oracle Bayesian quantile regression, as the BQR and BSQR methods are very close. Third, under other error distributions, our method performs worse than RQR and RSQR methods, but still performs better than the OLEQR method.

Table 3. Comparison between Bayesian streaming data method and other methods in the case of normal error distribution and m=50 batches.

τ	Method	MSE	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}
$\tau = 0.1$	BQR	0.032	1.99E-03	1.57E-03	3.33E-04	1.62E-02	7.34E-04	1.78E-04	9.46E-05	1.04E-02	2.28E-02	2.26E-03
	BSQR	0.034	1.12E-03	2.41E-03	1.49E-03	2.14E-02	4.44E-03	2.63E-03	1.03E-03	3.20E-03	5.46E-03	2.13E-03
	RQR	0.028	2.82E-03	2.72E-03	3.33E-03	3.15E-02	3.15E-03	2.77E-03	2.92E-03	2.28E-03	9.92E-04	7.63E-03
	RSQR	0.026	3.12E-03	2.33E-03	2.12E-03	8.30E-03	1.78E-03	1.79E-03	1.94E-03	2.02E-03	1.98E-03	2.43E-03
	OLEQR	0.051	4.12E-03	4.33E-03	5.12E-03	9.90E-03	4.78E-03	3.79E-03	1.54E-03	5.22E-03	5.98E-03	4.53E-03
$\tau = 0.3$	BQR	0.041	7.24E-04	1.79E-03	4.08E-04	1.50E-02	7.87E-03	7.75E-04	8.83E-04	3.34E-04	1.27E-02	1.38E-03
	BSQR	0.041	9.42E-04	2.35E-03	1.42E-03	2.03E-02	4.03E-03	2.17E-03	9.27E-04	3.57E-03	5.35E-03	7.81E-04
	RQR	0.021	2.02E-03	2.37E-03	1.47E-03	2.35E-02	1.96E-03	2.06E-03	2.39E-03	2.03E-03	2.41E-03	3.22E-03
	RSQR	0.019	9.11E-04	2.34E-03	1.31E-03	1.37E-03	2.18E-03	2.47E-03	3.05E-03	2.44E-03	1.78E-03	1.45E-03
	OLEQR	0.046	3.84E-03	1.47E-03	1.05E-03	2.86E-03	2.41E-03	1.69E-03	1.12E-03	3.78E-02	2.30E-03	3.34E-03
$\tau = 0.5$	BQR	0.040	9.16E-04	5.78E-03	2.52E-03	1.62E-02	3.50E-03	6.26E-04	7.71E-05	2.70E-03	8.09E-03	3.13E-04
	BSQR	0.041	6.67E-04	2.03E-03	1.14E-03	2.13E-02	4.26E-03	2.10E-03	7.64E-04	3.24E-03	5.32E-03	7.32E-04
	RQR	0.009	1.17E-03	8.77E-04	1.15E-03	7.73E-04	9.54E-04	7.48E-04	7.11E-04	7.47E-04	8.34E-04	7.74E-04
	RSQR	0.007	1.12E-03	1.33E-03	1.12E-03	1.30E-03	1.58E-03	9.79E-04	1.54E-03	2.22E-04	5.98E-04	5.84E-04
	OLEQR	0.042	2.07E-03	2.37E-03	1.34E-03	1.73E-03	1.41E-03	1.33E-03	1.58E-03	1.02E-03	1.10E-03	1.07E-03
$\tau = 0.7$	BQR	0.046	3.60E-04	1.38E-04	3.47E-03	1.98E-02	9.46E-03	1.82E-04	2.77E-03	1.75E-03	5.60E-03	2.91E-03
	BSQR	0.043	5.92E-04	2.00E-03	1.00E-03	2.11E-02	4.22E-03	1.98E-03	6.07E-04	3.28E-03	4.96E-03	4.10E-04
	RQR	0.015	1.94E-03	2.08E-03	1.38E-03	1.43E-02	1.09E-03	1.30E-03	1.40E-03	1.16E-03	1.26E-02	1.48E-03
	RSQR	0.015	1.01E-03	1.28E-02	6.94E-04	9.75E-04	3.07E-03	3.10E-03	1.38E-03	1.60E-02	2.34E-03	1.22E-02
	OLEQR	0.047	9.63E-04	2.13E-02	1.52E-03	1.20E-03	1.24E-03	1.26E-03	1.66E-03	1.31E-02	2.68E-03	1.48E-03
$\tau = 0.9$	BQR	0.039	1.27E-04	5.80E-04	2.06E-05	2.14E-02	5.26E-04	2.35E-03	9.00E-05	3.59E-03	9.84E-03	1.00E-03
	BSQR	0.040	5.27E-04	2.06E-03	9.60E-04	2.09E-02	4.23E-03	1.91E-03	5.45E-04	3.69E-03	4.93E-04	4.10E-04
	RQR	0.023	2.66E-03	3.17E-03	2.55E-03	2.45E-02	2.89E-03	2.60E-03	2.05E-03	3.62E-03	2.22E-02	2.86E-03
	RSQR	0.023	3.58E-03	2.15E-03	5.12E-03	8.30E-03	4.78E-03	9.09E-03	1.52E-03	1.98E-03	5.28E-03	2.01E-03
	OLEQR	0.048	3.05E-03	2.33E-03	9.12E-03	7.30E-03	3.78E-03	4.79E-03	1.88E-03	7.22E-02	5.98E-03	2.43E-03

Table 4. Comparison between Bayesian streaming data method and other methods in the case of normal error distribution and m=100 batches.

τ	Method	MSE	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}
$\tau = 0.1$	BQR	0.032	1.99E-03	1.57E-03	3.33E-04	1.62E-02	7.34E-04	1.78E-04	9.46E-05	1.04E-02	2.28E-02	2.26E-03
	BSQR	0.033	1.08E-03	3.00E-03	1.64E-03	2.02E-02	4.45E-03	2.51E-03	1.04E-03	3.60E-03	5.34E-03	9.94E-04
	RQR	0.020	1.50E-03	1.45E-03	2.97E-03	1.42E-03	2.12E-03	3.06E-03	6.78E-04	2.82E-03	4.99E-02	1.58E-03
	RSQR	0.020	2.72E-03	3.33E-03	8.25E-03	2.56E-03	3.35E-03	2.13E-03	1.42E-03	1.55E-03	1.47E-03	1.16E-03
	OLEQR	0.048	2.51E-03	2.47E-03	7.04E-03	6.03E-03	4.66E-03	8.57E-03	5.03E-03	6.50E-02	1.277E-03	1.06E-02
$\tau = 0.3$	BQR	0.041	7.24E-04	1.79E-03	4.08E-04	1.50E-02	7.87E-03	7.75E-04	8.83E-04	3.34E-04	1.27E-02	1.38E-03
	BSQR	0.043	8.46E-04	2.17E-03	1.28E-03	2.13E-03	4.26E-03	2.36E-03	1.03E-03	3.21E-03	5.32E-03	7.32E-04
	RQR	0.016	1.21E-03	1.71E-03	2.64E-03	1.19E-02	1.82E-03	1.20E-03	2.08E-03	9.26E-04	2.23E-03	2.89E-03
	RSQR	0.014	1.22E-03	6.43E-03	7.45E-03	1.89E-03	5.11E-04	7.33E-03	1.13E-03	1.19E-02	1.47E-03	1.16E-03
	OLEQR	0.045	3.96E-03	8.64E-02	2.47E-03	5.08E-03	2.38E-03	2.65E-03	6.34E-03	7.35E-03	5.98E-03	2.73E-03
$\tau = 0.5$	BQR	0.040	9.16E-04	5.78E-03	2.52E-03	1.62E-02	3.50E-03	6.26E-04	7.71E-05	2.70E-03	8.09E-03	3.13E-04
	BSQR	0.040	6.76E-04	2.23E-03	1.24E-03	2.08E-02	4.09E-03	2.03E-03	6.74E-04	3.48E-03	4.85E-03	5.43E-04
	RQR	0.009	7.25E-04	1.25E-03	2.56E-04	1.24E-03	4.41E-04	1.23E-03	1.18E-03	1.39E-03	2.68E-04	7.56E-04
	RSQR	0.007	1.27E-03	1.16E-03	1.96E-03	1.09E-03	1.20E-03	1.17E-03	5.11E-04	4.43E-04	8.50E-03	1.17E-03
	OLEQR	0.042	3.96E-03	8.64E-03	2.47E-03	5.08E-03	2.38E-03	2.65E-03	6.34E-03	7.35E-03	5.88E-03	2.73E-03
$\tau = 0.7$	BQR	0.046	3.60E-04	1.38E-04	3.47E-03	1.98E-02	9.46E-03	1.82E-04	2.77E-03	1.75E-03	5.60E-03	2.91E-03
	BSQR	0.039	5.66E-04	2.17E-03	1.20E-03	2.10E-02	4.09E-03	1.91E-03	4.35E-04	3.28E-03	4.64E-03	3.67E-04
	RQR	0.018	1.02E-03	1.64E-03	1.70E-03	1.72E-02	1.67E-03	1.31E-03	2.67E-03	2.44E-03	1.74E-02	2.41E-03
	RSQR	0.018	1.09E-03	1.16E-03	1.09E-03	8.27E-04	1.14E-03	2.03E-03	1.43E-03	1.83E-02	1.49E-03	1.48E-03
	OLEQR	0.045	1.02E-03	6.54E-02	4.93E-03	4.36E-03	4.55E-03	3.44E-03	3.85E-03	2.62E-02	7.03E-03	3.78E-03
$\tau = 0.9$	BQR	0.039	1.27E-04	5.80E-04	2.06E-05	2.14E-02	5.26E-04	2.35E-03	9.00E-05	3.59E-03	9.84E-03	1.00E-03
	BSQR	0.040	5.31E-04	2.15E-03	1.17E-03	2.15E-02	4.41E-03	1.74E-03	4.41E-04	3.26E-03	4.53E-04	4.44E-04
	RQR	0.021	2.81E-03	3.00E-03	2.66E-03	2.49E-02	2.31E-03	2.08E-03	1.90E-03	2.74E-03	2.18E-02	2.32E-03
	RSQR	0.021	1.22E-03	1.33E-02	5.12E-03	8.30E-03	4.78E-03	9.79E-03	1.54E-03	2.22E-02	5.98E-03	4.43-02
	OLEQR	0.047	1.51E-02	1.04E-02	3.63E-03	8.15E-03	6.74E-03	8.05E-03	6.05E-03	9.05E-03	1.05E-03	2.83E-03

Table 5. Comparison between Bayesian streaming data method and other methods in the case of t-distribution error distribution and m=50 batches.

τ	Method	MSE	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}
$\tau = 0.1$	BQR	0.049	6.53E-06	1.18E-04	6.64E-04	2.92E-02	1.81E-05	1.59E-03	2.72E-04	1.53E-03	3.32E-03	1.65E-03
	BSQR	0.048	1.31E-03	3.05E-03	2.45E-03	2.11E-02	3.91E-03	3.91E-03	1.35E-03	3.51E-03	6.63E-03	1.17E-03
	RQR	0.035	1.86E-03	1.89E-03	1.43E-03	2.84E-02	3.42E-03	3.72E-03	2.82E-03	6.85E-03	3.01E-02	5.39E-03
	RSQR	0.034	1.85E-03	2.53E-03	4.28E-03	6.49E-03	4.12E-03	3.83E-03	2.17E-03	4.32E-03	4.82E-03	4.10E-03
	OLEQR	0.061	1.02E-02	5.26E-03	1.05E-03	1.08E-02	3.86E-03	5.27E-03	5.60E-03	2.60E-02	2.14E-03	3.86E-02
$\tau = 0.3$	BQR	0.047	1.02E-03	3.02E-03	2.26E-03	2.11E-02	3.98E-03	3.77E-03	1.02E-04	3.48E-03	6.48E-02	9.08E-04
	BSQR	0.047	1.30E-03	2.59E-04	2.24E-03	2.27E-02	4.19E-03	2.29E-03	1.35E-03	3.59E-03	5.57E-03	1.06E-03
	RQR	0.021	2.43E-03	3.11E-03	1.84E-03	2.67E-02	1.74E-03	1.53E-03	1.12E-03	1.24E-03	2.34E-02	2.43E-03
	RSQR	0.020	3.80E-03	1.45E-03	3.64E-03	2.40E-03	3.36E-03	1.73E-03	2.04E-03	2.22E-02	2.70E-03	2.27E-03
	OLEQR	0.054	3.72E-03	5.24E-02	1.33E-02	3.39E-03	2.22E-03	1.06E-02	3.08E-03	5.92E-03	9.16E-03	2.19E-03
$\tau = 0.5$	BQR	0.044	8.24E-04	2.93E-03	1.83E-03	2.08E-02	3.68E-03	3.24E-03	8.19E-04	3.26E-03	6.18E-03	6.49E-04
	BSQR	0.046	1.05E-03	2.39E-03	2.10E-03	2.23E-02	4.43E-03	2.49E-03	1.03E-03	3.72E-03	5.57E-03	8.61E-04
	RQR	0.015	1.89E-03	8.26E-03	1.05E-03	1.10E-02	4.13E-03	6.59E-03	6.85E-03	1.10E-03	1.56E-03	1.26E-03
	RSQR	0.015	1.72E-03	9.26E-03	1.12E-03	9.30E-03	4.01E-03	3.79E-03	1.54E-03	3.21E-02	1.98E-03	5.43-04
	OLEQR	0.048	5.12E-03	1.63E-03	1.12E-03	8.30E-03	6.78E-03	7.79E-03	5.14E-03	5.12E-03	2.98E-03	5.51E-03
$\tau = 0.7$	BQR	0.045	4.79E-04	4.15E-03	1.61E-04	2.03E-02	4.15E-03	2.96E-04	1.21E-04	3.66E-04	6.12E-03	4.59E-04
	BSQR	0.044	9.14E-04	2.43E-03	1.14E-03	2.24E-02	4.82E-03	2.23E-03	8.31E-04	4.21E-03	5.79E-03	5.58E-04
	RQR	0.025	1.36E-03	2.11E-03	3.56E-03	1.38E-02	2.75E-03	2.94E-03	1.41E-03	3.16E-03	1.91E-02	2.20E-03
	RSQR	0.024	2.28E-03	1.41E-03	2.57E-03	3.55E-03	3.39E-03	1.94E-03	2.54E-03	1.41E-02	3.51E-03	2.86E-03
	OLEQR	0.050	1.12E-02	6.26E-03	1.26E-03	1.28E-03	3.86E-03	4.72E-03	5.60E-03	2.69E-03	2.14E-03	3.68E-03
$\tau = 0.9$	BQR	0.044	7.26E-03	2.63E-03	1.14E-04	2.26E-02	3.70E-03	2.77E-03	9.24E-04	3.57E-03	5.39E-03	6.72E-04
	BSQR	0.043	5.52E-04	2.72E-03	1.55E-03	2.13E-02	3.72E-03	2.88E-03	4.70E-04	3.56E-03	6.43E-03	5.92E-04
	RQR	0.033	6.73E-03	1.43E-03	3.51E-03	7.05E-02	2.25E-03	3.38E-03	4.35E-03	2.53E-03	2.70E-02	9.95E-04
	RSQR	0.030	6.23E-03	2.33E-03	8.12E-03	5.30E-03	1.78E-03	6.79E-03	3.54E-03	1.22E-02	8.98E-03	6.43E-03
	OLEQR	0.055	1.62E-02	1.32E-02	2.13E-03	1.35E-03	3.88E-03	4.75E-03	5.22E-03	2.74E-02	2.21E-03	3.72E-03

Table 6. Comparison between Bayesian streaming data method and other methods in the case of t-distribution error distribution and m=100 batches.

τ	Method	MSE	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}
$\tau = 0.1$	BQR	0.049	6.53E-06	1.18E-04	6.64E-04	2.92E-02	1.81E-05	1.59E-03	2.72E-04	1.53E-03	3.32E-03	1.65E-03
	BSQR	0.047	1.31E-03	3.05E-04	2.45E-03	2.11E-03	3.91E-03	3.91E-03	1.35E-03	3.51E-03	6.63E-03	1.17E-03
	RQR	0.038	3.69E-03	3.23E-03	5.21E-03	7.80E-03	3.15E-03	2.37E-03	2.69E-03	1.68E-03	3.52E-03	2.43E-03
	RSQR	0.036	1.85E-03	2.53E-03	5.34E-03	2.74E-03	2.38E-03	4.47E-03	5.16E-03	4.69E-03	4.07E-03	3.43E-03
	OLEQR	0.060	3.45E-03	6.55E-02	9.47E-03	2.66E-03	5.93E-03	8.90E-03	4.11E-03	8.99E-03	5.93E-03	1.11E-02
$\tau = 0.3$	BQR	0.047	1.02E-03	3.02E-04	2.22E-03	2.11E-02	3.98E-03	3.77E-03	1.02E-03	3.48E-03	6.48E-03	9.08E-04
	BSQR	0.048	1.04E-03	2.51E-04	2.90E-03	2.05E-02	3.43E-03	2.93E-03	1.43E-03	3.79E-03	9.09E-03	8.67E-04
	RQR	0.025	3.76E-03	3.25E-03	2.77E-03	1.32E-02	1.97E-03	1.76E-03	4.75E-03	1.90E-03	1.89E-02	3.55E-03
	RSQR	0.023	3.12E-03	3.33E-03	3.12E-03	2.30E-03	1.78E-03	1.79E-03	1.54E-02	4.22E-03	8.98E-03	3.43E-03
	OLEQR	0.051	5.12E-03	2.53E-02	5.12E-03	9.30E-03	6.78E-03	1.79E-03	1.54E-02	1.02E-02	6.98E-03	6.43E-03
$\tau = 0.5$	BQR	0.044	8.24E-04	2.93E-03	1.83E-03	2.08E-02	3.68E-03	3.24E-03	8.19E-04	3.26E-03	6.18E-03	6.49E-04
	BSQR	0.044	8.24E-04	2.93E-03	1.83E-03	2.08E-02	3.68E-03	3.24E-03	8.19E-04	3.26E-03	6.18E-03	6.49E-04
	RQR	0.018	8.93E-03	2.82E-03	2.23E-03	2.34E-02	2.08E-03	7.20E-03	1.07E-03	9.43E-03	1.15E-02	8.11E-03
	RSQR	0.018	8.67E-04	1.75E-02	2.93E-03	6.34E-04	2.84E-03	8.65E-04	2.20E-03	2.56E-02	1.96E-03	7.32E-03
	OLEQR	0.048	2.12E-03	3.33E-02	6.12E-03	5.30E-03	1.78E-03	1.79E-03	6.54E-03	1.22E-02	2.98E-03	1.43E-02
$\tau = 0.7$	BQR	0.045	4.79E-04	4.15E-03	1.61E-04	2.03E-02	4.15E-03	2.96E-04	1.21E-04	3.66E-04	6.12E-03	4.59E-04
	BSQR	0.043	5.06E-04	3.16E-03	1.81E-03	2.15E-02	4.28E-03	2.35E-03	6.68E-04	3.19E-03	5.22E-03	7.17E-04
	RQR	0.025	2.65E-03	1.66E-03	3.47E-03	4.19E-02	4.48E-03	2.56E-03	2.54E-03	1.05E-03	1.72E-02	1.88E-03
	RSQR	0.024	8.02E-03	2.63E-02	5.92E-03	5.30E-03	5.78E-03	3.79E-03	2.94E-03	3.12E-03	5.96E-03	2.03E-02
	OLEQR	0.051	4.32E-03	1.33E-02	5.12E-03	4.30E-03	4.78E-03	4.79E-03	6.14E-03	2.22E-03	5.98E-03	1.23E-02
$\tau = 0.9$	BQR	0.044	7.26E-03	2.63E-03	1.14E-04	2.26E-02	3.70E-03	2.77E-03	9.24E-04	3.57E-03	5.39E-03	6.72E-04
	BSQR	0.044	7.50E-04	2.20E-03	1.55E-03	2.14E-02	4.19E-03	2.96E-03	1.12E-04	3.73E-03	6.19E-03	7.48E-04
	RQR	0.040	3.01E-03	1.57E-03	7.49E-03	4.59E-02	7.03E-03	3.52E-03	1.78E-03	7.71E-03	4.06E-03	5.04E-03
	RSQR	0.040	5.84E-03	5.63E-03	1.02E-03	8.32E-03	4.52E-03	3.00E-03	2.22E-03	4.91E-03	2.21E-03	4.96E-03
	OLEQR	0.059	1.32E-03	1.33E-02	5.12E-03	4.30E-03	6.78E-03	2.79E-02	6.54E-03	1.02E-02	2.98E-03	1.43E-02

Additionally, we consider an interesting scenario 2 where streaming datasets arrive at a high speed and the batch size is small. For convenience, we fix the batch size $n_1 = \cdots = n_m = 100$, and let n increases from 10^5 to 10^6. The results are shown in Table 7 with *ALD* error distribution. The results for other quantile levels and error distributions are similar and thus omitted. We can see from Table 7 that the performance of our method improves (the MSE values all decrease) as the number of batches increase from 1000 to 10000, and it always works comparably with the oracle Bayesian quantile regression estimator. These results imply that our method is not affected as more data streams are processed; in other words, it is robust to the number of batches.

Table 7. MSE results of Bayesian experiment on streaming data with varying n and fixed batch sizes, $n_1 = n_2 = \cdots = n_m = 100$.

Method	N	$\tau = 0.3$	$\tau = 0.5$	$\tau = 0.7$
BSQR	10^5	0.056	0.052	0.057
	10^6	0.050	0.048	0.052
BQR	10^5	0.057	0.053	0.060
	10^6	0.052	0.048	0.054

For Algorithm 1, we conduct an analysis of its computational complexity. By analyzing the computational process of Algorithm 1, we determine that its computational complexity is $O((n_m k)^2)$. Furthermore, we record its running time and compare it with the distributed algorithm. For the convenience of recording and computational efficiency, we fix the batch size $n_1 = \cdots = n_m = 100$, and let m increases from 10 to 100. All results are based on 1000 draws obtained from the Gibbs samplers. Through this comparative analysis of running times, we observe that the streaming data algorithm has better computational efficiency. The comparison result is shown in Figure 1, where the horizontal axis represents the number of batches and the vertical axis represents the running time unit in minutes. Additionally, by comparing Tables 1–7 and Figure 2, it can be observed that our method is similar to the full data method, but our method has better effiency than the distributed algorithm. Therefore, the BSQR method can be used as an alternative to distributed algorithms to improve computational efficiency, especially when the number of data batches is large.

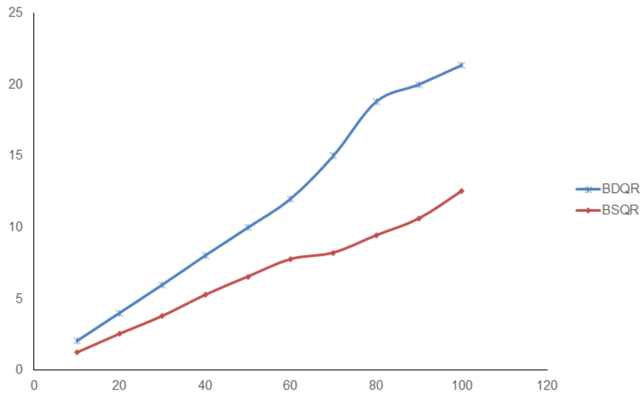

Figure 2. Comparative analysis of algorithm running times.

5. Real data analysis

Example 1. The online news popularity dataset. In this example, we illustrate our Bayesian streaming quantile regression method by using the Online News Popularity dataset. The dataset is from the UCI machine learning repository. This is a large-scale dataset with $N = 39,640$ observations. In our analysis, the number of shares in social networks is set as the response variable, and the perspective of web environment factors such as positive words, topic keywords, etc., are chosen as 7 predictors. For comprehensiveness, five different quantile levels from small to large are considered, i.e., $\tau = 0.1, 0.3, 0.5, 0.7$, and 0.9.

We fit our method to the above dataset by implementing Algorithm 1. In each τ, we partition the dataset into 100 subsets with the size of $n_m = 396$ for $m = 1, \cdots, 99$ and $n_{100} = 436$. All results are based on 15,000 draws obtained from the Gibbs samplers with a burn-in of 5000 iterations. The estimated coefficients and posterior standard deviations at the specified quantile levels are presented in Table 8.

Table 8. BSQR estimation of coefficient for the online news popularity data.

	τ=0.10		τ= 0.30		τ= 0.50		τ= 0.70		τ= 0.90	
	Coeff	Std	Coeff	Std	Coeff	Std	Coeff	Std	Coeff	Std
Intercept	3.1184	2.0746	5.2757	2.8087	5.9701	2.3922	6.2598	3.0828	9.5841	3.9952
length	0.0795	0.0362	0.1569	0.0387	0.1775	0.0619	0.2471	0.1224	0.3397	0.2512
num-keywords	0.4674	0.3502	0.6723	0.4962	0.7225	0.5124	0.7658	0.5625	0.7824	0.6314
title sentiment polarity	3.8472	3.2691	6.3682	4.2569	6.8923	4.3521	4.6130	3.9825	2.3569	1.9857
num-img	2.8009	1.7910	3.3239	2.1156	6.8448	3.0791	6.9821	4.1231	7.2656	4.5981
num-videos	1.0320	0.9432	1.6572	1.1152	1.8656	1.1974	1.9245	1.0358	2.0123	1.5699
num-positive	-0.6779	0.2505	-0.6185	0.3244	-0.5812	0.2216	-0.4213	0.1985	-0.3320	0.2569
num-negitive	1.4053	1.3506	1.5691	1.4651	2.2664	1.3015	2.5688	1.8856	2.9851	2.0344

We observe that the number of positive words is negatively correlated with the popularity, while length, num-img, num-video, and num-negitive have positive impacts across all quantiles. Additionally, as the quantile increases, the impact of title sentiment polarity first increases and then decreases, which means that the title of a popular news cannot be too polarized. This empirical study shows that the BSQR method we proposed is helpful in investigating the impact of different factors on news popularity in streaming data scenarios.

Example 2. The Beijing multi-site air-quality dataset. This example illustrates the streaming data algorithm for BSQR by using the Beijing multi-site air-quality dataset. The data were collected from the Beijing Municipal Environmental Monitoring Center, which is a reputable institution responsible for monitoring air quality, meteorological parameters, and other environmental factors in Beijing. The dataset includes measurements of pollutants such as $PM_{2.5}$, PM_{10}, NO_2, SO_2, CO, and O_3, as well as meteorological variables like temperature, humidity, wind speed, and wind direction. The data were recorded at multiple monitoring stations across the city, providing a spatially diverse dataset. The dataset is publicly available at https://www.bjmemc.com.cn/, spanning the period from July 1, 2017 to March 30, 2023. The dataset includes 430,524 hourly air pollutant data points from 12 district-controlled air quality monitoring stations in Beijing. The reasons for using air quality data are as follows: (1) Air quality data necessitates frequent real-time updates to accurately depict the current state of air quality. Streaming data frameworks are capable of rapidly processing and providing feedback on real-time data. (2) Air quality monitoring stations continuously and extensively generate data. These stations are dispersed across various locations, with each site persistently producing a substantial volume of data. Streaming data frameworks can efficiently handle this continuous data flow.

In this study, we explore the relationship between the $PM_{2.5}$ concentration (ug/m^3) and ten variables in Table 9. For model diagnostics, we use the AIC method for model selection. Specifically, we collected the historical Beijing multi-site air-quality dataset from March 2013 to February 2017, and calculated the AIC values under three different models based on the historical data, namely the classic linear regression model and the quantile regression model with $\tau = 0.1$ and 0.9. The AIC results are shown in Figure 3. From Figure 3, we can observe that the model with 10 variables is the best model.

Table 9. Covariates and their descriptions.

Name	Description
PM10	PM1.0 concentration (ug/m^3)
SO2	SO2 concentration (ug/m^3)
NO2	NO2 concentration (ug/m^3)
CO	CO concentration (ug/m^3)
O3	O3 concentration (ug/m^3)
TEMP	temperature (degrees Celsius)
PRES	pressure (hPa)
DEMP	dew point remperature (degrees Celsius)
RAIN	rainfall (mm/m^3)
WSPM	wind speed (m/s)

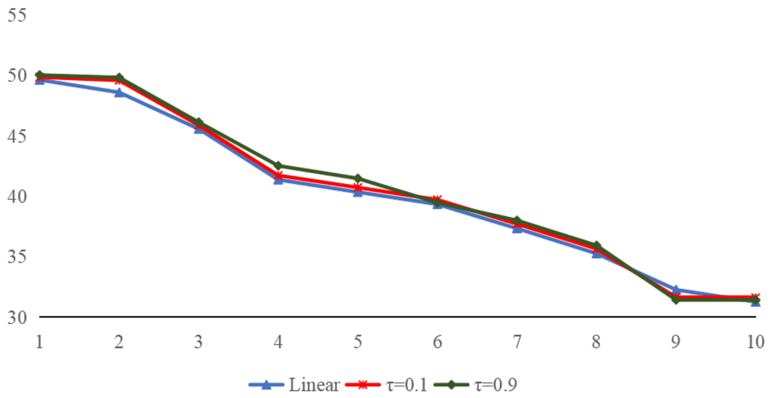

Figure 3. The results of AIC using historical data.

Because this data was collected by air quality monitoring sites in chronological order, we set the number of batches $b = 48$ based on the number of months. This air quality dataset and model have been analyzed in Jiang and Yu [37], which also focused on quantile regression for streaming data.

We fit the streaming data to the Bayesian quantile regression model by implementing Algorithm 1 for five quantile levels, specifically $\tau = 0.10$, 0.30, 0.50, 0.70, and 0.90. In each scenario, the entire set of observations is divided into 48 subsets. Subsequently, the informative g-prior is assigned with g=100. All reported results are based on 15,000 draws obtained from the Gibbs samplers, with a burn-in period of 5000 iterations. Table 10 and Figure 4 present the estimated coefficients for the specified quantile levels.

At different quantiles, the coefficients of five air factors are all significant and positive, indicating that these air pollutants have a positive impact on $PM_{2.5}$ concentration. As the quantile τ increases, the estimated coefficients of SO_2, NO_2, and O_3 decrease, which means that the impact of SO_2, NO_2, and O_3 on $PM_{2.5}$ concentration gradually decreases with the increase of $PM_{2.5}$ concentration. Additionally, the estimated coefficients of PM0 and CO increase with increasing quantile τ, indicating that as the concentration of $PM_{2.5}$ increases, the impact of PM_{10} and CO is increasing.

Table 10. BSQR estimation of the coefficient for the air quality data.

	τ=0.10		τ= 0.30		τ= 0.50		τ= 0.70		τ= 0.90	
	Coeff	Std	Coeff	Std	Coeff	Std	Coeff	Std	Coeff	Std
Intercept	-3.37E-04	1.12E-04	-4.58E-04	1.07E-04	-4.17E-04	1.25E-04	-2.83E-04	9.35E-05	-9.31E-04	3.93E-05
PM10	0.2051	1.78E-03	0.2661	2.79E-03	0.5478	3.27E-03	0.7472	2.95E-03	0.9397	2.34E-03
SO2	0.1393	7.49E-03	0.067	9.18E-03	0.0225	7.59E-03	0.0158	6.50E-03	0.0021	6.81E-03
NO2	0.0976	4.91E-03	0.0303	6.05E-03	0.0240	5.89E-03	-0.0730	5.53E-03	-0.075	3.53E-03
CO	0.0217	1.75E-04	0.0342	2.12E-04	0.0210	2.27E-04	0.0486	2.09E-04	0.0796	2.27E-04
O3	0.0873	2.04E-03	0.0902	2.81E-03	0.0692	3.19E-03	0.0507	2.14E-04	0.0111	1.45E-03
Temp	-0.2712	1.29E-02	-0.415	1.59E-02	-0.465	1.91E-03	-0.4800	1.48E-03	-0.8166	9.98E-03
PRES	-0.0083	3.27E-04	-0.0061	3.13E-04	-0.0004	3.79E-04	-0.0003	3.47E-04	-0.0003	2.09E-04
DEWP	0.3681	1.17E-02	0.5439	1.26E-02	0.6366	1.58E-02	0.5431	1.36E-02	0.9628	1.03E-03
RAIN	-0.4460	8.72E-02	-0.4162	1.05E-02	-0.4335	7.62E-01	-0.3509	6.97E-02	-0.2513	4.95E-02
WSPM	-0.4897	7.08E-02	-0.2915	9.63E-02	-0.1941	9.58E-02	-0.1218	8.23E-02	-0.0809	4.46E-02

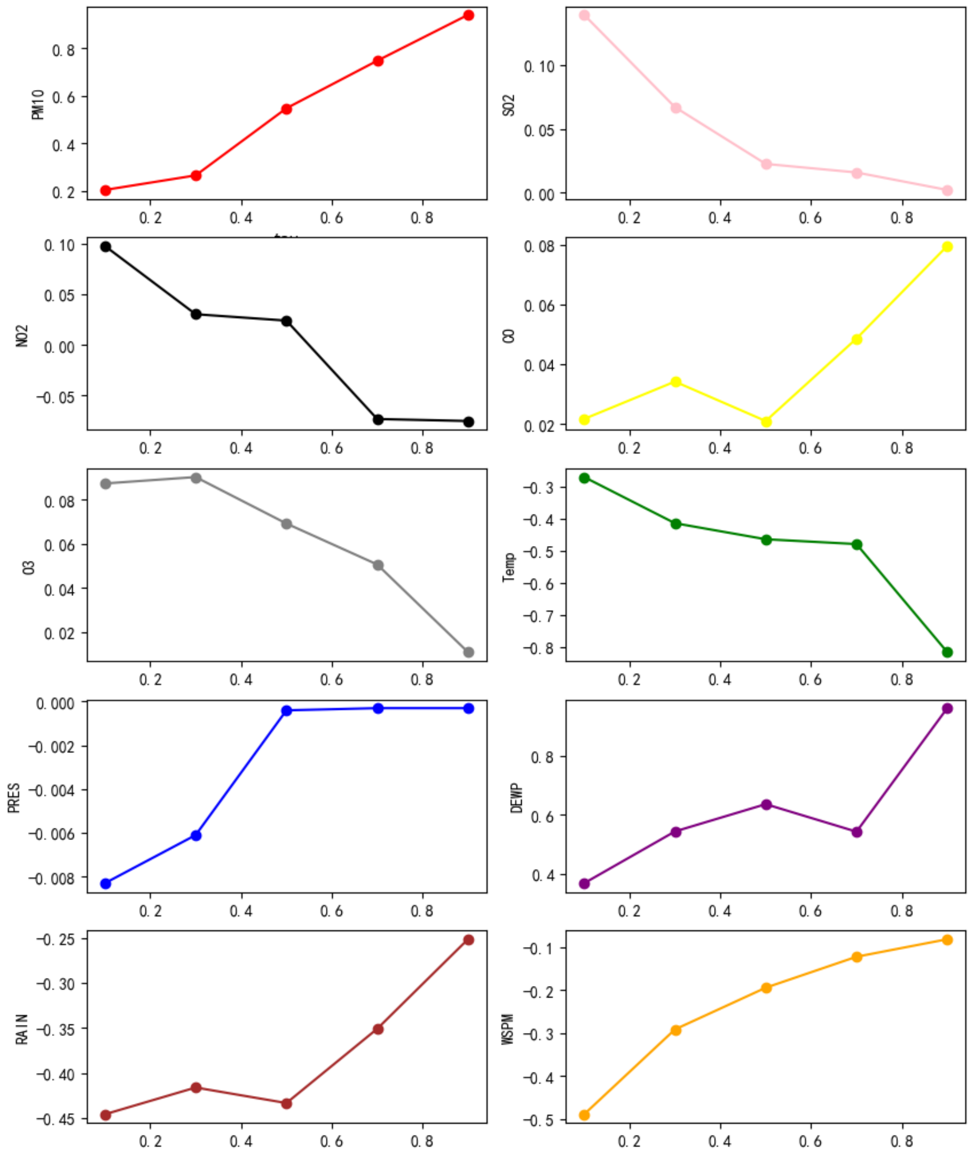

Figure 4. The estimated coefficients of BSQR under different quantiles for the air quality data.

For the other five meteorological factors, TEMP, PRES, DEWP, RAIN, and WSPM, RAIN and WSPM have significant and negative coefficients at each quantile, indicating that they have a negative impact on $PM_{2.5}$ concentration, and this effect become weaker as $PM_{2.5}$ concentration increases. The coefficient of PRES is not significant at the 0.9 quantile, indicating that air pressure has no significant effect on $PM_{2.5}$ at high concentrations. The TEMP coefficient is significantly negative at five quantiles, and the absolute value of the coefficient gradually increases, meaning that as the concentration of $PM_{2.5}$ increases, the impact of temperature becomes stronger. The coefficients of DEWP at each quantile are significantly positive, and as the quantile increases, the coefficients show an increasing trend. This indicates that as the concentration of $PM_{2.5}$ increases, the impact of DWEP increases.

Compared with the method in [37], we added three new covariates, PM_{10}, O_3, and RAIN, making the analysis of the real data more comprehensive. In addition, the conclusion of BSQR method indicates that the estimated coefficients of SO_2 and NO_2 decrease as quantity τ increases, and the estimated coefficients of WSPM are negative. These findings are different from the conclusions of [34], but are consistent with the research results of $PM_{2.5}$ in Li [38] and Zhang and Zhang [39].

6. Conclusions

This paper extends the traditional frequency QR. This extension involves the utilization of an ALD-based likelihood function, a conjugate NIG prior, and an adjusted NIG posterior. The streaming data method we proposed does not require storing historical data; it only needs to save the parameters of the posterior distribution and the current data. Furthermore, theoretical results establish the equivalency between the proposed posterior distribution for streaming data and the one computed using the full data. Both simulation studies and real data evaluations affirm the strong performance of the new approach proposed in this paper.

Real data analysis of Example 1 shows that the popularity of news is related to factors such as the number of positive words, images, video content, keywords and length. Among them, the number of positive words has a negative impact on the popularity of news. In Example 2, air pollutants and meteorological factors have varying impacts on $PM_{2.5}$ concentration, with the influence of some factors changing as $PM_{2.5}$ concentration increase. Specifically, the impact of air pollutants SO_2, NO_2, and O_3 decreases, while that of PM_{10} and CO increases. Among meteorological factors, the influence of RAIN and WSPM decreases, and the impact of TEMP and DEWP becomes more pronounced at high $PM_{2.5}$ concentrations.

This study reveals the degree and trend of the impact of different air pollutants and meteorological factors on $PM_{2.5}$ concentrations, which helps to more accurately understand the complex mechanisms of $PM_{2.5}$ formation and provides a scientific basis for air quality management. The following air quality management strategies are recommended: (1) Prioritize the control of SO_2, NO_2, and O_3 emissions, especially in areas with lower $PM_{2.5}$ concentrations, as these pollutants have a significant impact on $PM_{2.5}$; (2) in areas with higher $PM_{2.5}$ concentrations, strengthen the control of PM_{10} and CO to reduce their contribution to $PM_{2.5}$ concentrations; (3) strengthen the study of air pressure to determine its impact at different $PM_{2.5}$ concentrations, providing a more comprehensive basis for management strategies.

Nonetheless, there are several issues requiring future research. First, in conjunction with penalty functions such as LASSO [40] and SCAD [41], it is necessary to further explore the high-dimensional regularization for streaming data. Specifically, there are several versions of regularization that could be used [42]. In the frequency framework, Ma et al. [43] proposed an online updating coordinate descent algorithm, and a tuning lasso parameter selection was also suggested. Wang et al. [45] proposed a provable online feature selection algorithm that utilizes the online leverage score. For the Bayesian framework, we will explore Bayesian variable selection strategies in the streaming data context in our subsequent work. Based on current research, we aim to propose more accurate and efficient regularization methods for streaming data. Second, the new approach focuses on Bayesian linear QR models, and the issue of concept drift [44, 46] is not considered. Thus, one can further consider more complex Bayesian streaming data models like Bayesian deep neural networks, and invest the concept drift problem in the online BQR setting. Lastly, the new algorithm is geared to independent data, while in practical scenarios, dependent or non-stationary data [47] are more commonly used, including time series data or longitudinal data. Hence, it will be interesting to develop the corresponding algorithms for such kinds of datasets.

Acknowledgments

This research was funded by Natural Science Basic Research Program of Shaanxi (Program No. 2023-JC-QN-0059).

We would like to express our sincere gratitude to the editor and the reviewers for their valuable time, efforts, and expertise in reviewing our manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. M. Hilbert, Big data for development: A review of promises and challenges, *Dev. Policy. Rev.*, **34** (2016), 135–174. http://doi.org/10.1111/dpr.12142
2. C. Wang, J. Wu, J. Yan, Statistical methods and computing for big data, *Stat. Interface*, **9** (2016), 399. https://dx.doi.org/10.4310/SII.2016.v9.n4.a1
3. H. Wang, Y. Ma, Optimal subsampling for quantile regression in big data, *Biometrika*, **108** (2021), 99–112. https://doi.org/10.1093/biomet/asaa043
4. H. Wang, R. Zhu, P. Ma, Optimal subsampling for large sample logistic regression, *J. Am. Stat. Assoc.*, **117** (2022), 265–276. https://doi.org/10.1080/01621459.2020.1773832
5. X. Chen, W. Liu, X. Mao, Z. Yang, Distributed high-dimensional regression under a quantile loss function, *J. Mach. Learn. Res.*, **21** (2020), 7432–7474. https://doi.org/10.1214/18-AOS1777
6. A. Hu, Y. Jiao, Y. Liu, Y. Shi, Y. Wu, Distributed quantile regression for massive heterogeneous data, *Neurocomputing*, **448** (2021), 249–262. https://doi.org/10.1016/j.neucom.2021.03.041
7. R. Jiang, K. Yu, Smoothing quantile regression for a distributed system, *Neurocomputing*, **466** (2021), 311–326. https://doi.org/10.1016/j.neucom.2021.08.101
8. M. I. Jordan, J. D. Lee, Y. Yang, Communication-efficient distributed statistical inference, *J. Am. Stat. Assoc.*, **526** (2018), 668–681. https://doi.org/10.1080/01621459.2018.1429274
9. N. Lin, R. Xi, Aggregated estimating equation estimation, *Stat. Interface*, **4** (2011), 73–83. https://dx.doi.org/10.4310/SII.2011.v4.n1.a8
10. L. Luo, P. Song, Renewable estimation and incremental inference in generalized linear models with streaming data sets, *J. R. Stat. Soc. B*, **82** (2020), 69–97. https://doi.org/10.1111/rssb.12352
11. 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. https://doi.org/10.1080/01621459.2019.1710154

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

13. S. Mohamad, A. Bouchachia, Deep online hierarchical dynamic unsupervised learning for pattern mining from utility usage data, *Neurocomputing*, **390** (2020), 359–373. https://doi.org/10.1016/j.neucom.2019.08.093

14. H. M. Gomes, J. Read, A. Bifet, J. Paul, J. Gama, Machine learning for streaming data: State of the art, challenges, and opportunities, *ACM Sigkdd Explor. Newslett.*, **21** (2019), 6–22. https://doi.org/10.1145/3373464.3373470

15. L. Lin, W. Li, J. Lu, Unified rules of renewable weighted sums for various online updating estimations, *arXiv Preprint*, 2020. https://doi.org/10.48550/arXiv.2008.08824

16. C. Wang, M. 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. https://doi.org/10.1002/cjs.11330

17. J. Wu, M. Chen, Online updating of survival analysis, *J. Comput. Graph. Stat.*, **30** (2021), 1209–1223. https://doi.org/10.1080/10618600.2020.1870481

18. Y. Xue, H. Wang, J. Yan, E. D. Schifano, An online updating approach for testing the proportional hazards assumption with streams of survival data, *Biometrics*, **76** (2020), 171–182. https://doi.org/10.1111/biom.13137

19. S. Balakrishnan, D. Madigan, A one-pass sequential Monte Carlo method for Bayesian analysis of massive datasets, *Bayesian Anal.*, **1** (2006), 345–361. https://doi.org/10.1214/06-BA112

20. L. N. Geppert, K. Ickstadt, A. Munteanu, J. Quedenfeld, C. Sohler, Random projections for Bayesian regression, *Biometrics*, **27** (2017), 79–101. https://doi.org/10.1007/s11222-015-9608-z

21. R. Koenker, G. Bassett, Regression quantiles, *Econometrica*, 1978, 33–50. https://doi.org/10.2307/1913643

22. Y. Wei, A. Pere, R. Koenker, X. He, Quantile regression methods for reference growth charts, *Stat. Med.,* **25** (2006), 1369–1382. https://doi.org/10.1002/sim.2271

23. H. Wang, Z. Zhu, J. Zhou, Quantile regression in partially linear varying coefficient models, *Ann. Stat.*, 2009, 3841–3866. https://doi.org/10.1214/09-AOS695

24. X. He, B. Fu, W. K. Fung, Median regression for longitudinal data, *Stat. Med.*, **22** (2003), 3655–3669. https://doi.org/10.1002/sim.1581

25. M. Buchinsky, Changes in the US wage structure 1963–1987: Application of quantile regression, *Econometrica*, 1994, 405–458. https://doi.org/10.2307/2951618

26. A. J. Cannon, Quantile regression neural networks: Implementation in R and application to precipitation downscaling, *Comput. Geosci.*, **37** (2011), 1277–1284. https://doi.org/10.1002/sim.1581

27. Q. Xu, K. Deng, C. Jiang, F. Sun, X. Huang, Composite quantile regression neural network with applications, *Expert Syst. Appl.*, **76** (2017), 129–139. https://doi.org/10.1016/j.eswa.2017.01.054

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

29. L. Chen, Y. Zhou, Quantile regression in big data: A divide and conquer based strategy, *Comput. Stat. Data. An.*, **144** (2020), 106892. https://doi.org/10.1016/j.csda.2019.106892

30. K. Wang, H. Wang, S. Li, Renewable quantile regression for streaming datasets, *Knowl.-Based Syst.*, **235** (2022), 107675. https://doi.org/10.1016/j.knosys.2021.107675

31. Y. Chu, Z. Yin, K. Yu, Bayesian scale mixtures of normals linear regression and Bayesian quantile regression with big data and variable selection, *J. Comput. Appl. Math.*, **428** (2023), 115192. https://doi.org/10.1016/j.cam.2023.115192

32. K. Lum, A. E. Gelfand, Spatial quantile multiple regression using the asymmetric Laplace process, *Bayesian Anal.*, **7** (2012), 235–258. https://doi.org/10.1214/12-BA708

33. M. Smith, R. Kohn, Nonparametric regression using Bayesian variable, *J. Econometrics*, **75** (1996), 317–343. https://doi.org/10.1016/0304-4076(95)01763-1

34. M. Dao, M. Wang, S. Ghosh, K. Ye, Bayesian variable selection and estimation in quantile regression using a quantile-specific prior, *Computation. Stat.*, **37** (2022), 1339–1368. https://doi.org/10.1007/s00180-021-01181-5

35. K. E. Lee, N. Sha, E. R. Dougherty, M. Vannucci, B. K. Mallick, Gene selection: A Bayesian variable selection approach, *Bioinformatics*, **19** (2003), 90–97. https://doi.org/10.1093/bioinformatics/19.1.90

36. R. Chen, C. Chu , T. Lai, Y. Wu, Stochastic matching pursuit for Bayesian variable selection, *Stat. Comput.*, **21** (2011), 247–259. https://doi.org/10.1007/s11222-009-9165-4

37. R. Jiang, K. Yu, Renewable quantile regression for streaming data sets, *Neurocomputing*, **508** (2022), 208–224. https://doi.org/10.1016/j.knosys.2021.107675

38. X. Li, The influencing factors on $PM_{2.5}$ concentration of Lanzhou based on quantile eegression, *HGU. J.*, **41** (2018), 61–68. https://doi.org/10.13937/j.cnki.hbdzdxxb.2018.06.009

39. X. Zhang, W. Zhang, Spatial and temporal variation of $PM_{2.5}$ in Beijing city after rain, *Ecol. Environ. Sci.*, **23** (2014), 797–805. https://doi.org/10.3969/j.issn.1674-5906.2014.05.011

40. R. Tibshirani, Regression shrinkage and selection via the Lasso, *J. R. Stat. Soc. B*, **58** (2018), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

41. J. Fan, R. Li, Variable selection via nonconcave penalized likelihood and its oracle properties, *J. Am. Stat. Assoc.*, **96** (2011), 1348–1360. https://doi.org/10.1198/016214501753382273

42. F. E. Streib, M. Dehmer, High-dimensional LASSO-based computational regression models: Regularization, shrinkage, and selection, *Mach. Learn. Know. Extr.*, **1** (2019), 359–383. https://doi.org/10.3390/make1010021

43. X. Ma, L. Lin, Y. Gai, A general framework of online updating variable selection for generalized linear models with streaming datasets, *J. Stat. Comput. Sim.*, **93** (2023), 325–340. https://doi.org/10.1080/00949655.2022.2107207

44. A. Liu, J. Lu, F. Liu, G. Zhang, Accumulating regional density dissimilarity for concept drift detection in data streams, *Pattern Recogn.*, **76** (2018), 256–272. https://doi.org/10.1016/j.patcog.2017.11.009

45. J. Wang, J. Shen, P. Li, *Provable variable selection for streaming features*, International Conference On Machine Learning, **80** (2018), 5171–5179. Available from: `https://proceedings.mlr.press/v80/wang18g.html`.
46. J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang, Learning under concept drift: A review, *IEEE T. Knowl. Data En.*, **31** (2018), 2346–2363. https://doi.org/10.1109/TKDE.2018.2876857
47. R. Elwell, R. Polikar, Incremental learning of concept drift in nonstationary environments, *IEEE T. Neural Networ.*, **22** (2011), 1517–1531. https://doi.org/10.1109/TNN.2011.2160459
48. D. Rezende, S. Mohamed, *Variational inference with normalizing flows*, International Conference On Machine Learning, **22** (2015), 1530–1538. Available from: `https://proceedings.mlr.press/v37/rezende15`.
49. P. Müller, F. A. Quintana, A. Jara, T. Hanson, *Bayesian nonparametric data analysis*, New York: Springer Press, 2015. https://doi.org/10.1007/978-0-387-69765-9-7
50. R. Koenker, J. A. Machado, Goodness of fit and related inference processes for quantile regression, *J. Am. Stat. Assoc.*, **94** (1999), 1296–1310. https://doi.org/10.1109/TNN.2011.2160459
51. K. Yu, R. A. Moyeed, Bayesian quantile regression, *Stat. Probab. Lett.*, **54** (2001), 437–447. https://doi.org/10.1016/S0167-7152(01)00124-9
52. M. Geraci, Linear quantile mixed models: The lqmm package for Laplace quantile regression, *J. Stat. Softw.*, **57** (2014), 1–29. https://doi.org/10.18637/jss.v057.i13
53. M. Geraci, M. Bottai, Quantile regression for longitudinal data using the asymmetric laplace distribution, *Biostatistics*, **8** (2007), 140–154. https://doi.org/10.1093/biostatistics/kxj039
54. D. F. Benoit, D. V. den Poel, bayesQR: A Bayesian approach to quantile regression, *J. Stat. Softw.*, **76** (2017), 1–32. https://doi.org/10.18637/jss.v076.i07

Appendix

For streaming data $D_1, D_2, ...D_m, \cdots$, the NIG form of the posterior distribution for D_{m-1} is,

$$f\left(\beta, \sigma | Y^*_{\tau,m-1}, X^*_{m-1}, V\right) \propto NIG\left(\bar{\mu}_{\tau,m-1}, \bar{\Lambda}_{m-1}, \bar{a}_{m-1}, \bar{b}_{\tau,m-1}\right),$$

where, $\bar{\mu}_{\tau,m-1} = \left[(1+\frac{1}{g})X^{*T}_{m-1}\Sigma^{-1}_{m-1}X^*_{m-1}\right]^{-1} X^{*T}_{m-1}\Sigma^{-1}_{m-1}Y^*_{\tau,m-1}$, $\bar{\Lambda}_{m-1} = (1+\frac{1}{g})X^{*T}_{m-1}\Sigma^{-1}_{m-1}X^*_{m-1}$, $\bar{a}_{m-1} = \frac{3}{2}n$, $\overline{b}_{\tau,m-1} = \frac{1}{2}Y^{*T}_{\tau,m-1}\Sigma^{-1}_{m-1}Y^*_{\tau,m-1} - \frac{1}{2}\bar{\mu}^T_{\tau,m-1}\bar{\Lambda}_{m-1}\bar{\mu}_{\tau,m-1}$.

Thus, the likelihood function of D_m data is

$$\begin{aligned} f(Y^*_{\tau,m}|X^*_m, \beta, \sigma, V, \Sigma) &\propto \sigma^{-\frac{n_m-k}{2}} exp\{-\frac{1}{2\sigma}[Y^*_{\tau,m} - {X^*_m}^T\hat{\beta}_{\tau,m}]\Sigma^{-1}_m[Y^*_{\tau,m} - {X^*_m}^T\hat{\beta}_{\tau,m}]\} \\ &\quad \sigma^{-\frac{k}{2}} exp\{-\frac{1}{2\sigma}(\beta - \hat{\beta}_{\tau,m})^T\Lambda_m(\beta - \hat{\beta}_{\tau,m})\} \\ &= (\sigma)^{-(a+\frac{k}{2}+1)} exp\{-\frac{1}{\sigma}[b_{\tau,m} + \frac{1}{2}(\beta - \mu_{\tau,m})^T\Lambda(\beta - \mu_{\tau,m})]\}. \end{aligned}$$

The likelihood function has a representation of the NIG distribution $NIG(\mu_{\tau,m}, \Lambda_m, a_m, b_{\tau,m})$, which means

$$f(Y^*_{\tau,m}|X^*_m, \beta, \sigma, V, \Sigma) \propto NIG\left(\mu_{\tau,m}, \Lambda_m, a_m, b_{\tau,m}\right),$$

where, $\mu_{\tau,m} = \hat{\beta}_{\tau,m}$, $\Lambda = X_m^{*T}\Sigma_m^{-1}X_m^*$, $a_m = \frac{n_m-k-2}{2}$, $b_{\tau,m} = \frac{1}{2}[Y_{\tau,m}^* - X_m^*\hat{\beta}_{\tau,m}]^T\Sigma_m^{-1}[Y_{\tau,m}^* - X_m^*\hat{\beta}_{\tau,m}]$.

According to Remark 1, the prior of the Bayesian algorithm for streaming data consists of the adjusted historical posterior and the g-NIG prior distribution of the current data information

$$NIG(\bar{\mu}_{\tau,m-1}, \bar{\Lambda}_{m-1}, \bar{a}_{m-1} - \frac{k+2}{2}, \bar{b}_{\tau,m-1}) \times NIG(\mu_0, \Lambda_{m,0}, 0, 0).$$

The prior distribution of the m-th batch of streaming data is defined as

$$NIG(\mu_{p,\tau,m}, \Lambda_{p,m}, a_{p,m}, b_{p,\tau,m}),$$

where,

$$\begin{aligned}
\mu_{p,\tau,m} &= (\bar{\Lambda}_{m-1} + \Lambda_{m,0})^{-1}(\bar{\Lambda}_{m-1}\bar{\mu}_{\tau,m-1}) \\
&= [(1+\frac{1}{g})\sum_{i}^{m-1} X_i^{*T}\Sigma_i^{-1}X_i^* + \frac{X_m^{*T}\Sigma_m^{-1}X_m^*}{g}]^{-1}(X_{m-1}^{*T}\Sigma_{m-1}^{-1}Y_{\tau,m-1}^*),
\end{aligned}$$

$$\Lambda_{p,m} = \bar{\Lambda}_{m-1} + \Lambda_{m,0} = (1+\frac{1}{g})\sum_{i}^{m-1} X_i^{*T}\Sigma_i^{-1}X_i^* + \frac{X_m^{*T}\Sigma_m^{-1}X_m^*}{g},$$

$$a_{p,m} = \bar{a}_{m-1} + a_0 + \frac{k+2}{2} - \frac{k+2}{2} = \frac{n_{m-1}}{2},$$

$$b_{p,\tau,m} = \bar{b}_{\tau,m-1} + b_0 + \frac{1}{2}(\bar{\mu}_{\tau,m-1} - \mu_{p,\tau,m})^T\Lambda_{m-1}(\bar{\mu}_{\tau,m-1} - \mu_{p,\tau,m}) + \frac{1}{2}\mu_{p,\tau,m}^T\Lambda_{m,0}\mu_{p,\tau,m}.$$

Now, we can infer that the posterior distribution of our algorithm in m-th batch is

$$NIG_k(\bar{\mu}_{\tau,m}, \bar{\Lambda}_m, \bar{a}_m, \bar{b}_{\tau,m}) = NIG(\mu_{p,m}, \Lambda_{p,m}, a_{p,m}, b_{p,m}) \times NIG\left(\mu_{\tau,m}, \Lambda_m, a_m, b_{\tau,m}\right).$$

According to Proposition 1, we can obtain the NIG posterior parameters as follows,

$$\begin{aligned}
\bar{\mu}_{\tau,m} &= (\Lambda_{p,m} + \Lambda_m)^{-1}(\Lambda_{p,m}\mu_{p,\tau,m} + \Lambda_m\mu_{\tau,m}) \\
&= ((1+\frac{1}{g})\sum_{i}^{m-1} X_i^{*T}\Sigma_i^{-1}X_i^* + \frac{X_m^{*T}\Sigma_m^{-1}X_m^*}{g} + X_m^{*T}\Sigma_m^{-1}X_m^*)^{-1}(X_{m-1}^{*T}\Sigma_{m-1}^{-1}Y_{\tau,m-1}^* + X_m^{*T}\Sigma_m^{-1}Y_{\tau,m}^*) \\
&= ((1+\frac{1}{g})\sum_{i}^{m} X_i^{*T}\Sigma_i^{-1}X_i^*)^{-1}(\sum_{i}^{m} X_i^{*T}\Sigma_i^{-1}Y_{\tau,i}^*),
\end{aligned}$$

$$\bar{\Lambda}_m = \Lambda_{p,m} + \Lambda_m = (1+\frac{1}{g})\sum_{i=1}^{m-1} X_i^{*T}\Sigma_i^{-1}X_i^* + \frac{X_m^{*T}\Sigma_m^{-1}X_m^*}{g} + X_m^{*T}\Sigma_m^{-1}X_m^* = (1+\frac{1}{g})\sum_{i=1}^{m} X_i^{*T}\Sigma_i^{-1}X_i^*,$$

$$\bar{a}_m = a_{p,m} + a_m + \frac{k+2}{2} = \frac{n_{m-1}}{2} + \frac{n_m-k-2}{2} + \frac{k+2}{2} = \frac{n_m + N_{m-1}}{2} = \frac{N_m}{2},$$

$$\begin{aligned}
\bar{b}_{\tau,m} &= b_{p,\tau,m} + b_m + \frac{1}{2}(\mu_{p,m} - \bar{\mu}_{\tau,m})^T \Lambda_{p,m}(\mu_{p,\tau,m} - \bar{\mu}_{\tau,m}) + \frac{1}{2}(\mu - \bar{\mu}_{\tau,m})^T \Lambda_m(\mu - \bar{\mu}_{\tau,m}) \\
&= \frac{1}{2}\sum_{i}^{m-1} Y_{\tau,i}^{*T}\Sigma_i Y_{\tau,i}^{*} - \frac{1}{2}\bar{\mu}_{\tau,m-1}^T \bar{\Lambda}_{m-1}\bar{\mu}_{\tau,m-1} - \frac{1}{2}\mu_{p,\tau,m}^T\bar{\Lambda}_{m-1}\bar{\mu}_{\tau,m-1} + \frac{1}{2}\mu_{p,\tau,m}^T\bar{\Lambda}_{m-1}\mu_{p,\tau,m} \\
&+ \frac{1}{2}\mu_{p,\tau,m}^T\Lambda_{m,0}\mu_{p,\tau,m} + \frac{1}{2}Y_{\tau,m}^{*T}\Sigma_m Y_{\tau,m}^{*} - \frac{1}{2}Y_{\tau,m}^{*T}\Sigma_m X^{*T}\mu_{\tau,m} - \frac{1}{2}\mu_{\tau,m}^T X_m^{*T}\Sigma_m^{-1}Y_{\tau,m}^{*T} + \frac{1}{2}\mu_{\tau,m}^T X_m^{*T}\Sigma_m^{-1}X_m^{*}\mu_{\tau,m} \\
&+ \frac{1}{2}\mu_{p,\tau,m}^T\Lambda_{p,m}\mu_{p,\tau,m} - \frac{1}{2}\mu_{p,\tau,m}^T\Lambda_{p,m}\bar{\mu}_{\tau,m} - \frac{1}{2}\bar{\mu}_{\tau,m}^T\Lambda_{p,m}\mu_{p,\tau,m} + \frac{1}{2}\bar{\mu}_{\tau,m}^T\Lambda_{p,m}\bar{\mu}_{\tau,m} \\
&+ \frac{1}{2}\mu_{\tau,m}^T\Lambda_m\mu_{\tau,m} - \frac{1}{2}\mu_{\tau,m}^T\Lambda_m\bar{\mu}_{\tau m} - \frac{1}{2}\bar{\mu}_{\tau m}^T\Lambda_m\mu_{\tau,m} + \frac{1}{2}\bar{\mu}_{\tau,m}^T\Lambda_m\bar{\mu}_{\tau,m} \\
&= \frac{1}{2}\sum_{i=1}^{m-1} Y_{\tau,i}^{*T}\Sigma_i Y_{\tau,i}^{*} + \frac{1}{2}Y_{\tau,m}^{*T}\Sigma_m Y_{\tau,m}^{*} - \frac{1}{2}\sum_{i=1}^{m-1} Y_{\tau,i}^{*T}\Sigma_i X_i^{*}\bar{\mu}_{\tau,m} \\
&- \frac{1}{2}\bar{\mu}_{\tau,m}^T\sum_{i=1}^{m-1} X_i^{*T}\Sigma_i Y_{\tau,i}^{*} + \frac{1}{2}\bar{\mu}_{\tau,m}^T(\Lambda_{p,m} + \Lambda_m)\bar{\mu}_{\tau,m} - \frac{1}{2}Y_{\tau,m}^{*T}\Sigma_m^{-1}X^{*}\bar{\mu}_{\tau,m} - \frac{1}{2}\bar{\mu}_m X_m^{*T}\Sigma_m Y_{\tau,m}^{*} \\
&= \frac{1}{2}\sum_{i=1}^{m-1} Y_{\tau,i}^{*T}\Sigma_i Y_{\tau,i}^{*} + \frac{1}{2}Y_{\tau,m}^{*T}\Sigma_m Y_{\tau,m}^{*} - \frac{1}{2}\bar{\mu}_{\tau,m}^T\sum_{i=1}^{m-1} X_i^{*T}\Sigma_i Y_{\tau i}^{*} - \frac{1}{2}\bar{\mu}_m X_m^{*T}\Sigma_m Y_{\tau,m}^{*} \\
&= \frac{1}{2}\sum_{i=1}^{m} Y_{\tau,i}^{*T}\Sigma_i Y_{\tau,i}^{*} - \frac{1}{2}\bar{\mu}_{\tau,m}^T\sum_{i=1}^{m} X_i^{*T}\Sigma_i Y_{\tau,i}^{*}.
\end{aligned}$$

Thus, the proof of Theorem is completed.