Electronic Journal of Statistics
Vol. 17 (2023) 3443–3471
ISSN: 1935-7524
https://doi.org/10.1214/23-EJS2182

Online inference in high-dimensional generalized linear models with streaming data

Lan Luo*

Department of Biostatistics and Epidemiology, Rutgers School of Public Health, New Jersey, USA
e-mail: l.luo@rutgers.edu

Ruijian Han*

Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, China
e-mail: ruijian.han@polyu.edu.hk

Yuanyuan Lin†

Department of Statistics, The Chinese University of Hong Kong, Hong Kong, China
e-mail: ylin@sta.cuhk.edu.hk

Jian Huang†

Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, China
e-mail: j.huang@polyu.edu.hk

Abstract: In this paper we develop an online statistical inference approach for high-dimensional generalized linear models with streaming data for real-time estimation and inference. We propose an online debiased lasso method that aligns with the data collection scheme of streaming data. Online debiased lasso differs from offline debiased lasso in two important aspects. First, it updates component-wise confidence intervals of regression coefficients with only summary statistics of the historical data. Second, online debiased lasso adds an additional term to correct approximation errors accumulated throughout the online updating procedure. We show that our proposed online debiased estimators in generalized linear models are asymptotically normal. This result provides a theoretical basis for carrying out real-time interim statistical inference with streaming data. Extensive numerical experiments are conducted to evaluate the performance of our proposed online debiased lasso method. These experiments demonstrate the effectiveness of our algorithm and support the theoretical results. Furthermore, we illustrate the application of our method with a high-dimensional text dataset.

MSC2020 subject classifications: Primary 62J07; secondary 62J12, 62F25.

*Equally contributing authors.
†Co-corresponding authors.

Keywords and phrases: Confidence interval, generalized linear models, online debiased lasso, high-dimensional data.

Received September 2022.

Contents

1. Introduction

Streaming data refers to the type of data that is generated continuously over time, typically in high volumes and at high velocity. It includes a wide variety of data types such as log files generated by mobile or web applications, ecommerce purchases, information from social networks, and financial trading floors. To reduce the demand on computing memory and achieve real-time processing, the nature of streaming data calls for the development of incremental algorithms that do not require access to the full dataset. In this paper, we focus on the generalized linear models in a high-dimensional regression setting. We develop a real-time estimation and inference procedure that is highly scalable with respect to fast growing data volumes, but with no loss of efficiency in statistical inference in the presence of a large number of features.

Streaming data processing essentially falls into the field of online learning. This line of research can be dated back seven decades to Robbins and Monro

[21], who proposed a stochastic approximation algorithm that laid a foundation for the popular stochastic gradient descent (SGD) algorithm [22]. The SGD algorithm and its variants have been extensively studied for online estimation and prediction [28], but the work of developing online statistical inference remains unexplored. A recent paper by Fang [11] proposed a perturbation-based resampling method to construct confidence intervals for SGD, but it does not achieve desirable statistical efficiency and may produce misleading inference in high-dimensional settings. In addition to the SGD types of recursive algorithms, several online updating methods have been proposed to specifically perform sequential updating of regression coefficient estimators, including the online least squares estimator for the linear model, the cumulative estimating equation estimator, the cumulatively updated estimating equation estimator by Schifano et al. [23] and the renewable estimator by Luo and Song [18] for nonlinear models.

Most of the aforementioned online algorithms are developed under low dimensional settings where the number of features is far less than the total sample size. However, a prominent concern in high-dimensional streaming data analysis is that only a subset of the variables have nonzero coefficients. Besides the small sample size issue at the early stage of data collection, processing such data stream without properly accounting for the sparsity in feature set may introduce significant bias and invalid statistical inference. It is worth noting that even if the cumulative sample size exceeds the number of features as time goes by, traditional estimation methods in low-dimensional settings such as maximum likelihood estimation (MLE) may still incur large bias especially in generalized linear models [26]. Therefore, current state-of-art online learning algorithms in low-dimensional settings may be insufficient for processing high-dimensional data streams.

In the traditional offline settings, many methods have been developed for analyzing high-dimensional static data. Most of the work on variable selection in high dimensional regression problems is along the line of lasso [27], the Smoothly Clipped Absolute Deviation (SCAD) penalty [10], and the minimax convex penalty (MCP) [31]. However, variable selection methods focus on point estimation without quantifying the uncertainty in estimates. Later on, statistical inference problems in high-dimensional settings, including interval estimation and hypothesis testing, have attracted much attention since the pioneering works of Zhang and Zhang [32], van de Geer et al. [30], Javanmard and Montanari [16], among others. Recently, a novel splitting and smoothing inference approach for high-dimensional generalized linear models was proposed by Fei and Li [12].

While significant progress has been made on statistical inference for high dimensional regression problems under the traditional offline settings, variable selection and statistical inference for high-dimensional models with streaming data is still at its infancy stage. Sun et al. [25] introduced a systematic framework for online variable selection based on some popular offline methods such as MCP. But their focus is not on statistical inference. Different from this work, there are some existing methods considering the problem of inference. For example,

Deshpande, Javanmard and Mehrabi [7] proposed a class of online estimators for high-dimensional auto-regressive models. One of the most relevant works is a novel inference procedure in generalized linear models based on recursive online-score estimation [24]. However, in both works, the entire dataset is assumed to be available at an initial stage for computing an initial estimator, e.g. the lasso estimator; thereafter, a recursively forward bias correction procedure is conducted along sequentially arrived data points. However, the availability of the entire dataset at an initial stage is not a natural setup in online learning. To address this issue, Han et al. [14] proposed an online debiased lasso method for statistical inference in high-dimensional linear models with streaming data.

Unlike the case of high-dimensional linear models where the loss function depends on data only through sufficient statistics [14], parameters and data are not linearly separable in generalized linear models. Motivated by the renewable estimation method in low-dimensional generalized linear models [18], we start off by taking a first-order Taylor expansion on the quadratic loss function to bypass the need of historical individual-level data. The key idea centers around using "approximate summary statistics" resulting from Taylor expansions. However, this is not a trivial extension of the methods developed under low-dimensional settings. In high-dimensional settings where predictors are spuriously correlated, a data-splitting strategy is typically used for decorrelation where variable selection and estimation are conducted using two different sub-datasets [24, 12]. A prominent concern of using such approximate summary statistics that involve previous estimates is that it may incur dependency in the corresponding estimating equation. Theoretically speaking, the dependency among recursively updated estimators poses extra technical challenge in establishing the non-asymptotic error bound. In our proposed online method for real-time confidence interval construction, we aim to address the following questions: (i) what types of approximate summary statistics to be stored to carry out an online debiasing procedure? (ii) will the error accumulate along the updating steps if we use the approximate summary statistics? (iii) will the online debiasing procedure maintain similar oracle properties to its offline counterpart? and (iv) how to choose the tuning parameter adaptively in an online setting where cross-validation that relies on splitting the entire dataset is not feasible.

The focus of this paper is to develop an online debiased lasso estimator in high-dimensional generalized linear models with streaming datasets for real-time estimation and inference. Our new contributions include: (i) we propose a two-stage online estimation and debiasing framework that aligns with streaming data collection scheme; (ii) online debiased lasso accounts for sparsity feature in a candidate set of predictors and provides valid statistical inference results; and (iii) online debiased lasso estimators for the generalized linear models are shown to be asymptotically normal. This result provides a theoretical basis for carrying out real-time interim statistical inference with streaming data. Online debiased lasso is inspired by the offline debiased lasso method [32, 30, 16], however, it differs from the offline debiased lasso in two important aspects. First, in computing the estimate at the current stage, it only uses summary statistics

of the historical data. Second, in addition to debiasing an online lasso estimator, online debiased lasso corrects an approximation error term arising from online updating with streaming data. This correction is crucial to guarantee the asymptotic normality of the online debiased lasso estimator.

This paper is organized as follows. Section 2 introduces the model formulation followed by our proposed online two-stage debiasing method to process high-dimensional streaming data. Section 3 includes some large sample properties concerning the theoretical guarantees for our proposed method. Simulation experiments are given in Section 4 to evaluate the performance of our proposed method in comparison to both MLE and offline debiased estimator. We illustrate the proposed online debiased lasso method and apply it to analyze a real data example in Section 5. Finally, we make some concluding remarks in Section 6. All technical proofs are provided in the supplementary material.

1.1. Notation

For a matrix $\boldsymbol{X} \in \mathbb{R}^{n\times p}$, we let $\boldsymbol{X}_{i\cdot}, \boldsymbol{X}_{\cdot j}$ and $\boldsymbol{X}_{ij}$ denote the i-th row, j-th column and (i,j)-element of matrix $\boldsymbol{X}$. $\boldsymbol{X}_{i,-j}$ is a sub-vector of $\boldsymbol{X}_{i\cdot}$ with the j-th element deleted and $\boldsymbol{X}_{-i,-j}$ is a sub-matrix of $\boldsymbol{X}$ with the i-th row and the j-th column deleted while other elements remain unchanged. For a sequence of random variables $\{\xi_n\}_{n\in\mathbb{N}}$ and a corresponding sequence of constants $\{a_n\}_{n\in\mathbb{N}}$. We say that $\xi_n = \mathcal{O}_p(a_n)$ if for any $\epsilon > 0$, there exist two finite numbers $M, N > 0$ such that $P(|\xi_n/a_n| > M) < \epsilon$ for any $n > N$. Generally speaking, $\xi_n = \mathcal{O}_p(a_n)$ denotes ξ_n/a_n is stochastically bounded. $\xi_n = o_p(a_n)$ means that ξ_n/a_n converges to zero in probability. With the consideration of the streaming data, we use $\boldsymbol{X}^{(j)}$ and $\boldsymbol{Y}^{(j)}$ to stand for $\boldsymbol{X}$ and $\boldsymbol{y}$, arriving in j-th batch respectively. In addition, $\boldsymbol{X}_\star^{(j)}$ and $\boldsymbol{Y}_\star^{(j)}$ (with star index) are the cumulative variables of $\boldsymbol{X}^{(j)}$ and $\boldsymbol{Y}^{(j)}$. For example, $\boldsymbol{X}_\star^{(j)} = ((\boldsymbol{X}^{(1)})^\top, \ldots, (\boldsymbol{X}^{(j)})^\top)^\top$. For a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, let $\Lambda_{\max}(\boldsymbol{A})$ and $\Lambda_{\min}(\boldsymbol{A})$ denote the maximum and minimum eigenvalues of $\boldsymbol{A}$ respectively.

2. Methodology

In this section, we describe the proposed estimation method with streaming data, including online lasso estimation and online debiased lasso estimation. With the consideration of practical implementation, we also provide an adaptive tuning method to select the regularization parameter. A rundown of our algorithm is summarized at the end of this section.

Consider up to a time point $b \geq 2$, there is a total of N_b samples arriving in a sequence of b data batches, denoted by $\mathcal{D}_b^\star = \{\mathcal{D}_1, \ldots, \mathcal{D}_b\}$, and each contains $n_j = |\mathcal{D}_j|$ samples, $j = 1, \ldots, b$. Assume each observation y_i is independently sampled from the generalized linear model with density function

$$f(y \mid \boldsymbol{x}; \boldsymbol{\beta}^0) = a(y) \exp\left\{\frac{y\boldsymbol{x}\boldsymbol{\beta}^0 - g(\boldsymbol{x}\boldsymbol{\beta}^0)}{\phi_0}\right\},$$

where $g(\cdot)$ is a convex function and ϕ_0 is a fixed and known parameter, and $a(\cdot)$ is a normalizing factor. The underlying regression coefficient $\boldsymbol{\beta}^0 \in \mathbb{R}^p$ is of our interest, which is assumed to be sparse with s_0 nonzero elements. Specifically, we let $S_0 = \{r : \beta_r^0 \neq 0\}$ be the active set of variables and its cardinality is s_0. Our main goal is to conduct point-wise statistical inference for the components of the parameter vector β_r^0 $(r = 1, \ldots, p)$ upon the arrival of every new data batch $\mathcal{D}_j$, $j = 1, 2, \ldots, b$. The log-likelihood function for the cumulative dataset $\mathcal{D}_b^\star$ is

$$\begin{aligned}\ell(\boldsymbol{\beta}; \mathcal{D}_b^\star) &= \frac{1}{N_b} \sum_{i \in \mathcal{D}_b^\star} \log f(y_i \mid \boldsymbol{x}_i, \boldsymbol{\beta}) \\ &= \frac{1}{N_b} \sum_{i \in \mathcal{D}_b^\star} \log a(y_i) - \frac{1}{2N_b\phi_0} \sum_{i \in \mathcal{D}_b^\star} \{g(\boldsymbol{x}_i\boldsymbol{\beta}) - y_i \boldsymbol{x}_i\boldsymbol{\beta}\}.\end{aligned}$$

Based on $\mathcal{D}_b^\star$, the standard offline lasso estimator is defined as

$$\bar{\boldsymbol{\beta}}^{(b)} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left\{ \frac{1}{2N_b} \sum_{i \in \mathcal{D}_b^\star} \{g(\boldsymbol{x}_i\boldsymbol{\beta}) - y_i \boldsymbol{x}_i\boldsymbol{\beta}\} + \lambda_b \|\boldsymbol{\beta}\|_1 \right\}, \tag{2.1}$$

where $N_b = \sum_{j=1}^b n_j$ is the cumulative sample size and λ_b is the regularization parameter. However, as discussed in Luo and Song [18] and Han et al. [14], the classical lasso estimator obtained through minimizing equation (2.1) requires re-accessing the historical raw data $\mathcal{D}_{b-1}^\star$ which is not preferable in an online setting. Therefore, an online estimation and debiasing procedure needs to be developed for real-time estimation and inference with high-dimensional streaming data. For the sake of clarity, we refer to the lasso estimator in (2.1) as the offline lasso estimator. Sections 2.1 and 2.2 are devoted to the construction of the online lasso estimator and the online debiased method, respectively.

2.1. Online lasso estimator

We first consider an online lasso estimator through the gradient descent method. Define the score function as $\boldsymbol{u}(y_i; \boldsymbol{x}_i, \boldsymbol{\beta}) = \nabla_{\boldsymbol{\beta}} \{g(\boldsymbol{x}_i\boldsymbol{\beta}) - y_i\boldsymbol{x}_i\boldsymbol{\beta}\} = \boldsymbol{x}_i^\top (g'(\boldsymbol{x}_i\boldsymbol{\beta}) - y_i)$, and the aggregated score function for the cumulative dataset $\mathcal{D}_b^\star$ is $\bar{\boldsymbol{U}}^{(b)}(\boldsymbol{\beta}) = \sum_{i \in \mathcal{D}_b^\star} \boldsymbol{u}(y_i; \boldsymbol{x}_i, \boldsymbol{\beta})$. Let $U^{(j)}(\boldsymbol{\beta}) = \sum_{i \in \mathcal{D}_j} \boldsymbol{u}(y_i; \boldsymbol{x}_i, \boldsymbol{\beta})$ be the score function for data batch $\mathcal{D}_j$, and $\bar{\boldsymbol{U}}^{(b)}(\boldsymbol{\beta})$ can be rewritten as $\bar{\boldsymbol{U}}^{(b)}(\boldsymbol{\beta}) = \sum_{j=1}^b U^{(j)}(\boldsymbol{\beta})$. To derive an online estimator upon the arrival of $\mathcal{D}_b$, a key step is to update $\bar{\boldsymbol{U}}^{(b-1)}(\boldsymbol{\beta})$ to $\bar{\boldsymbol{U}}^{(b)}(\boldsymbol{\beta})$ without re-accessing the cumulative historical raw data $\mathcal{D}_{b-1}^\star$.

To illustrate the idea, we first consider a simple case with two data batches, that is, $\mathcal{D}_2$ arrives after $\mathcal{D}_1$. The lasso estimator based on the first batch data is denoted by $\widehat{\boldsymbol{\beta}}^{(1)}$, which is the offline estimator $\bar{\boldsymbol{\beta}}^{(1)}$ that minimizes the objective function in equation (2.1). To avoid using individual-level raw data in $\mathcal{D}_1$, we approximate $U^{(1)}(\boldsymbol{\beta})$ through a first-order Taylor expansion at $\widehat{\boldsymbol{\beta}}^{(1)}$, that is,

$$U^{(1)}(\boldsymbol{\beta}) = U^{(1)}(\widehat{\boldsymbol{\beta}}^{(1)}) + \boldsymbol{J}^{(1)}(\widehat{\boldsymbol{\beta}}^{(1)})(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(1)}) + N_1 \mathcal{O}_p(\|\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(1)}\|_2^2),$$

where $\boldsymbol{J}^{(1)}(\boldsymbol{\beta}) = \partial U^{(1)}(\boldsymbol{\beta})/\partial\boldsymbol{\beta}$. It is worth noting that $\widehat{\boldsymbol{\beta}}^{(1)}$ is not the solution of $U^{(1)}(\boldsymbol{\beta}) = 0$. Nonetheless, according to the Karush-Kuhn-Tucker (KKT) conditions, $\widehat{\boldsymbol{\beta}}^{(1)}$ satisfies $\|U^{(1)}(\widehat{\boldsymbol{\beta}}^{(1)})\|_\infty = \mathcal{O}_p(\lambda_1 N_1)$, which will be much smaller than N_1 with a proper choice of λ_1. As a result, we can approximate $U^{(1)}(\boldsymbol{\beta})$ by $\boldsymbol{J}^{(1)}(\widehat{\boldsymbol{\beta}}^{(1)})(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(1)})$.

Recall that $\bar{\boldsymbol{U}}^{(2)}(\boldsymbol{\beta}) = U^{(1)}(\boldsymbol{\beta}) + U^{(2)}(\boldsymbol{\beta})$. We further propose to approximate $\bar{\boldsymbol{U}}^{(2)}(\boldsymbol{\beta})$ by

$$\widehat{\boldsymbol{U}}^{(2)}(\boldsymbol{\beta}) = \boldsymbol{J}^{(1)}(\widehat{\boldsymbol{\beta}}^{(1)})(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(1)}) + U^{(2)}(\boldsymbol{\beta}).$$

Apparently, calculating $\widehat{\boldsymbol{U}}^{(2)}(\boldsymbol{\beta})$ only requires access to the summary statistics $\{\widehat{\boldsymbol{\beta}}^{(1)}, \boldsymbol{J}^{(1)}(\widehat{\boldsymbol{\beta}}^{(1)})\}$ rather than the individual-level data in $\mathcal{D}_1$.

The above approximation could be further generalized to an arbitrary data batch $\mathcal{D}_b$. Let $\widehat{\boldsymbol{J}}^{(b-1)} = \sum_{j=1}^{b-1} \boldsymbol{J}^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)})$ denote the aggregated information matrix. Here, we evaluate each batch-specific information matrix $\boldsymbol{J}^{(j)}$ by plugging in $\widehat{\boldsymbol{\beta}}^{(j)}$ rather than $\widehat{\boldsymbol{\beta}}^{(b)}$ to avoid retrospective calculations. Then the approximation procedure becomes

$$\begin{aligned}\widehat{\boldsymbol{U}}^{(b)}(\boldsymbol{\beta}) &= \left\{\sum_{j=1}^{b-1} \boldsymbol{J}^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)})\right\}(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(b-1)}) + U^{(b)}(\boldsymbol{\beta}) \\ &= \widehat{\boldsymbol{J}}^{(b-1)}(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(b-1)}) + U^{(b)}(\boldsymbol{\beta}).\end{aligned}$$

The aggregated gradient $\widehat{\boldsymbol{U}}^{(b)}(\boldsymbol{\beta})$ depends only on $\{\widehat{\boldsymbol{\beta}}^{(b-1)}, \widehat{\boldsymbol{J}}^{(b-1)}\}$. Hence, we compute $\widehat{\boldsymbol{\beta}}^{(b)}$ through the following procedure.

- Step 1: update $\widehat{\boldsymbol{\beta}}^{(b)}$ through gradient descent with learning rate η:

$$\widehat{\boldsymbol{\beta}}^{(b)} \leftarrow \widehat{\boldsymbol{\beta}}^{(b)} + [\eta \widehat{\boldsymbol{U}}^{(b)}(\widehat{\boldsymbol{\beta}}^{(b)})/2N_b]. \tag{2.2}$$

- Step 2: apply the soft-thresholding operator $\mathcal{S}(x; \eta\lambda_b)$ to each component of $\widehat{\boldsymbol{\beta}}^{(b)}$ obtained in Step 1, where $\mathcal{S}(x; \eta\lambda_b) = sgn(x)(|x| - \eta\lambda_b)_+$ and λ_b is the regularization parameter for step b, that is,

$$\widehat{\beta}_r^{(b)} \leftarrow \mathcal{S}(\widehat{\beta}_r^{(b)}; \eta\lambda_b),\ r = 1, \ldots, p. \tag{2.3}$$

The above two steps are carried out iteratively till convergence to obtain the online lasso estimator $\widehat{\boldsymbol{\beta}}^{(b)}$. This is an online modification and extension of the iterative shrinkage-thresholding algorithm (ISTA) for the generalized linear models [6, 3]. In this algorithm, gradient descent is combined with soft-thresholding [9] to produce a sequence of sparse solutions. It is an online algorithm where the iterations proceed along with new samples rather than a fixed dataset. In the implementation, we set the stopping criterion to be $\|\eta \times \widehat{\boldsymbol{U}}^{(b)}(\widehat{\boldsymbol{\beta}}^{(b)})/2N_b\|_2 \leq 10^{-6}$. In summary, our proposed online estimator $\widehat{\boldsymbol{\beta}}^{(b)}$ can be defined as

$$\widehat{\boldsymbol{\beta}}^{(b)} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\left[\frac{1}{2N_b}\left\{\sum_{i\in\mathcal{D}_b}\{g(\boldsymbol{x}_i\boldsymbol{\beta}) - y_i\boldsymbol{x}_i\boldsymbol{\beta}\}\right.\right.$$

$$+\frac{1}{2}(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}^{(b-1)})^{\top}\widehat{\boldsymbol{J}}^{(b-1)}(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}^{(b-1)})\Big\}+\lambda_b\|\boldsymbol{\beta}\|_1\Big]. \tag{2.4}$$

In contrast to the standard offline lasso estimator in (2.1), our proposed online estimator in (2.4) depends on the data only through the summary statistics $\{\widehat{\boldsymbol{\beta}}^{(b-1)}, \widehat{\boldsymbol{J}}^{(b-1)}\}$ instead of $\mathcal{D}_{b-1}^{\star}$.

2.2. Online debiased lasso

We now proceed to study the online statistical inference and construct confidence intervals for the r-th component of the regression parameter vector, $r = 1, \ldots, p$. However, as pointed out by [33] and [4], the lasso-type estimator is not root-n consistent and does not have a tractable limiting distribution in the high-dimensional setting. An additional debiased step is needed. To do that, we define the following estimator that will be used in the low-dimensional projection:

$$\widehat{\boldsymbol{\gamma}}_r^{(b)} = \underset{\boldsymbol{\gamma}\in\mathbb{R}^{(p-1)}}{\arg\min}\left\{\frac{1}{2N_b}\left(\widehat{\boldsymbol{J}}_{r,r}^{(b)} - 2\widehat{\boldsymbol{J}}_{r,-r}^{(b)}\boldsymbol{\gamma} + \boldsymbol{\gamma}^{\top}\widehat{\boldsymbol{J}}_{-r,-r}^{(b)}\boldsymbol{\gamma}\right) + \lambda_b\|\boldsymbol{\gamma}\|_1\right\}. \tag{2.5}$$

where N_b and λ_b are the same as in (2.4). Letting $\widehat{W}^{(j)} \in \mathbb{R}^{n_j\times n_j}$ be the diagonal matrix with diagonal elements $\sqrt{g''(\boldsymbol{X}^{(j)}\widehat{\boldsymbol{\beta}}^{(j)})}$, where $g''(\cdot)$ is the second derivative of the function $g(\cdot)$, and $\widehat{\boldsymbol{X}}^{(j)}$ be the weighted design matrix $\widehat{W}^{(j)}\boldsymbol{X}^{(j)}$. Equation (2.5) can be further recast into

$$\widehat{\boldsymbol{\gamma}}_r^{(b)} = \underset{\boldsymbol{\gamma}\in\mathbb{R}^{(p-1)}}{\arg\min}\left\{\frac{1}{2N_b}\sum_{j=1}^{b}\left\|\widehat{\boldsymbol{X}}_{\cdot r}^{(j)} - \widehat{\boldsymbol{X}}_{\cdot -r}^{(j)}\boldsymbol{\gamma}\right\|_2^2 + \lambda_b\|\boldsymbol{\gamma}\|_1\right\}. \tag{2.6}$$

It is worth noting that $\widehat{\boldsymbol{\gamma}}_r^{(b)}$ can be computed in a similar way to the online lasso estimator defined in equation (2.4). Specifically, (2.6) has the same form as (2.1) if we choose the function $g(t) = t^2/2$. It implies that we can compute $\widehat{\boldsymbol{\gamma}}_r^{(b)}$ according to the procedure (2.2)–(2.3) in Section 2.1. To solve (2.6) in the online fashion as (2.2)–(2.3), the summary statistic is $(\widehat{\boldsymbol{J}}_{r,-r}^{(b)}, \widehat{\boldsymbol{J}}_{-r,-r}^{(b)})$, which has been stored as $\widehat{\boldsymbol{J}}^{(b)}$ in previous lasso estimation step. Besides that, we introduce two notations: $\widehat{\tau}_r^{(b)} = \widehat{\boldsymbol{J}}_{r,r}^{(b)} - \widehat{\boldsymbol{J}}_{r,-r}^{(b)}\widehat{\boldsymbol{\gamma}}_r^{(b)}$ and $\widetilde{\boldsymbol{\gamma}}_r^{(j)} = (\widehat{\gamma}_{r,1}^{(j)}, \ldots, -1, \ldots, \widehat{\gamma}_{r,p}^{(j)})^{\top} \in \mathbb{R}^p$, whose r-th element is -1. Then, upon the arrival of the batch data $\mathcal{D}_b$, we define the online debiased lasso estimator as

$$\widehat{\beta}_{\mathrm{on},r}^{(b)} = \widehat{\beta}_r^{(b)} - \frac{1}{\widehat{\tau}_r^{(b)}}\Bigg[\{\widetilde{\boldsymbol{\gamma}}_r^{(b)}\}^{\top}\sum_{j=1}^{b}\{\boldsymbol{X}^{(j)}\}^{\top}\left\{\boldsymbol{y}^{(j)} - g'(\boldsymbol{X}^{(j)}\widehat{\boldsymbol{\beta}}^{(j)})\right\} + \{\widetilde{\boldsymbol{\gamma}}_r^{(b)}\}^{\top}\sum_{j=1}^{b}\boldsymbol{J}^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)})\{\widehat{\boldsymbol{\beta}}^{(j)} - \widehat{\boldsymbol{\beta}}^{(b)}\}\Bigg]$$

$$\equiv \widehat{\beta}_r^{(b)} + \text{debiasing term} + \text{online error correction term}. \tag{2.7}$$

The debiased lasso estimator involves the initial lasso estimator defined in (2.4), as well as two additional terms: a debiasing term and an online error correction term. van de Geer et al. [30] studied the offline version of debiased lasso for generalized linear models. Our debiased term can be viewed as an online generalization of the offline counterpart in van de Geer et al. [30]. However, they are fundamentally different because the debiasing term in (2.7) is not sufficient to establish the asymptotic normality of $\widehat{\beta}_{\text{on,r}}^{(b)}$. As we used $\widehat{\boldsymbol{\beta}}^{(j)}$ to approximate $\widehat{\boldsymbol{\beta}}^{(b)}$, the approximation error accumulates even if each $\widehat{\boldsymbol{\beta}}^{(j)}, j = 1, \ldots, b$, is consistent to $\boldsymbol{\beta}^0$. The additional "online error correction term" in (2.7) is used to eliminate the approximation error arising from the online updates where we do not do retrospective calculations by plugging $\widehat{\boldsymbol{\beta}}^{(b)}$ into $g'(\boldsymbol{X}^{(j)}\widehat{\boldsymbol{\beta}}^{(j)})$'s for $j = 1, \ldots, b-1$.

Meanwhile, the proposed debiased lasso estimator with the online error correction term aligns with the online learning framework, as (2.7) only requires the following summary statistics rather than the entire dataset $\mathcal{D}_b^\star$:

$$\boldsymbol{M}_1^{(b)} = \sum_{j=1}^{b} \{\boldsymbol{X}^{(j)}\}^\top \left\{\boldsymbol{y}^{(j)} - g'(\boldsymbol{X}^{(j)}\widehat{\boldsymbol{\beta}}^{(j)})\right\}, \quad \boldsymbol{M}_2^{(b)} = \sum_{j=1}^{b} \boldsymbol{J}^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)})\widehat{\boldsymbol{\beta}}^{(j)}, \tag{2.8}$$

which keep the same size when new data arrive, and can be easily updated. Then, (2.7) could be written as

$$\widehat{\beta}_{\text{on},r}^{(b)} = \widehat{\beta}_r^{(b)} - \frac{\{\widetilde{\boldsymbol{\gamma}}_r^{(b)}\}^\top}{\widehat{\tau}_r^{(b)}}\{\boldsymbol{M}_1^{(b)} + \boldsymbol{M}_2^{(b)} - \widehat{\boldsymbol{J}}^{(b)}\widehat{\boldsymbol{\beta}}^{(b)}\}.$$

The asymptotic normality of the online debiased lasso and the oracle inequality of two lasso-typed estimators in (2.4) and (2.5) are established in Section 3. For variance estimation, let

$$\widehat{v}^{(b)} = \{\widetilde{\boldsymbol{\gamma}}_r^{(b)}\}^\top \widehat{\boldsymbol{J}}^{(b)}\widetilde{\boldsymbol{\gamma}}_r^{(b)}. \tag{2.9}$$

The estimated standard error $\widehat{\sigma}_r^{(b)} = \{\widehat{v}^{(b)}\}^{1/2}/\widehat{\tau}_r^{(b)}$ can also be updated online accordingly.

2.3. Practical guidance: adaptive tuning

In an offline setting, the regularization parameter λ is typically determined by cross-validation where the entire dataset is split into training and test sets multiple times. However, since the full dataset is not accessible in an online setting, such a procedure is not feasible. To align with the nature of streaming datasets, we use the "rolling-original-recalibration" procedure with the mean squared prediction error (MSPE) as the cross-validation criterion [14], as practical guidance for choosing tuning parameters. Specifically, we define a sequence of candidate

sets, that is, $T_{\lambda,1}$ and $T_{\lambda,j} = \{a/\sqrt{N_j} : a \in T_{\lambda,1}\}, j = 2, \ldots, b$. The construction of $T_{\lambda,j}$ aligns with the rate $\mathcal{O}(\sqrt{\log(p)/N_j})$ of tuning parameter in Theorem 1. At time point b, the cumulative dataset up to time point $b-1$ serves as the training set while the new data batch $\mathcal{D}_b$ is the test set. It is worth noting that for every $\lambda \in T_{\lambda,j}$ where $T_{\lambda,j}$ is the candidate set, we update the lasso estimates $\widehat{\boldsymbol{\beta}}^{(j)}(\lambda)$ along $j = 1, \ldots, b-1$ and save the most recent one, denoted by $\widehat{\boldsymbol{\beta}}^{(b-1)}(\lambda)$. Therefore, as we proceed to step b, instead of re-accessing raw data $\{\mathcal{D}_1, \ldots, \mathcal{D}_{b-1}\}$, we plug in $\widehat{\boldsymbol{\beta}}^{(b-1)}(\lambda)$ to evaluate the MSPE defined below:

$$\text{MSPE}_b(\lambda) = n_b^{-1} \left\| \boldsymbol{y}^{(b)} - g'\left(\boldsymbol{X}^{(b)} \widehat{\boldsymbol{\beta}}^{(b-1)}(\lambda)\right) \right\|_2^2, \ \lambda \in T_{\lambda,b}, \tag{2.10}$$

and choose λ such that $\lambda_b = \arg\min_{\lambda \in T_{\lambda,b}} \text{MSPE}_b(\lambda)$. In calculating $\text{MSPE}_b(\lambda)$, we plug in $\widehat{\boldsymbol{\beta}}^{(b-1)}$ rather than $\widehat{\boldsymbol{\beta}}^{(b)}$ because the latter involves the test set $\mathcal{D}_b$ and may lead to an issue of over-fitting. The initial λ_1 is selected by the classical offline cross-validation.

2.4. Algorithm

We present the procedure discussed in Sections 2.1–2.3 in Figure 1 and Algorithm 1. It consists of two main blocks: one is *online lasso estimation* and the other is *online low-dimensional projection*. Outputs from both blocks are used to compute the online debiased lasso estimator as well as the construction of confidence intervals in real-time. In particular, when a new data batch $\mathcal{D}_b$ arrives, it is first sent to the online lasso estimation block, where the summary statistics $\{\widehat{\boldsymbol{\beta}}^{(b-1)}, \widehat{\boldsymbol{J}}^{(b-1)}\}$ are used to compute $\widehat{\boldsymbol{U}}^{(b)}$. Then we use gradient descent to update the lasso estimator $\widehat{\boldsymbol{\beta}}^{(b-1)}$ to $\widehat{\boldsymbol{\beta}}^{(b)}$ at a sequence of tuning parameter values without retrieving the whole dataset. At the same time, regarding the cumulative dataset that produces the old lasso estimate $\widehat{\boldsymbol{\beta}}^{(b-1)}$ as training set and the newly arrived $\mathcal{D}_b$ as test set, we can choose the tuning parameter λ_b that gives the smallest prediction error. Now, the selected λ_b and sub-matrices of $\widehat{\boldsymbol{J}}^{(b)}$ are passed to the low-dimensional projection block for the calculation of $\widehat{\boldsymbol{\gamma}}_r^{(b)}(\lambda_b)$. The resulting projection $\widehat{\tau}_r^{(b)}$ from the low-dimensional projection block together with the lasso estimator $\widehat{\boldsymbol{\beta}}^{(b)}$ will be used to compute the debiased lasso estimator $\widehat{\beta}_{\text{on},r}^{(b)}$ and its estimated standard error $\widehat{\sigma}_r^{(b)}$.

Algorithm 1. *Online debiased lasso algorithm in generalized linear models.*

For $b = 1, 2, \ldots$
Receive the streaming dataset $\mathcal{D}_b$*;*
For a sequence of $\lambda \in T_{\lambda,b}$*, update online lasso estimator* $\widehat{\boldsymbol{\beta}}^{(b)}(\lambda)$ *defined in* (2.4)
Determine λ_b *from* $\lambda_b = \arg\min_{\lambda \in T_{\lambda,b}} MSPE_b(\lambda)$ *defined in* (2.10)
Update and store the summary statistics $\{\widehat{\boldsymbol{\beta}}^{(b)}, \widehat{\boldsymbol{J}}^{(b)}\}$
Given λ_b*, update the estimator in low-dimensional projection* $\widehat{\boldsymbol{\gamma}}_r^{(b)}$ *defined in* (2.5)
Update and store the summary statistics $\boldsymbol{M}_1^{(b)}$ *and* $\boldsymbol{M}_2^{(b)}$ *by* (2.8)
Compute $\widehat{\beta}_{on,r}^{(b)}$ *by* (2.7) *and* $\widehat{\sigma}_r^{(b)}$
Output $\widehat{\beta}_{on,r}^{(b)}$ *and its estimated standard error* $\widehat{\sigma}_r^{(b)}$

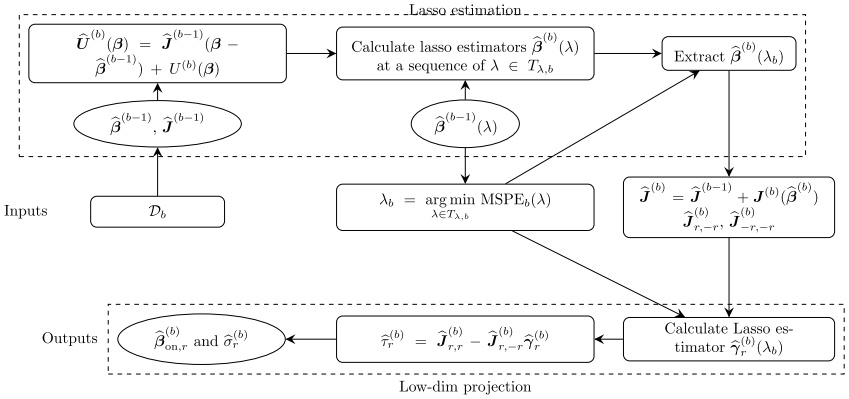

FIG 1. *Flowchart of the online debiasing algorithm. When a new data batch $\mathcal{D}_b$ arrives, it is sent to the lasso estimation block for updating $\widehat{\boldsymbol{\beta}}^{(b-1)}$ to $\widehat{\boldsymbol{\beta}}^{(b)}$. At the same time, it is also viewed as a test set for adaptively choosing tuning parameter λ_b. In the low-dim projection block, we extract sub-matrices from the updated information matrix $\widehat{\boldsymbol{J}}^{(b)}$ to compute $\widehat{\boldsymbol{\gamma}}^{(b)}_r(\lambda_b)$ and the corresponding low-dimensional projection $\widehat{\tau}^{(b)}_r$. Outputs $\widehat{\beta}^{(b)}_r(\lambda_b)$, and $\widehat{\tau}^{(b)}_r$ are further used to compute the debiased lasso estimator $\widehat{\beta}^{(b)}_{on,r}$ and its estimated standard error $\widehat{\sigma}^{(b)}_r$.*

Remark 1. When p is large, it may be challenging to implement the online debiasing algorithm since the space complexity to store the aggregated information matrix $\widehat{\boldsymbol{J}}^{(b)}$ is $\mathcal{O}(p^2)$. To reduce memory usage, we can compute the eigenvalue decomposition (EVD) of $\widehat{\boldsymbol{J}}^{(b)} = Q_b \Lambda_b Q_b^\top$, where Q_b is the $p \times N_b$ columns orthogonal matrix of the eigenvectors, Λ_b is the $N_b \times N_b$ diagonal matrix whose diagonal elements are the eigenvalues of $\widehat{\boldsymbol{J}}^{(b)}$. We only need to store Q_b and Λ_b. Since $r_b = \mathrm{rank}(\Lambda_b) \leq \min\{N_b, p\}$, we can use an incremental EVD approach [5] to update Q_b and Λ_b. Then the space complexity reduces to $\mathcal{O}(r_b p)$. The space complexity can be further reduced by setting a threshold. For example, select the principal components which explain most of the variations in the predictors. However, incremental EVD could increase the computational cost since it requires additional $\mathcal{O}(r_b^2 p)$ computational complexity. Indeed, there is a trade-off between space complexity and computational complexity. How to balance this trade-off is an important computational issue and deserves careful analysis, but is beyond the scope of this study.

3. Theoretical properties

In this section, we state our main theoretical results: the oracle inequality of lasso estimators $\widehat{\boldsymbol{\beta}}^{(b)}$ and $\widehat{\boldsymbol{\gamma}}^{(b)}_r$ defined in (2.4) and (2.5) respectively, as well as the asymptotic normality of the online debiased estimator $\widehat{\beta}^{(b)}_{\mathrm{on},r}$. Recall that $\boldsymbol{\beta}^0$ is the underlying true coefficient. Consider a random design matrix $\boldsymbol{X}$ with i.i.d rows. Let $\boldsymbol{\Sigma} = \mathbb{E}[(\boldsymbol{X}^{(1)})^\top \boldsymbol{X}^{(1)}]/N_1$, $\boldsymbol{J} = \mathbb{E}[\boldsymbol{J}^{(1)}(\boldsymbol{\beta}^0)]/N_1$ and $\boldsymbol{\Theta} = \boldsymbol{J}^{-1}$ be its

inverse. Then, the ground truth of $\widehat{\boldsymbol{\gamma}}_r^{(b)}$ given in (2.5) is defined as

$$\boldsymbol{\gamma}_r^0 = \arg\min_{\boldsymbol{\gamma}\in\mathbb{R}^{(p-1)}} \mathbb{E}\left(\boldsymbol{J}_{r,r} - 2\boldsymbol{J}_{r,-r}\boldsymbol{\gamma} + \boldsymbol{\gamma}^\top \boldsymbol{J}_{-r,-r}\boldsymbol{\gamma}\right).$$

In addition, we let $S_0 = \{r : \beta_r^0 \neq 0\}$, $s_0 = |S_0|$, $S_r = \{k \neq r : \boldsymbol{\Theta}_{k,r} \neq 0\}$ and $s_r = |S_r|$ for $r = 1, \ldots, p$. The following assumptions are needed to build the oracle inequality of the lasso estimators $\widehat{\boldsymbol{\beta}}^{(b)}$ and $\widehat{\boldsymbol{\gamma}}_r^{(b)}$ defined in (2.4) and (2.5).

Assumption 1. *Suppose that*

(A1) The pairs of random variables $\{y_i, \boldsymbol{x}_i\}_{i\in\mathcal{D}_b^\star}$ *are i.i.d.. The covariates are bounded by some constant* $K > 0$*, that is,* $\sup_{i\in\mathcal{D}_b^\star} \|\boldsymbol{x}_i\|_\infty \leq K$ *with probability one.*

(A2) $\sup_{i\in\mathcal{D}_b^\star} |\boldsymbol{x}_i\boldsymbol{\beta}^0| = \mathcal{O}(1)$ *and* $\sup_{i\in\mathcal{D}_b^\star} |(\boldsymbol{x}_i)_{-r}\boldsymbol{\gamma}_r^0| = \mathcal{O}(K)$*, where* $(\boldsymbol{x}_i)_{-r}$ *is the sub-vector of* $\boldsymbol{x}_i$ *with* r*-th element deleted. In addition,* $\sup_{i\in\mathcal{D}_b^\star} |1/g''(\boldsymbol{x}_i\boldsymbol{\beta}^0)| = \mathcal{O}(1)$.

(A3) For some δ*-neighborhood* ($\delta > 0$)*,* $g''(\cdot)$ *is Lipschitz with constant* l_g*, that is,*

$$\sup_{i\in\mathcal{D}_b^\star} \sup_{u,v\in\{v:|v-\boldsymbol{x}_i\boldsymbol{\beta}^0|\leq\delta\}} \frac{|g''(u) - g''(v)|}{|u - v|} \leq l_g.$$

(A4) The smallest eigenvalue of $\boldsymbol{J}$ *is bounded away from zero. In addition,* $0 \leq c_1 \leq \Lambda_{\min}(\boldsymbol{\Sigma}) \leq \Lambda_{\max}(\boldsymbol{\Sigma}) \leq c_2 < \infty$ *for two absolute constants* c_1 *and* c_2.

(A1) assumes that the streaming data is homogeneous, and the covariates follow bounded distributions for some finite K or the covariates are sub-gaussian random variables with $K = \sqrt{\log\{\max(N_b, p)\}}$. (A2) requires the boundness of $|\boldsymbol{x}_i\boldsymbol{\beta}^0|$ and $|(\boldsymbol{x}_i)_{-r}\boldsymbol{\gamma}_r^0|$. Such an assumption is regular for high-dimensional models [30, 2]. (A3) requires the Lipschitz property of the derivative of the mean function around the truth value. It can be easily verified that the popular logistic regression, a special case of generalized linear models, satisfies (A3). (A4) ensures that the compatibility condition [29] holds.

Theorem 1. *Assume Assumption 1 holds. Suppose that the first batch size* $n_1 \geq cK^2 s_0^2 \log p$ *for some constant* c *and* $b = o(\log N_b)$*, and the tuning parameter* $\lambda_j = C\{\log(p)/N_j\}^{1/2}, j = 1, \ldots b$ *for some constant* C*. If there exists an* $\epsilon > 0$ *such that* $K^2 s_0^2 \log(p) N_b^{-1+\epsilon} = o(1)$*, then, for any* $j = 1, \ldots, b$*, with probability at least* $1 - p^{-2}$*, the proposed online estimator in* (2.4) *satisfies*

$$\|\widehat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^0\|_1 \leq c_1^{(j)} s_0 \lambda_j, \ \ \|\boldsymbol{X}_\star^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^0)\|_2^2 \leq c_2^{(j)} s_0 N_j \lambda_j^2.$$

Remark 2. Theorem 1 provides upper bounds of the estimation error and the prediction error of the online lasso estimator $\widehat{\boldsymbol{\beta}}^{(j)}$. The constants $c_1^{(j)}$ and $c_2^{(j)}$, $j = 1, \ldots, b$, possibly depend on the batch step j. Since the lasso estimator $\widehat{\boldsymbol{\beta}}^{(b)}$ depends on $\widehat{\boldsymbol{\beta}}^{(b-1)}$, the estimation error in the previous step will be carried onto the updated estimators. As a result, it is inevitable that some constants in the oracle inequality depend on b; nonetheless, they are well under control as long as $b = o(\log N_b)$.

The next corollary shows the consistency of the proposed online lasso estimator in (2.4).

Corollary 1. *Assume those conditions in Theorem 1 hold. Then, the lasso estimator in (2.4) satisfies* $\|\widehat{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0\|_1 \to_p 0$ *as* $N_b \to \infty$, *where* $\to_p$ *means convergence in probability.*

We present the oracle inequality for $\widehat{\boldsymbol{\gamma}}_r^{(b)}$ in the next theorem.

Theorem 2. *Assume Assumption 1 holds. Suppose that the cumulative batch size satisfies* $N_j \geq cc_1^{(j)} K^2 s_0^2 s_r^2 \log p, j = 1, \ldots, b$ *for some constant* c, *and the tuning parameter* $\lambda_j = C\{\log(p)/N_j\}^{1/2}$ *for some constant* C, *then, for any* $j = 1, \ldots, b$, *with probability at least* $1 - p^{-2}$, *the estimator in low-dimensional projection defined in (2.5) satisfies* $\|\widehat{\boldsymbol{\gamma}}_r^{(j)} - \boldsymbol{\gamma}_r^0\|_1 \leq c_3 s_r \lambda_j$.

Combining the results in Theorem 1 and Theorem 2, we can establish the asymptotic normality of the online debiased lasso estimator.

Theorem 3. *Assume those conditions in Theorem 1 and Theorem 2 hold. If there exists an* $\epsilon > 0$ *such that*

$$s_0^2 K^3 \log(p) \log(N_b) N_b^{-\frac{1}{2}+\epsilon} = o(1), \;\; s_0^2 s_r^2 K^2 \log(p) N_b^{-1+\epsilon} = o(1),$$

then for any fixed r,

$$\frac{\widehat{\tau}_r^{(b)}}{\sqrt{N_b}}(\widehat{\beta}_{on,r}^{(b)} - \beta_r^0) = W_r + V_r,$$

$$W_r = \frac{1}{\sqrt{N_b}}\{\widetilde{\boldsymbol{\gamma}}_r^{(b)}\}^\top \sum_{j=1}^{b} \{\boldsymbol{X}^{(j)}\}^\top \left(g'(\boldsymbol{X}^{(j)}\boldsymbol{\beta}^0) - \boldsymbol{y}^{(j)}\right), \;\; V_r = o_p(1).$$

According to Theorem 3, the asymptotic expression of $\widehat{\tau}_r^{(b)}(\widehat{\beta}_{\text{on},r}^{(b)} - \beta_r^0)/\sqrt{N_b}$ is a sum of W_r and V_r, where W_r converges in distribution to a normal random variable by the martingale central limit theorem and V_r diminishes as N_b goes to infinity. Note that $\widehat{\tau}_r^{(b)} = \mathcal{O}_p(N_b)$, which implies the convergence rate of $\widehat{\beta}_{\text{on},r}^{(b)} - \beta_r^0$ is at the order of $1/\sqrt{N_b}$.

Remark 3. Theorem 3 implies that the total data size N_b could be as small as the logarithm of the dimensionality p, which is a common condition for offline debiased lasso in the literature [32, 30]. However, due to the lack of access to the whole dataset, it is increasingly difficult to derive the asymptotic property of online debiased lasso. One major difficulty arises from the dependence among $\widehat{\boldsymbol{\beta}}^{(1)}, \ldots, \widehat{\boldsymbol{\beta}}^{(b)}$. Another difficulty is dealing with the approximation error that accumulates in the online updating, especially under high-dimensional settings. In contrast, the classical offline lasso does not have these two problems. Even for the online debiased lasso in the linear model [14], the above two problems can be bypassed by making use of the special structure of the least squares in the linear model.

4. Simulation studies

4.1. Setup

In this section, we conduct simulation studies to examine the finite-sample performance of the proposed online debiased lasso. We randomly generate a total of N_b samples arriving in a sequence of b data batches, denoted by $\{\mathcal{D}_1, \ldots, \mathcal{D}_b\}$, from a logistic regression model. Specifically,

$$P(y_i^{(j)} = 1 \mid \boldsymbol{x}_i^{(j)}) = \frac{\exp(\boldsymbol{x}_i^{(j)}\boldsymbol{\beta}^0)}{1 + \exp(\boldsymbol{x}_i^{(j)}\boldsymbol{\beta}^0)}$$
$$P(y_i^{(j)} = 0 \mid \boldsymbol{x}_i^{(j)}) = 1 - P(y_i^{(j)} = 1 \mid \boldsymbol{x}_i^{(j)}),\ i = 1, \ldots, n_j;\ j = 1, \ldots, b,$$

where $\{\boldsymbol{x}_i^{(j)}\}^\top \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ and $\boldsymbol{\beta}^0 \in \mathbb{R}^p$ is a p-dimensional sparse parameter vector with $s_0 = 10$ denoting the number of nonzero components in $\boldsymbol{\beta}^0$. We set half of these nonzero coefficients to be 1 (relatively strong signals), and another half to be 0.01 (weak signals). We consider the following settings: (i) $N_b = 624$, $b = 12$, $n_j = 52$ for $j = 1, \ldots, 12$, $p = 600$; (ii) $N_b = 1{,}000$, $b = 10$, $n_j = 100$ for $j = 1, \ldots, 10$, $p = 1{,}000$; (iii) $N_b = 1{,}000$, $b = 10$, $n_j = 100$ for $j = 1, \ldots, 10$, $p = 2000$. In all settings, we choose $\boldsymbol{\Sigma} = 0.1 \times \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$ and the learning rate is set to $\eta = 0.005$.

The objective is to conduct both estimation and inference along the arrival of a sequence of data batches. The evaluation criteria include: averaged absolute bias in estimating $\boldsymbol{\beta}^0$ (A.bias); averaged estimated standard error (ASE); empirical standard error (ESE); coverage probability (CP) of the 95% confidence intervals; averaged length of the 95% confidence interval (ACL). These metrics will be evaluated separately for three groups: (i) $\beta_r^0 = 0$, (ii) $\beta_r^0 = 0.01$ and (iii) $\beta_r^0 = 1$. Comparison is made among (i) the maximum likelihood estimator obtained by fitting the conventional generalized linear model at the terminal point where $N_b \geq p$, (ii) the offline debiased ℓ_1-penalized estimator at the terminal point which is also the benchmark method (included in $p = 600$ only because of the computation burden), and (iii) our proposed online debiased lasso estimator at several intermediate points from $j = 1, \ldots, b$. Two offline methods included in comparison are executed with existing R packages `hdi` [8] and `glm`, respectively. The results are reported in Tables 1–3.

4.2. Bias and coverage probability

It can be seen from Tables 1–3 that the estimation bias of the online debiased lasso (ODL) estimator decreases rapidly as the number of data batches b increasing from 2 to 10. Both the estimated standard errors and the average length of 95% confidence intervals show a similar decreasing trend over time, and almost coincide with those by the offline benchmark method in Table 1 at the terminal points. Besides that, the coverage probability of ODL always performs well. For example, in Tables 1–3, the coverage probabilities of ODL are close to 95% across all updating points $j = 2, 4, \ldots, 10$.

TABLE 1

$N_b = 624$, $b = 12$, $p = 600$, $s_0 = 10$, $\mathbf{\Sigma} = 0.1 \times \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$. Performance on statistical inference. "MLE" is the offline estimator obtained by fitting the traditional GLM, "offline" corresponds to the offline debiased ℓ_1-norm penalized estimator, and "ODL" represents our proposed online debiased lasso estimator.

	$\beta_{0,r}$	MLE	offline	ODL					
data batch index				2	4	6	8	10	12
A.bias	0	1.594	0.018	0.042	0.034	0.031	0.029	0.027	0.025
	0.01	1.554	0.016	0.044	0.031	0.026	0.012	0.010	0.010
	1	11.85	0.077	0.192	0.168	0.159	0.147	0.137	0.134
ASE	0	2.43×10^6	0.307	0.576	0.416	0.347	0.304	0.274	0.251
	0.01	2.45×10^6	0.307	0.577	0.416	0.347	0.304	0.274	0.251
	1	2.42×10^6	0.308	0.576	0.416	0.347	0.304	0.275	0.251
ESE	0	27.18	0.300	0.576	0.411	0.341	0.298	0.269	0.246
	0.01	27.68	0.296	0.576	0.403	0.330	0.287	0.266	0.240
	1	27.49	0.291	0.559	0.395	0.326	0.295	0.257	0.240
CP	0	1.000	0.955	0.947	0.949	0.949	0.949	0.948	0.947
	0.01	1.000	0.957	0.949	0.958	0.951	0.958	0.951	0.956
	1	1.000	0.951	0.946	0.939	0.929	0.925	0.931	0.926
ACL	0	9.51×10^6	1.202	2.259	1.631	1.361	1.193	1.076	0.982
	0.01	9.59×10^6	1.203	2.261	1.632	1.361	1.193	1.076	0.983
	1	9.48×10^6	1.206	2.257	1.629	1.361	1.193	1.077	0.983
C.Time (min)		0.06	112.3	6.99					

TABLE 2

$N_b = 1{,}000$, $b = 10$, $p = 1{,}000$, $s_0 = 10$, $\mathbf{\Sigma} = 0.1 \times \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$. Performance on statistical inference. "MLE" is the offline estimator obtained by fitting the traditional GLM, and "ODL" is our proposed online debiased lasso estimator.

	$\beta_{0,r}$	MLE	ODL				
data batch index			2	4	6	8	10
A.bias	0	99.47	0.032	0.025	0.022	0.019	0.018
	0.01	151.69	0.036	0.026	0.026	0.019	0.020
	1	100.99	0.113	0.123	0.128	0.127	0.122
ASE	0	7.91×10^6	0.449	0.317	0.259	0.224	0.200
	0.01	8.36×10^6	0.450	0.317	0.259	0.224	0.201
	1	8.05×10^6	0.449	0.317	0.259	0.224	0.201
ESE	0	1692.27	0.434	0.309	0.252	0.218	0.196
	0.01	1611.81	0.435	0.317	0.258	0.225	0.198
	1	1774.24	0.413	0.293	0.242	0.214	0.191
CP	0	1.000	0.956	0.953	0.951	0.950	0.948
	0.01	1.000	0.955	0.949	0.945	0.949	0.941
	1	1.000	0.965	0.963	0.947	0.928	0.917
ACL	0	3.10×10^7	1.761	1.243	1.014	0.879	0.786
	0.01	3.28×10^7	1.765	1.244	1.015	0.879	0.786
	1	3.15×10^7	1.764	1.244	1.016	0.879	0.786

It is worth mentioning that at the terminal point in Tables 1 and 2 where the cumulative sample size N_b is equal to or slightly larger than p, we can still fit the conventional generalized linear model to obtain the MLE. It fails to provide reliable coverage probabilities due to severely large biases and estimated standard errors. In particular, the estimation bias of MLE is around hundreds times that of the offline or online debiased lasso when $p = 600$ as shown in Tables 1, and it further increases to thousands times of the online debiased estimator when $p = 1{,}000$. Furthermore, as clearly indicated by the large empirical standard errors, MLE under this setting suffers from severe instability. Such an invalid

TABLE 3
$N_b = 1000$, $b = 10$, $p = 2000$, $s_0 = 10$, $\mathbf{\Sigma} = 0.1 \times \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$. Performance on statistical inference. "ODL" represents our proposed online debiased lasso estimator.

data batch index	$\beta_{0,r}$	ODL 2	ODL 4	ODL 6	ODL 8	ODL 10
A.bias	0	0.028	0.021	0.018	0.016	0.015
	0.01	0.019	0.013	0.014	0.011	0.009
	1	0.091	0.102	0.104	0.102	0.110
ASE	0	0.449	0.317	0.259	0.224	0.200
	0.01	0.449	0.317	0.259	0.224	0.200
	1	0.449	0.317	0.258	0.224	0.200
ESE	0	0.442	0.313	0.255	0.221	0.197
	0.01	0.458	0.318	0.258	0.218	0.196
	1	0.419	0.305	0.245	0.218	0.195
CP	0	0.953	0.951	0.951	0.950	0.950
	0.01	0.947	0.951	0.947	0.949	0.958
	1	0.964	0.946	0.935	0.934	0.920
ACL	0	1.760	1.242	1.014	0.878	0.785
	0.01	1.761	1.241	1.014	0.878	0.785
	1	1.763	1.243	1.015	0.878	0.785

estimation and inference result by MLE further demonstrates the advantage of our proposed online debiased method under the high-dimensional sparse logistic regression setting with streaming datasets.

4.3. Computational efficiency

We make a computation time comparison including data loading time and algorithm execution time of different methods in Table 1 only, as the offline debiased lasso implemented with `hdi` becomes computationally prohibitive when p increases to 1000. Moreover, while achieving comparable statistical performance, the computational advantage of our proposed online debiased lasso is clear when $p = 600$, as it is almost 16 times faster than its offline counterpart.

5. Real data analysis

New technologies have made available vast quantities of digital text, recording an ever-increasing share of human interactions, communication, and culture [1, 20]. The information encoded in text serves as a rich complement to the more structured traditional data in research. For example, text from financial news, social media, and company filings can be used to predict asset price movements and study the causal impact of new information; text from advertisements and product reviews may also be utilized to study the drivers of consumer decision making [13]. One of the most prominent features of text data is that it is high-dimensional in nature due to the large volume of the word dictionary. Existing works mostly focused on point estimation without much consideration of statistical inference such as interval estimation [17]. Thus, a computationally-efficient online interval estimation method in high-dimensional setting is desired. Our proposed method can fill in this gap.

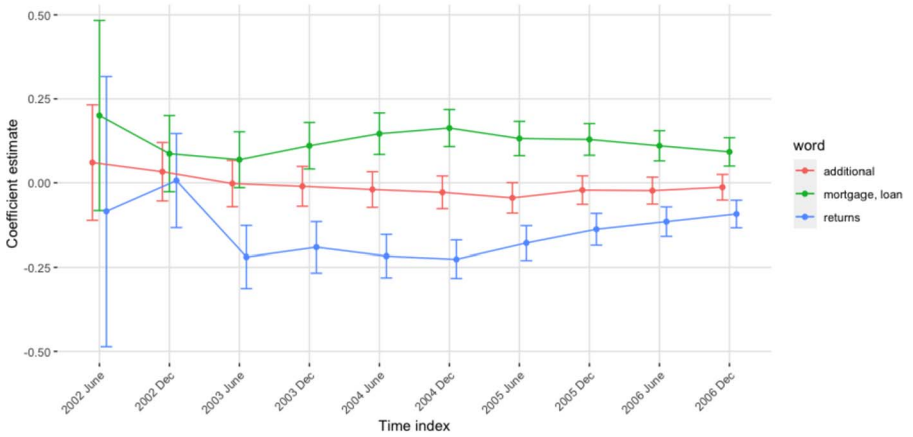

FIG 2. *Trace plot of the point estimate and 95% confidence bands of regression coefficients corresponding to phrases "additional", "mortgage loans" and "return", respectively.*

In this section, we analyze the 10-K reports dataset collected from year 2002 to year 2006 [17]. This dataset includes $n \approx 16{,}800$ annual financial reports from all publicly-traded corporations and the corresponding stock future volatilities. Our goal of this analysis is to investigate the influence of annual financial reports on stock volatility. In particular, we will fit a logistic regression model where the response variable y encodes the increase ($y = 1$) or the decrease ($y = 0$) in stock utility, and the high-dimensional covariate vector x is the top-3000 phrases (including unigrams and bigrams) with highest frequencies, i.e. $p = 3{,}000$.

We apply our proposed online debiased method to construct confidence intervals along with sequentially collected data points. We obtain an updated point estimate of the regression coefficient and its 95% confidence interval every six months, and the trajectories are plotted in Figure 2. In this analysis, we look into the associations between the frequencies of some phrases and the odds of increased stock utility. Particularly, we focus on "additional", "mortgage loan" and "return". As shown in Figure 2, all confidence intervals cover zero at the beginning, i.e. in June 2002. Later on, as data accumulates, we find that "mortgage loan" tends to show a positive association to the odds of increased stock utility while "return" shows a negative association. Furthermore, the estimated regression coefficient of "additional" stays close to zero and its confidence interval always covers zero.

These results are intuitively reasonable. First, "mortgage loan" stands for a secured loan that allows one to take funds by providing an immovable asset such as a house or commercial property. It typically happens when company faces a cash flow shortfall. Consequently, it will undermine investors' confidence and increase the volatility of the stock price. In contrast, "return" indicates a healthy development of the company, which strengthens investors' confidence and reduces the volatility. Moreover, our interval estimation can help identify

insignificant coefficients. For example, the word "additional", which is found to be positively associated with the volatility in [17], turns out to be statistically insignificant.

6. Discussion

In this paper, we propose an online debiased lasso method for statistical inference in high-dimensional generalized linear models. The method is applicable to streaming data, that is, only the historical summary statistics, not the raw historical data, are needed in updating the estimate at the current stage. Under regularity conditions similar to those in the offline setting and mild conditions on the batch sizes, we prove the online debiased lasso (with an online correction term) is asymptotically normal. The numerical studies further demonstrate the effectiveness of our algorithm and support the theoretical results.

There are several open questions in the area of online inference. First, our method are developed for homogeneous data, where the streaming data are assumed to be i.i.d. sampled. While there are some existing works that address dependence and time-varying effects in low-dimensional settings [19], it will be interesting to explore how they can be extended to high-dimensional settings. Second, the loss function we consider in this paper is the negative log-likelihood function. It is unclear whether other loss functions, including non-smooth robust loss functions such as the Huber's loss [15], could be used for online inference. Third, we did not address the issue of the online variable selection. The major difficulty in this problem is how to recover the significant variables which may be dropped at the early stages of the stream. We hope to address these interesting questions in the future.

Appendix A: Proofs of theoretical results

This section provides detailed proofs for the theorems described in the main text.

A.1. Proof of Theorem 1

Proof of Theorem 1. For the prior data batch $\mathcal{D}_1$, we have $\widehat{\boldsymbol{\beta}}^{(1)} = \bar{\boldsymbol{\beta}}^{(1)}$ where $\bar{\boldsymbol{\beta}}^{(1)}$ is the offline lasso estimator. Since the oracle inequality of $\bar{\boldsymbol{\beta}}^{(1)}$ is well-established in Section 6.7 of [4], Theorem 1 holds when $b = 1$. Now we prove the oracle inequality of $\widehat{\boldsymbol{\beta}}^{(b)}$ for an arbitrary $b \geq 2$ by the mathematical induction.

Suppose that $\widehat{\boldsymbol{\beta}}^{(b-1)}$ satisfies

$$\|\widehat{\boldsymbol{\beta}}^{(b-1)} - \boldsymbol{\beta}^0\|_1 \leq c_1^{(b-1)} s_0 \lambda_{b-1}, \quad \|\boldsymbol{X}_\star^{(b-1)}(\widehat{\boldsymbol{\beta}}^{(b-1)} - \boldsymbol{\beta}^0)\|_2^2 \leq c_2^{(b-1)} s_0 N_j \lambda_{b-1}^2$$

with constants $c_1^{(b-1)}$ and $c_2^{(b-1)}$. We claim that $\|\widehat{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0\|_1 \leq c_1^{(b)} s_0 \lambda_b$. Otherwise, we consider the following linear combination,

$$\widetilde{\boldsymbol{\beta}}^{(b)} = t\widehat{\boldsymbol{\beta}}^{(b)} + (1-t)\boldsymbol{\beta}^0, \text{ where } t = \frac{c_1^{(b)} s_0 \lambda_b}{c_1^{(b)} s_0 \lambda_b + \|\widehat{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0\|_1}. \tag{A.1}$$

Then $\|\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0\|_1 \le c_1^{(b)} s_0 \lambda_b$. Since $\|\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0\|_1 \le c_1^{(b)} s_0 \lambda_b/2$ if and only if $\|\widehat{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0\|_1 \le c_1^{(b)} s_0 \lambda_b$, it suffices to show $\|\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0\|_1 \le c_1^{(b)} s_0 \lambda_b/2$.

Let $\mathcal{L}(\boldsymbol{\beta}; \mathcal{D}_j) = \sum_{i \in \mathcal{D}_j} \{g(\boldsymbol{x}_i \boldsymbol{\beta}) - y_i \boldsymbol{x}_i \boldsymbol{\beta}\}$, $j = 1, \ldots, b$. Due to the convexity of the objective function, we have

$$\begin{aligned}
&\frac{1}{2N_b}\left\{\mathcal{L}(\widetilde{\boldsymbol{\beta}}^{(b)}; \mathcal{D}_b) + \frac{1}{2}(\widetilde{\boldsymbol{\beta}}^{(b)} - \widehat{\boldsymbol{\beta}}^{(b-1)})^\top \widehat{\boldsymbol{J}}^{(b-1)}(\widetilde{\boldsymbol{\beta}}^{(b)} - \widehat{\boldsymbol{\beta}}^{(b-1)})\right\} + \lambda_b \|\widetilde{\boldsymbol{\beta}}^{(b)}\|_1 \\
&\le \frac{1}{2N_b}\left\{\mathcal{L}(\boldsymbol{\beta}^0; \mathcal{D}_b) + \frac{1}{2}(\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}^{(b-1)})^\top \widehat{\boldsymbol{J}}^{(b-1)}(\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}^{(b-1)})\right\} + \lambda_b \|\boldsymbol{\beta}^0\|_1,
\end{aligned} \tag{A.2}$$

where $\widehat{\boldsymbol{J}}^{(b-1)} = \sum_{j=1}^{b-1} \boldsymbol{J}^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)})$. Recall that $\mathcal{D}_{b-1}^\star = \{\mathcal{D}_1, \ldots, \mathcal{D}_{b-1}\}$. A Taylor's expansion gives that

$$\begin{aligned}
&\mathcal{L}(\widetilde{\boldsymbol{\beta}}^{(b)}; \mathcal{D}_{b-1}^\star) - \mathcal{L}(\boldsymbol{\beta}^0; \mathcal{D}_{b-1}^\star) \\
&= \{\bar{\boldsymbol{U}}^{(b-1)}(\boldsymbol{\beta}^0)\}^\top (\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0) + \frac{1}{2}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)^\top \left\{\widehat{\boldsymbol{J}}^{(b-1)}(\xi)\right\}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0),
\end{aligned}$$

where $\widehat{\boldsymbol{J}}^{(b-1)}(\xi) = \sum_{j=1}^{b-1} \boldsymbol{J}^{(j)}(\xi)$ and $\xi = t_2 \boldsymbol{\beta}^0 + (1 - t_2)\widetilde{\boldsymbol{\beta}}^{(b)}$ for some $0 < t_2 < 1$. Then

$$\begin{aligned}
&\frac{1}{2}(\widetilde{\boldsymbol{\beta}}^{(b)} - \widehat{\boldsymbol{\beta}}^{(b-1)})^\top \widehat{\boldsymbol{J}}^{(b-1)}(\widetilde{\boldsymbol{\beta}}^{(b)} - \widehat{\boldsymbol{\beta}}^{(b-1)}) - \frac{1}{2}(\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}^{(b-1)})^\top \widehat{\boldsymbol{J}}^{(b-1)}(\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}^{(b-1)}) \\
&= \frac{1}{2}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)^\top \widehat{\boldsymbol{J}}^{(b-1)}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0) - (\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)^\top \widehat{\boldsymbol{J}}^{(b-1)}(\widehat{\boldsymbol{\beta}}^{(b-1)} - \boldsymbol{\beta}^0) \\
&= \mathcal{L}(\widetilde{\boldsymbol{\beta}}^{(b)}; \mathcal{D}_{b-1}^\star) - \mathcal{L}(\boldsymbol{\beta}^0; \mathcal{D}_{b-1}^\star) \\
&\quad + \frac{1}{2}\sum_{j=1}^{b-1}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)^\top \left\{\boldsymbol{J}^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)}) - \boldsymbol{J}^{(j)}(\xi)\right\}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0) \\
&\quad - \sum_{j=1}^{b-1}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)^\top \left\{\boldsymbol{J}^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)})\right\}(\widehat{\boldsymbol{\beta}}^{(b-1)} - \boldsymbol{\beta}^0) - \{\bar{\boldsymbol{U}}^{(b-1)}(\boldsymbol{\beta}^0)\}^\top (\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0) \\
&:= \mathcal{L}(\widetilde{\boldsymbol{\beta}}^{(b)}; \mathcal{D}_{b-1}^\star) - \mathcal{L}(\boldsymbol{\beta}^0; \mathcal{D}_{b-1}^\star) + \Delta_1^{(b)} - \Delta_2^{(b)} - \Delta_3^{(b)}.
\end{aligned}$$

Substituting the above equation into (A.2), we have

$$\begin{aligned}
&\frac{1}{2N_b}\mathcal{L}(\widetilde{\boldsymbol{\beta}}^{(b)}; \mathcal{D}_b^\star) + \lambda_b \|\widetilde{\boldsymbol{\beta}}^{(b)}\|_1 + \frac{1}{2N_b}(\Delta_1^{(b)} - \Delta_2^{(b)} - \Delta_3^{(b)}) \\
&\le \frac{1}{2N_b}\mathcal{L}(\boldsymbol{\beta}^0; \mathcal{D}_b^\star) + \lambda_b \|\boldsymbol{\beta}^0\|_1.
\end{aligned} \tag{A.3}$$

The remaining part is the same as the proof of Theorem 6.4 in van de Geer [29]. Define

$$\begin{aligned}
\mathcal{E}(\boldsymbol{\beta}) &:= \frac{1}{2N_b}\mathbb{E}\left\{\mathcal{L}(\boldsymbol{\beta}; \mathcal{D}_b^\star) - \mathcal{L}(\boldsymbol{\beta}^0; \mathcal{D}_b^\star)\right\}, \\
v(\boldsymbol{\beta}; \mathcal{D}_b^\star) &:= \mathcal{L}(\boldsymbol{\beta}; \mathcal{D}_b^\star) - \mathbb{E}\left\{\mathcal{L}(\boldsymbol{\beta}; \mathcal{D}_b^\star)\right\}, \ \boldsymbol{\beta} \in \mathbb{R}^p,
\end{aligned}$$

which are the excess risk and the empirical process respectively. Note that $\mathcal{E}(\boldsymbol{\beta})$ does not depend on $\mathcal{D}_b^\star$ since the data are *i.i.d.* samples. Then, (A.3) could be further written as

$$\begin{aligned}
&\mathcal{E}(\widetilde{\boldsymbol{\beta}}^{(b)}) + \lambda_b \|\widetilde{\boldsymbol{\beta}}^{(b)}\|_1 \\
\le& -\frac{1}{2N_b}\left\{v(\widetilde{\boldsymbol{\beta}}^{(b)}; \mathcal{D}_b^\star) - v(\boldsymbol{\beta}^0; \mathcal{D}_b^\star)\right\} + \lambda_b \|\boldsymbol{\beta}^0\|_1 + \frac{1}{2N_b}(\Delta_1^{(b)} - \Delta_2^{(b)} - \Delta_3^{(b)}) \\
=& -\frac{1}{2N_b}\Delta_4^{(b)} + \lambda_b \|\boldsymbol{\beta}^0\|_1 + \frac{1}{2N_b}(\Delta_1^{(b)} - \Delta_2^{(b)} - \Delta_3^{(b)}).
\end{aligned}$$

Recall that $\boldsymbol{X}_\star^{(b-1)} = ((\boldsymbol{X}^{(1)})^\top, \ldots, (\boldsymbol{X}^{(b-1)})^\top)^\top \in \mathbb{R}^{N_{b-1} \times p}$. The next lemma provides the upper bound of $|\Delta_i^{(b)}|, i = 1, 2, 3, 4$, whose proof is given at the end of Appendix.

Lemma 1. *Under the conditions of Theorem 1, with probability at least* $1 - p^{-3}$,

$$\begin{aligned}
&|\Delta_1^{(b)}| \le 2K l_g c_1^{(1)} s_0 \lambda_1 \|\boldsymbol{X}_\star^{(b-1)}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)\|_2^2, \\
&|\Delta_2^{(b)}| \le K_2 \|\boldsymbol{X}_\star^{(b-1)}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)\|_2 \|\boldsymbol{X}_\star^{(b-1)}(\widehat{\boldsymbol{\beta}}^{(b-1)} - \boldsymbol{\beta}^0)\|_2, \\
&|\Delta_3^{(b)}| \le \lambda_{b-1} N_{b-1} \|\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0\|_1/8, \ |\Delta_4^{(b)}| \le \lambda_b N_b \|\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0\|_1/8,
\end{aligned}$$

where l_g *is Lipschitz constant defined in Assumption 1,* $K = \sup_{i \in \mathcal{D}_b^\star} \|\boldsymbol{x}_i\|_\infty$ *and* $K_2 = \sup_{i \in \mathcal{D}_b^\star} |g''(\boldsymbol{x}_i \boldsymbol{\beta})|$.

The upper bound of $|\Delta_1|$ could be absorbed in the upper bound of $|\Delta_2|$. According to Lemma 1,

$$\begin{aligned}
&\Delta_1^{(b)} - \Delta_2^{(b)} - \Delta_3^{(b)} - \Delta_4^{(b)} \\
\le& 2K_2 \|\boldsymbol{X}_\star^{(b-1)}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)\|_2 \|\boldsymbol{X}_\star^{(b-1)}(\widehat{\boldsymbol{\beta}}^{(b-1)} - \boldsymbol{\beta}^0)\|_2 + \frac{1}{4}\lambda_b N_b \|\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0\|_1 \\
\le& 2K_2 (N_{b-1} c_2^{(b-1)} s_0)^{1/2} \lambda_{b-1} \|\boldsymbol{X}_\star^{(b)}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)\|_2 + \frac{1}{4} c_1^{(b)} N_b s_0 \lambda_b^2.
\end{aligned}$$

Consequently,

$$\begin{aligned}
&\mathcal{E}(\widetilde{\boldsymbol{\beta}}^{(b)}) + \lambda_b \|\widetilde{\boldsymbol{\beta}}^{(b)}\|_1 - \lambda_b \|\boldsymbol{\beta}^0\|_1 \\
\le& \frac{1}{2N_b}\left\{2K_2 (N_{b-1} c_2^{(b-1)} s_0)^{1/2} \lambda_{b-1} \|\boldsymbol{X}_\star^{(b)}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)\|_2 + \frac{1}{4} c_1^{(b)} N_b s_0 \lambda_b^2\right\} \\
=& \left\{K_2 \left(\frac{c_2^{(b-1)} s_0}{N_b}\right)^{1/2} \|\boldsymbol{X}_\star^{(b)}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)\|_2 + \frac{1}{8} c_1^{(b)} s_0 \lambda_b\right\} \lambda_b = \Delta^{(b)} \lambda_b. \qquad \text{(A.4)}
\end{aligned}$$

Recall that $S_0 = \{j : \beta_j^0 \neq 0\}$. For $\boldsymbol{\beta} \in \mathbb{R}^p$, define $\boldsymbol{\beta}_{S_0} = (\beta_{j,S_0})_{j=1}^p$ where $\beta_{j,S_0} = \beta_j I_{\{j \in S_0\}}$. Then, $\boldsymbol{\beta} = \boldsymbol{\beta}_{S_0} + \boldsymbol{\beta}_{S_0^c}$. It follows from (A.4) that

$$\mathcal{E}(\widetilde{\boldsymbol{\beta}}^{(b)}) + \lambda_b \|\widetilde{\boldsymbol{\beta}}_{S_0^c}^{(b)}\|_1 \le \lambda_b \|\boldsymbol{\beta}^0\|_1 - \lambda_b \|\widetilde{\boldsymbol{\beta}}_{S_0}^{(b)}\|_1 + \Delta^{(b)} \lambda_b \le \lambda_b \|\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}_{S_0}^{(b)}\|_1 + \Delta^{(b)} \lambda_b.$$

Here, some discussions on the value of $\|\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}_{S_0}^{(b)}\|_1$ are needed.

Case I. Suppose that $\|\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)}_{S_0}\|_1 \geq \Delta^{(b)}/2$. Then,

$$\mathcal{E}(\widetilde{\boldsymbol{\beta}}^{(b)}) + \lambda_b \|\widetilde{\boldsymbol{\beta}}^{(b)}_{S_0^c}\|_1 \leq 3\lambda_b \|\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)}_{S_0}\|_1,$$

impling $\|\widetilde{\boldsymbol{\beta}}^{(b)}_{S_0^c}\|_1 \leq 3\|\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)}_{S_0}\|_1$. Then, we can adopt the empirical compatibility condition, that is,

$$\|\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)}_{S_0}\|_1^2 \leq \frac{s_0}{\phi_0^2 N_b} \|\boldsymbol{X}^{(b)}_\star (\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)})\|_2^2,$$

where $\phi_0 > 0$ is the compatibility constant. Thus,

$$\begin{aligned}
&\mathcal{E}(\widetilde{\boldsymbol{\beta}}^{(b)}) + \lambda_b \|\widetilde{\boldsymbol{\beta}}^{(b)}_{S_0^c}\|_1 + \lambda_b \|\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)}_{S_0}\|_1 \\
&\leq \frac{2\lambda_b}{\phi_0} \left(\frac{s_0}{N_b}\right)^{1/2} \|\boldsymbol{X}^{(b)}_\star (\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)})\|_2 + \Delta^{(b)} \lambda_b \\
&= \left(\frac{2}{\phi_0} + K_2 \sqrt{c_2^{(b-1)}}\right) \lambda_b \left(\frac{s_0}{N_b}\right)^{1/2} \|\boldsymbol{X}^{(b)}_\star (\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)})\|_2 + \frac{1}{8} c_1^{(b)} s_0 \lambda_b^2 \\
&= C\lambda_b \left(\frac{s_0}{N_b}\right)^{1/2} \|\boldsymbol{X}^{(b)}_\star (\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)})\|_2 + \frac{1}{8} c_1^{(b)} s_0 \lambda_b^2.
\end{aligned}$$

Let $k_2 = 1/\sup_{i \in \mathcal{D}_b^\star} |4/g''(\boldsymbol{x}_i \boldsymbol{\beta})|$. Based on Lemma 1, we have

$$\frac{k_2}{N_b} \|\boldsymbol{X}^{(b)}_\star (\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)})\|_2^2 \leq \mathcal{E}(\widetilde{\boldsymbol{\beta}}^{(b)}) + \frac{1}{2N_b} \Delta_4^{(b)} \leq \mathcal{E}(\widetilde{\boldsymbol{\beta}}^{(b)}) + \frac{1}{16} c_1^{(b)} s_0 \lambda_b^2,$$

which is also known as the margin condition [29]. Next, we apply the arithmetic mean-geometric mean inequality and obtain

$$C\lambda_b \left(\frac{s_0}{N_b}\right)^{1/2} \|\boldsymbol{X}^{(b)}_\star (\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)})\|_2 \leq \frac{C^2}{2k_2} s_0 \lambda_b^2 + \frac{\mathcal{E}(\widetilde{\boldsymbol{\beta}}^{(b)})}{2} + \frac{1}{32} c_1^{(b)} s_0 \lambda_b^2. \quad \text{(A.5)}$$

Then, it follows that

$$\frac{\mathcal{E}(\widetilde{\boldsymbol{\beta}}^{(b)})}{2} + \lambda_b \|\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)}\|_1 \leq \left(\frac{C^2}{2k_2} + \frac{5}{32} c_1^{(b)}\right) s_0 \lambda_b^2. \quad \text{(A.6)}$$

On the one hand, since $\mathcal{E}(\widetilde{\boldsymbol{\beta}}^{(b)}) > 0$,

$$\|\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)}\|_1 \leq \left(\frac{C^2}{2k_2} + \frac{5}{32} c_1^{(b)}\right) s_0 \lambda_b.$$

With suitable choice of $c_2^{(b-1)}$,

$$\frac{C^2}{2k_2} + \frac{5}{32} c_1^{(b)} = \frac{1}{2k_2} \left(\frac{2}{\phi_0} + K_2 \sqrt{c_2^{(b-1)}}\right)^2 + \frac{5}{32} c_1^{(b)} \leq \frac{13}{32} c_1^{(b)},$$

we have

$$\|\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)}\|_1 \le \frac{1}{2} c_1^{(b)} s_0 \lambda_b.$$

Here we require that $K_2\sqrt{c_2^{(1)}} \ge 2/\phi_0$, and $8K_2^2 c_2^{(b-1)}/k_2 = c_1^{(b)}$. On the other hand, combining (A.5) and (A.6), we obtain

$$\|\boldsymbol{X}_\star^{(b)}(\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)})\|_2 \le \frac{5C}{2k_2}(s_0 N_b)^{1/2}\lambda_b.$$

Again, since $C \le 2K_2\sqrt{c_2^{(b-1)}}$, we obtain

$$\|\boldsymbol{X}_\star^{(b)}(\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)})\|_2^2 \le \left(\frac{5K_2}{k_2}\right)^2 c_2^{(b-1)} s_0 \lambda_b^2 N_b \le c_2^{(b)} s_0 \lambda_b^2 N_b. \tag{A.7}$$

Case II. Suppose that $\|\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)}_{S_0}\|_1 < \Delta^{(b)}/2$. Then,

$$\begin{aligned}&\mathcal{E}(\widetilde{\boldsymbol{\beta}}^{(b)}) + \lambda_b\|\widetilde{\boldsymbol{\beta}}^{(b)}_{S_0^c}\|_1 + \lambda_b\|\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)}_{S_0}\|_1 \le 2\Delta^{(b)}\lambda_b \\ &= 2K_2\lambda_b\left(\frac{c_2^{(b-1)}s_0}{N_b}\right)^{1/2}\|\boldsymbol{X}_\star^{(b)}(\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)})\|_2 + \frac{1}{4}c_1^{(b)}s_0\lambda_b^2.\end{aligned}$$

Then, by the margin condition, it is straightforward to show that

$$\|\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)}\|_1 \le \frac{1}{2}c_1^{(b)}s_0\lambda_b,\ \|\boldsymbol{X}_\star^{(b)}(\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)})\|_2^2 \le \frac{36K_2^2}{k_2^2}c_2^{(b-1)}s_0\lambda_b^2 N_b = c_2^{(b)}s_0\lambda_b^2 N_b. \tag{A.8}$$

In summary, we obtain $\|\boldsymbol{\beta}^0 - \widetilde{\boldsymbol{\beta}}^{(b)}\|_1 \le c_1^{(b)}s_0\lambda_b/2$ in both cases. According to (A.1), we have shown that $\|\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}^{(b)}\|_1 \le c_1^{(b)}s_0\lambda_b$. The remaining step is to repeat the above arguments and obtain the upper bound of the estimation error in (A.7) or (A.8). It is worth pointing out that $c_1^{(b)} = 8K_2^2 c_2^{(b-1)}/k_2$ and $c_2^{(b)} = 36K_2^2 c_2^{(b-1)}/k_2^2$, where K_2 and k_2 do not depend on b. The proof is completed by taking a union bound on the events considered in Lemma 1. □

A.2. Proof of Theorem 2

Before proving Theorem 2, we state Lemma 3 in [14] to compute the cumulative error.

Lemma 2 (Lemma 3 in [14]). *Let n_j and N_j be the batch size and the cumulative batch size respectively when the j-th data arrives, $j = 1, \ldots, b$. Then,*

$$\sum_{j=1}^{b}\frac{n_j}{N_j} \le 1 + \log\frac{N_b}{n_1}, \tag{A.9}$$

$$\sum_{j=1}^{b}\frac{n_j}{\sqrt{N_j}} \le 2\sqrt{N_b}. \tag{A.10}$$

Proof of Theorem 2. Recall that

$$\widehat{\boldsymbol{\gamma}}_r^{(j)} = \underset{\boldsymbol{\gamma}\in\mathbb{R}^{(p-1)}}{\arg\min} \left\{ \frac{1}{2N_j} \left(\widehat{\boldsymbol{J}}_{r,r}^{(j)} - 2\widehat{\boldsymbol{J}}_{r,-r}^{(j)}\boldsymbol{\gamma} + \boldsymbol{\gamma}^\top \widehat{\boldsymbol{J}}_{-r,-r}^{(j)}\boldsymbol{\gamma} \right) + \lambda_j \|\boldsymbol{\gamma}\|_1 \right\}$$

is a standard lasso estimator. The proof follows the standard argument as long as estimated information matrix $\widehat{\boldsymbol{J}}^{(j)}/N_j$ satisfies the compatibility condition. Therefore, it suffices to demonstrate the compatibility condition holds. Recall that $\widehat{\boldsymbol{J}}^{(j)}/N_j = \sum_{i=1}^{j} \boldsymbol{J}^{(i)}(\widehat{\boldsymbol{\beta}}^{(i)})/N_j$. Then we have

$$\begin{aligned}
&\widehat{\boldsymbol{J}}^{(j)}/N_j - \boldsymbol{J}^0 = \frac{1}{N_j}\sum_{i=1}^{j} \left\{ \boldsymbol{J}^{(i)}(\widehat{\boldsymbol{\beta}}^{(i)}) - \mathbb{E}[\boldsymbol{J}^{(i)}(\boldsymbol{\beta}^0)] \right\} \\
&= \frac{1}{N_j}\sum_{i=1}^{j} \left\{ \boldsymbol{J}^{(i)}(\widehat{\boldsymbol{\beta}}^{(i)}) - \boldsymbol{J}^{(i)}(\boldsymbol{\beta}^0) \right\} + \frac{1}{N_j}\sum_{i=1}^{j} \left[\boldsymbol{J}^{(i)}(\boldsymbol{\beta}^0) - \mathbb{E}\{\boldsymbol{J}^{(i)}(\boldsymbol{\beta}^0)\} \right].
\end{aligned}$$

According to the error bound of $\widehat{\boldsymbol{\beta}}^{(i)}$ provided in Theorem 1 and (A.10), we obtain

$$\begin{aligned}
&\left\| \frac{1}{N_j}\sum_{i=1}^{j} \left\{ \boldsymbol{J}^{(i)}(\widehat{\boldsymbol{\beta}}^{(i)}) - \boldsymbol{J}^{(i)}(\boldsymbol{\beta}^0) \right\} \right\|_\infty \le \frac{K^2}{N_j}\sum_{i=1}^{j} n_i \|\widehat{\boldsymbol{\beta}}^{(i)} - \boldsymbol{\beta}^0\|_1 \\
&= \mathcal{O}_p\left(K^2 c_1^{(j)} s_0 \left(\frac{\log(p)}{N_j} \right)^{1/2} \right),
\end{aligned}$$

where K is defined in (A1) of Assumption 1. Meanwhile, based on Hoeffding's inequality, it follows

$$\left\| \frac{1}{N_j}\sum_{i=1}^{j} \left[\boldsymbol{J}^{(i)}(\boldsymbol{\beta}^0) - \mathbb{E}\{\boldsymbol{J}^{(i)}(\boldsymbol{\beta}^0)\} \right] \right\|_\infty = \mathcal{O}_p\left(\left(\frac{\log(p)}{N_j} \right)^{1/2} \right).$$

In summary,

$$\left\| \widehat{\boldsymbol{J}}^{(j)}/N_j - \boldsymbol{J}^0 \right\|_\infty = \mathcal{O}_p\left(K^2 c_1^{(j)} s_0 \left(\frac{\log(p)}{N_j} \right)^{1/2} \right).$$

Consequently, the compatibility condition holds by Corollary 6.8 in [4]. □

A.3. Proof of Theorem 3

Proof of Theorem 3. We first deal with the debiased term. Recall that $\widetilde{\boldsymbol{\gamma}}_r^{(b)} \in \mathbb{R}^p$ is the extension of $\widehat{\boldsymbol{\gamma}}_r^{(b)} \in \mathbb{R}^{p-1}$. Then,

$$\{\widetilde{\boldsymbol{\gamma}}_r^{(b)}\}^\top \sum_{j=1}^{b} \{\boldsymbol{X}^{(j)}\}^\top \left\{ \boldsymbol{y}^{(j)} - g'(\boldsymbol{X}^{(j)}\widehat{\boldsymbol{\beta}}^{(j)}) \right\}$$

$$= \{\widetilde{\gamma}_r^{(b)}\}^\top \sum_{j=1}^{b} \{\boldsymbol{X}^{(j)}\}^\top \left[\left\{ \boldsymbol{y}^{(j)} - g'(\boldsymbol{X}^{(j)}\boldsymbol{\beta}^0) \right\} + \{ g'(\boldsymbol{X}^{(j)}\boldsymbol{\beta}^0) - g'(\boldsymbol{X}^{(j)}\widehat{\boldsymbol{\beta}}^{(j)}) \} \right].$$

For $1 \le k \le n$, we let $\boldsymbol{X}_k^{(j)} \in \mathbb{R}^{1\times p}$ denote k-th row in $\boldsymbol{X}^{(j)}$, namely the covariates of k-th data in j-th batch. According to the mean value theorem,

$$g'(\boldsymbol{X}_k^{(j)}\widehat{\boldsymbol{\beta}}^{(j)}) = g'(\boldsymbol{X}_k^{(j)}\boldsymbol{\beta}^0) - g''(\eta_k^{(j)})\boldsymbol{X}_k^{(j)}(\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}^{(j)}),$$

where $\eta_k^{(j)} \in [\boldsymbol{X}_k^{(j)}\boldsymbol{\beta}^0, \boldsymbol{X}_k^{(j)}\widehat{\boldsymbol{\beta}}^{(j)}]$. Let $\eta^{(j)} = (\eta_1^{(j)}, \ldots, \eta_{n_j}^{(j)})^\top$ and $\boldsymbol{\Lambda}^{(j)} \in \mathbb{R}^{n_j \times n_j}$ is diagonal matrix with the diagonal element $\{g''(\boldsymbol{X}_k^{(j)}\widehat{\boldsymbol{\beta}}^{(j)}) - g''(\eta_k^{(j)})\}_{k=1}^{n_j}$. As a result,

$$\begin{aligned}
&\{\widetilde{\gamma}_r^{(b)}\}^\top \sum_{j=1}^{b} \{\boldsymbol{X}^{(j)}\}^\top \left\{ \boldsymbol{y}^{(j)} - g'(\boldsymbol{X}^{(j)}\widehat{\boldsymbol{\beta}}^{(j)}) \right\} \\
&= \{\widetilde{\gamma}_r^{(b)}\}^\top \sum_{j=1}^{b} \{\boldsymbol{X}^{(j)}\}^\top \left\{ \boldsymbol{y}^{(j)} - g'(\boldsymbol{X}^{(j)}\boldsymbol{\beta}^0) \right\} \\
&\quad + \{\widetilde{\gamma}_r^{(b)}\}^\top \sum_{j=1}^{b} \boldsymbol{J}^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)})(\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}^{(j)}) - \Pi_1,
\end{aligned}$$

where $\Pi_1 = \{\widetilde{\gamma}_r^{(b)}\}^\top \sum_{j=1}^{b} \{\boldsymbol{X}^{(j)}\}^\top \boldsymbol{\Lambda}^{(j)} \boldsymbol{X}^{(j)}(\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}^{(j)})$. Now, we focus on the online debiased lasso estimator. According to the above results,

$$\begin{aligned}
&\widehat{\tau}_r^{(b)}(\widehat{\beta}_{\mathrm{on},r}^{(b)} - \beta_r^0) \\
&= \{\widetilde{\gamma}_r^{(b)}\}^\top \sum_{j=1}^{b} \{\boldsymbol{X}^{(j)}\}^\top \left\{ g'(\boldsymbol{X}^{(j)}\boldsymbol{\beta}^0) - \boldsymbol{y}^{(j)} \right\} - \{\widetilde{\gamma}_r^{(b)}\}^\top \sum_{j=1}^{b} \boldsymbol{J}^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)})(\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}^{(j)}) \\
&\quad + \Pi_1 - \{\widetilde{\gamma}_r^{(b)}\}^\top \sum_{j=1}^{b} \boldsymbol{J}^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)})(\widehat{\boldsymbol{\beta}}^{(j)} - \widehat{\boldsymbol{\beta}}^{(b)}) + \widehat{\tau}_r^{(b)}(\widehat{\beta}_r^{(b)} - \beta_r^0) \\
&= \{\widetilde{\gamma}_r^{(b)}\}^\top \sum_{j=1}^{b} \{\boldsymbol{X}^{(j)}\}^\top \left\{ g'(\boldsymbol{X}^{(j)}\boldsymbol{\beta}^0) - \boldsymbol{y}^{(j)} \right\} + \Pi_1 \\
&\quad + \{\widetilde{\gamma}_r^{(b)}\}^\top \sum_{j=1}^{b} \boldsymbol{J}^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)})(\widehat{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0) + \widehat{\tau}_r^{(b)}(\widehat{\beta}_r^{(b)} - \beta_r^0) \\
&= \{\widetilde{\gamma}_r^{(b)}\}^\top \sum_{j=1}^{b} \{\boldsymbol{X}^{(j)}\}^\top \left\{ g'(\boldsymbol{X}^{(j)}\boldsymbol{\beta}^0) - \boldsymbol{y}^{(j)} \right\} + \Pi_1 + \Pi_2,
\end{aligned}$$

where

$$\Pi_2 = \sum_{j=1}^{b} \{\widetilde{\gamma}_r^{(b)}\}^\top \boldsymbol{J}^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)})(\widehat{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0) - \sum_{j=1}^{b} \{\widetilde{\gamma}_r^{(b)}\}^\top \boldsymbol{J}_r^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)})(\widehat{\beta}_r^{(b)} - \beta_r^0)$$

Therefore, the remaining part is to show that $|\Pi_i| = o_p(\sqrt{N_b}), i = 1, 2$.

First of all, according to the Lipschitz condition of $g''(\cdot)$,

$$\left|g''(\eta_k^{(j)})\boldsymbol{X}_k^{(j)}(\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}^{(j)}) - g''(\boldsymbol{X}_k^{(j)}\widehat{\boldsymbol{\beta}}^{(j)})\boldsymbol{X}_k^{(j)}(\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}^{(j)})\right| \leq l_g\{\boldsymbol{X}_k^{(j)}(\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}^{(j)})\}^2.$$

Then,

$$\left\|\boldsymbol{\Lambda}^{(j)}\boldsymbol{X}^{(j)}(\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}^{(j)})\right\|_1 \leq l_g\left\|\boldsymbol{X}^{(j)}(\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}^{(j)})\right\|_2^2 \leq K^2 l_g(c_1^{(j)}s_0\lambda_j)^2 n_j,$$

where the last inequality follows from Theorem 1 and the bounded assumption of $\boldsymbol{X}$. Meanwhile, the boundedness of $\|\boldsymbol{X}^{(j)}\widetilde{\boldsymbol{\gamma}}_r^{(b)}\|_\infty$ (bounded by K) could be shown by $(A2)$ in Assumption 1 and Theorem 2. Therefore, we obtain an upper bound for $|\Pi_1|$:

$$\begin{aligned}|\Pi_1| &\leq \sum_{j=1}^{b}\|\boldsymbol{X}^{(j)}\widetilde{\boldsymbol{\gamma}}_r^{(b)}\|_\infty\left\|\boldsymbol{\Lambda}^{(j)}\boldsymbol{X}^{(j)}(\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}^{(j)})\right\|_1 \\ &\leq \sum_{j=1}^{b} K^3 l_g(c_1^{(j)}s_0\lambda_j)^2 n_j = \mathcal{O}_p\left(c_1^{(b)}s_0^2K^3\log(p)\log(N_b)\right),\end{aligned}$$

where the last equation is from (A.9) in Lemma 2.

Next, we apply the Karush-Kuhn-Tucker (KKT) conditions. Let $\boldsymbol{e}_r \in \mathbb{R}^p$ denote the zero-vector except that the r-th element is one. Write

$$\begin{aligned}|\Pi_2| &= \left|\left(\{\widetilde{\boldsymbol{\gamma}}_r^{(b)}\}^\top\widehat{\boldsymbol{J}}^{(b)} - \{\widetilde{\boldsymbol{\gamma}}_r^{(b)}\}^\top\widehat{\boldsymbol{J}}_r^{(b)}\boldsymbol{e}_r^\top\right)(\widehat{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)\right| \\ &\leq \left\|\{\widetilde{\boldsymbol{\gamma}}_r^{(b)}\}^\top\widehat{\boldsymbol{J}}^{(b)} - \{\widetilde{\boldsymbol{\gamma}}_r^{(b)}\}^\top\widehat{\boldsymbol{J}}_r^{(b)}\boldsymbol{e}_r^\top\right\|_\infty\left\|\widehat{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0\right\|_1 \\ &\leq N_b\lambda_b \times c_1^{(b)}s_0\lambda_b = c_1^{(b)}s_0N_b\lambda_b^2 = \mathcal{O}_p\left(c_1^{(b)}s_0\log(p)\right).\end{aligned}$$

By the conditions in Theorem 3, we conclude $|\Pi_i| = o_p(\sqrt{N_b}), i = 1, 2$. As a result,

$$\widehat{\tau}_r^{(b)}(\widehat{\beta}_{\text{on},r}^{(b)} - \beta_r^0) = \{\widetilde{\boldsymbol{\gamma}}_r^{(b)}\}^\top\sum_{j=1}^{b}\{\boldsymbol{X}^{(j)}\}^\top\left\{g'(\boldsymbol{X}^{(j)}\boldsymbol{\beta}^0) - \boldsymbol{y}^{(j)}\right\} + o_p(\sqrt{N_b}).$$

We refer the two terms on the right-hand side as W_r and V_r in Theorem 3. □

A.4. Proof of Lemma 1

For the sake of completeness, we provide the proof of Lemma 1.

Proof of Lemma 1. We start from $|\Delta_1^{(b)}|$. Recall that

$$\Delta_1^{(b)} = \frac{1}{2}\sum_{j=1}^{b-1}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)^\top\left\{\boldsymbol{J}^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)}) - \boldsymbol{J}^{(j)}(\xi)\right\}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)$$

Let $\boldsymbol{v}^{(j)} = \boldsymbol{X}^{(j)}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)$, $\boldsymbol{w}^{(j)} = g''(\boldsymbol{X}^{(j)}\widehat{\boldsymbol{\beta}}^{(j)}) - g''(\boldsymbol{X}^{(j)}\xi)$ and $\mathrm{diag}(\boldsymbol{w}^{(j)})$ denote the diagonal matrix with diagonal element $\boldsymbol{w}^{(j)}$. Then,

$$\Delta_1^{(b)} = \frac{1}{2}\sum_{j=1}^{b-1}(\boldsymbol{v}^{(j)})^\top \mathrm{diag}(\boldsymbol{w}^{(j)})\boldsymbol{v}^{(j)} = \frac{1}{2}\boldsymbol{v}^\top \mathrm{diag}(\boldsymbol{w})\boldsymbol{v},$$

where $\boldsymbol{v} = ((\boldsymbol{v}^{(1)})^\top, \ldots, (\boldsymbol{v}^{(b-1)})^\top)^\top$ and $\boldsymbol{w} = ((\boldsymbol{w}^{(1)})^\top, \ldots, (\boldsymbol{w}^{(b-1)})^\top)^\top$. As a result,

$$|\Delta_1^{(b)}| \le \frac{1}{2}\|\boldsymbol{v}\|_2^2 \|\mathrm{diag}(\boldsymbol{w})\|_2 = \frac{1}{2}\|\boldsymbol{X}_\star^{(b-1)}(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)\|_2^2 \|\boldsymbol{w}\|_\infty.$$

It remains to find the upper bound of $\|\boldsymbol{w}\|_\infty$, that is,

$$\begin{aligned}\|\boldsymbol{w}\|_\infty &= \max_{1\le j\le b-1}\|\boldsymbol{w}^{(j)}\|_\infty = \max_{1\le j\le b-1}\|g''(\boldsymbol{X}^{(j)}\widehat{\boldsymbol{\beta}}^{(j)}) - g''(\boldsymbol{X}^{(j)}\xi)\|_\infty \\ &\le l_g \max_{1\le j\le b-1}\|\boldsymbol{X}^{(j)}(\widehat{\boldsymbol{\beta}}^{(j)} - \xi)\|_\infty \le 3Kl_g \max_{1\le j\le b-1} c_1^{(j)} s_0\lambda_j\end{aligned}$$

For ease of understanding, we could assume that $c_1^{(1)}s_0\lambda_1 = \max_{1\le j\le b-1} c_1^{(j)}s_0\lambda_j$ since the first data batch containing the least information may lead to the largest estimation error. Besides that, this assumption will not affect the outcome. The upper bound of $|\Delta_1|$ could always be absorbed in the upper bound of $|\Delta_2|$ due to $b = o(\log N_b)$ and $K^2 s_0^2 \log(p) N_b^{-1+\epsilon} = o(1)$.

Regarding Δ_2, the proof is structurally similar and is omitted. Recall that $|\Delta_3^{(b)}| = \{\bar{\boldsymbol{U}}^{(b-1)}(\boldsymbol{\beta}^0)\}^\top(\widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^0)$ and $\bar{\boldsymbol{U}}^{(b-1)}(\boldsymbol{\beta}^0)$ could be written as the sum of *i.i.d.* random variables, that is, $\bar{\boldsymbol{U}}^{(b-1)}(\boldsymbol{\beta}^0) = \sum_{i\in\mathcal{D}_{b-1}^\star}\boldsymbol{u}(y_i;\boldsymbol{x}_i,\boldsymbol{\beta}^0)$. Since $\mathbb{E}[\bar{\boldsymbol{U}}^{(b-1)}(\boldsymbol{\beta}^0)] = \boldsymbol{0}$ and $\|\boldsymbol{x}_i\|_\infty \le K$, by Hoeffding's inequality, with probability at least $1 - p^{-3}$,

$$\|\bar{\boldsymbol{U}}^{(b-1)}(\boldsymbol{\beta}^0)\|_\infty \le \lambda_{b-1}N_{b-1}/8.$$

The result of $|\Delta_4^{(b)}|$ can be shown in a straightforward fashion by Theorem 14.5 in [4].

The proof of Lemma 1 is complete. □

Acknowledgments

We are grateful to the editor, the associate editor, and the reviewers for their comments which led to a substantial improvement in the manuscript.

Funding

Luo is supported by the National Institute on Aging of the National Institutes of Health (R21AG083364) and the Startup Funds from Rutgers School of Public Health. Han is supported by the Hong Kong Research Grants Council, University Grants Committee (14301821) and The Hong Kong Polytechnic

University (P0044617, P0045351). Lin is supported by the Hong Kong Research Grants Council (14306219, 14306620), the National Natural Science Foundation of China (11961028) and Direct Grants for Research from the Chinese University of Hong Kong. Huang is supported by The Hong Kong Polytechnic University (P0042888, A0045417, A0045931).

References

[1] Agarwal, A., Xie, B., Vovsha, I., Rambow, O. and Passonneau, R. J. (2011). Sentiment analysis of twitter data. In *Proceedings of the Workshop on Language in Social Media at 2011 Association for Computational Linguistics* 30–38.

[2] Battey, H., Fan, J., Liu, H., Lu, J. and Zhu, Z. (2018). Distributed testing and estimation under sparse high dimensional models. *The Annals of Statistics* **46** 1352. MR3798006

[3] Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences* **2** 183–202. MR2486527

[4] Bühlmann, P. and van de Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media. MR2807761

[5] Cardot, H. and Degras, D. (2018). Online principal component analysis in high dimension: which algorithm to choose? *International Statistical Review* **86** 29–50. MR3796510

[6] Daubechies, I., Defrise, M. and De Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics* **57** 1413–1457. MR2077704

[7] Deshpande, Y., Javanmard, A. and Mehrabi, M. (2021). Online debiasing for adaptively collected high-dimensional data with applications to time series analysis. *Journal of the American Statistical Association* 1–14. MR4595482

[8] Dezeure, R., Bühlmann, P., Meier, L. and Meinshausen, N. (2015). High-dimensional inference: confidence intervals, p-values and R-software hdi. *Statistical Science* **30** 533–558. MR3432840

[9] Donoho, D. L. (1995). De-noising by soft-thresholding. *IEEE Transactions on Information Theory* **41** 613–627. MR1331258

[10] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96** 1348–1360. MR1946581

[11] Fang, Y. (2019). Scalable statistical inference for averaged implicit stochastic gradient descent. *Scandinavian Journal of Statistics* 1–16. MR4033800

[12] Fei, Z. and Li, Y. (2021). Estimation and inference for high dimensional generalized linear models: a splitting and smoothing approach. *Journal of Machine Learning Research* **22** 1–32. MR4253751

[13] Gentzkow, M., Kelly, B. and Taddy, M. (2019). Text as data. *Journal of Economic Literature* **57** 535–574.
[14] Han, R., Luo, L., Lin, Y. and Huang, J. (2021). Online debiased lasso. *arXiv preprint arXiv:2106.05925*.
[15] Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics* **35** 73–101. MR0161415
[16] Javanmard, A. and Montanari, A. (2014). Confidence intervals and hypothesis testing for high-dimensional regression. *Journal of Machine Learning Research* **15** 2869–2909. MR3277152
[17] Kogan, S., Levin, D., Routledge, B. R., Sagi, J. S. and Smith, N. A. (2009). Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* 272–280.
[18] Luo, L. and Song, P. X. K. (2020). Renewable estimation and incremental inference in generalized linear models with streaming datasets. *Journal of the Royal Statistical Society: Series B* **82** 69–97. MR4060977
[19] Luo, L., Wang, J. and Hector, E. C. (2023). Statistical inference for streamed longitudinal data. *Biometrika*. asad010. https://doi.org/10.1093/biomet/asad010. MR4667425
[20] Ma, J., Saul, L. K., Savage, S. and Voelker, G. M. (2009). Identifying suspicious URLs: an application of large-scale online learning. In *Proceedings of the 26th annual international conference on machine learning* 681–688.
[21] Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics* **22** 400–407. MR0042668
[22] Sakrison, D. J. (1965). Efficient recursive estimation: application to estimating the parameter of a covariance function. *International Journal of Engineering Science* **3** 461–483. MR0182082
[23] Schifano, E. D., Wu, J., Wang, C., Yan, J. and Chen, M. H. (2016). Online updating of statistical inference in the big data setting. *Technometrics* **58** 393–403. MR3520668
[24] Shi, C., Song, R., Lu, W. and Li, R. (2020). Statistical inference for high-dimensional models via recursive online-score estimation. *Journal of the American Statistical Association* 1–12. MR4309274
[25] Sun, L., Wang, M., Guo, Y. and Barbu, A. (2020). A novel framework for online supervised learning with feature selection. *arXiv preprint arXiv:1803.11521*.
[26] Sur, P. and Candès, E. J. (2019). A modern maximum-likelihood theory for high-dimensional logistic regression. *Proceedings of the National Academy of Sciences* **116** 14516–14525. MR3984492
[27] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B* **58** 267–288. MR1379242
[28] Toulis, P. and Airoldi, E. M. (2017). Asymptotic and finite-sample properties of estimators based on stochastic gradients. *The Annals of Statistics* **45** 1694–1727. MR3670193

[29] VAN DE GEER, S. (2008). High-dimensional generalized linear models and the lasso. *The Annals of Statistics* **36** 614–645. https://doi.org/10.1214/009053607000000929. MR2396809

[30] VAN DE GEER, S., BÜHLMANN, P., RITOV, Y. A. and DEZEURE, R. (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics* **42** 1166–1202. https://doi.org/10.1214/14-AOS1221. MR3224285

[31] ZHANG, C. H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics* **38** 894–942. MR2604701

[32] ZHANG, C. H. and ZHANG, S. S. (2014). Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B* **76** 217–242. MR3153940

[33] ZHAO, P. and YU, B. (2006). On model selection consistency of Lasso. *Journal of Machine Learning Research* **7** 2541–2563. MR2274449