# Renewable Estimation and Incremental Inference in Generalized Linear Models with Streaming Datasets

Lan Luo

*Department of Biostatistics, University of Michigan, Ann Arbor, Michigan 48109, U.S.A.*

E-mail: luolsph@umich.edu

Peter X.-K. Song

*Department of Biostatistics, University of Michigan, Ann Arbor, Michigan 48109, U.S.A.*

E-mail: pxsong@umich.edu

**Summary**. This paper presents an incremental updating algorithm to analyze streaming datasets using generalized linear models. The proposed method is formulated within a new framework of renewable estimation and incremental inference, in which the maximum likelihood estimator is renewed with current data and summary statistics of historical data. Our framework can be implemented within a popular distributed computing environment, known as Apache Spark, to scale up computation. Consisting of two data-processing layers, the Rho architecture enables to accommodate inference related statistics and to facilitate sequential updating of the statistics used in both estimation and inference. We establish estimation consistency and asymptotic normality of the proposed renewable estimator, in which the Wald test is utilized for an incremental inference. Our methods are examined and illustrated by various numerical examples from both simulation experiments and a real-world data analysis.

*Keywords*: Incremental statistical analysis; Lambda architecture; Online learning; Stochastic gradient descent algorithm; Spark computing platform

## 1. Introduction

We consider a classical problem where a series of cross-sectional datasets becomes available sequentially. Such type of data collection is pervasive in practice, which is referred to as streaming datasets throughout this paper. Statistical analysis of streaming datasets has recently drawn a considerable attention in the emerging field of Big Data analytics due to the availability of modern powerful computing platforms such as the Apache Spark (Bifet et al., 2015). The key methodology relevant to such data analysis pertains to algorithms that allow to sequentially update certain statistics of interest. For example, sample mean may be recursively updated along data streams in which only previous sample means, instead of the entire historical subject-level data, is needed. Specifically, consider two datasets arriving sequentially, where $D_1 = (x_{11}, ..., x_{1n_1})$ denotes the first dataset of $n_1$ observations. Suppose one wants to update the sample mean when the second data batch $D_2 = (x_{21}, ..., x_{2n_2})$ of $n_2$ observations arrives. Let $\delta(D_1)$ denote the sample mean for $D_1$, which can be easily updated with the new batch $D_2$; that is,

$$\delta(D_1 \cup D_2) = \frac{1}{n_1+n_2}\left(\sum_{i=1}^{n_1} x_{1i} + \sum_{i=1}^{n_2} x_{2i}\right) = \frac{1}{n_1+n_2}\left(n_1\delta(D_1) + \sum_{i=1}^{n_2} x_{2i}\right). \quad (1)$$

The defining feature in the above operation is that the mean from the previous data, $\delta(D_1)$, rather than the data $D_1$ itself, is used in the calculation. In this paper, a statistic that satisfies such property is termed as a *renewable estimator*. Indeed, the recursive operation exemplified in (1) works for many other statistics, such as sample moments and the least squares estimator in the linear model (Stengel, 1994). This is because these statistics take certain linear functions of data, so that a decomposition similar to (1) between current and past data is feasible (see

Section 3.3 for the detail). Using only summary statistics of previous data, instead of historical raw data, is conceptually linked to sufficient statistic, and is of critical importance in handling Big Data as far as computing memory and speed concern. This strategy has been widely advocated in the literature of online learning, incremental analytics, matrix or tensor decomposition and classification, and online Bayesian inference; see Bucak and Gunsel (2009); Cardot and Degras (2015); Nion and Sidiropoulos (2009); Qamar et al. (2014), among others.

Whether or not, and if so, to which extent, does the above renewability property seen in (1) hold in general? For example, can the maximum likelihood estimation (MLE), one of the most important statistical estimation and inference methods, may be updated sequentially in a similar fashion to the renewable procedure given in (1)? If not, how about MLE as a sufficient statistic? Answers to these questions are not trivial, because the maximum likelihood estimator is typically a nonlinear function of data, and often has no closed-form expression. Thus, MLE solution can only be obtained numerically by iterative algorithms, such as Newton-Raphson. In this paper, we choose the class of generalized linear models (GLMs) as an exemplary setting to illustrate the feasibility for finding answers to the above questions. It is known that the GLMs play a central role in regression analysis, and the renewable analytics developed in such context will provide a useful arsenal for regression analysis of streaming data. Moreover, in the GLM setting, the class of exponential dispersion (ED) models (Jørgensen, 1997) gives a connection between sufficient statistics and MLEs, which helps find solutions to the above questions.

The interest in developing procedures allowing "quick" updates of parameter estimates along with sequentially arrived data may be dated back five decades or so. Robbins and Monro (1951) proposed a seminal recursive estimation method that has become a very popular technique, namely the well-known *stochastic gradient descent* (SGD) algorithm that has been extensively used in the field of machine learning. The SGD method is applied for a data sequence in the form of an open-ending set of independent observations, $y_i \overset{i.i.d.}{\sim} f(y;\boldsymbol{\theta}_0)$, under a model $f(\cdot)$ with a common unknown parameter $\boldsymbol{\theta}_0$. Estimation of $\boldsymbol{\theta}_0$ may be carried out sequentially by a forward updating procedure, with a single data point $y_i$ involved at each iteration. That is, $\boldsymbol{\theta}_i^{\mathrm{sgd}} = \boldsymbol{\theta}_{i-1}^{\mathrm{sgd}} + \gamma_i \boldsymbol{C}_i \nabla_{\boldsymbol{\theta}} \log f(y_i;\boldsymbol{\theta}_{i-1}^{\mathrm{sgd}})$, where $\gamma_i > 0$ is a pre-specified learning rate sequence such that $i\gamma_i \to \gamma$ as $i \to \infty$ and $\{\boldsymbol{C}_i\}$ is a certain sequence of positive-definite matrices. Throughout this paper, $\nabla_{\boldsymbol{\theta}}$ denotes the gradient operation with respect to the model parameter $\boldsymbol{\theta}$. This updating procedure is later termed as "explicit SGD" in Toulis et al. (2014). Under the condition that $\gamma_i \boldsymbol{C}_i \to \boldsymbol{\mathcal{I}}^{-1}(\boldsymbol{\theta}_0)$, $i \to \infty$ where $\boldsymbol{\mathcal{I}}(\boldsymbol{\theta}_0)$ is the Fisher information matrix, this updating method enjoys some theoretical guarantees. For example, as $i \to \infty$, $\boldsymbol{\theta}_i^{\mathrm{sgd}} \xrightarrow{p} \boldsymbol{\theta}_0$ with the optimal asymptotic efficiency, namely, its asymptotic covariance matrix is $\boldsymbol{\mathcal{I}}^{-1}(\boldsymbol{\theta}_0)$.

However, the SGD method is generally not robust to learning rate misspecification, and the algorithm may fail to converge if $\gamma$ is too large. An improvement, called "implicit SGD" by Toulis et al. (2014), is given by $\boldsymbol{\theta}_i^{\mathrm{im}}$ that appears in both sides of the updating equation, *i.e.*, $\boldsymbol{\theta}_i^{\mathrm{im}} = \boldsymbol{\theta}_{i-1}^{\mathrm{im}} + \gamma_i \boldsymbol{C}_i \nabla_{\boldsymbol{\theta}} \log f(y_i;\boldsymbol{\theta}_i^{\mathrm{im}})$. According to the comparison of these two versions of SGD algorithms in the GLMs, Toulis et al. (2014) concluded that the implicit SGD appeared more robust to learning rate misspecification. To improve statistical efficiency, Toulis et al. (2014) further proposed the averaged implicit SGD (AI-SGD); see the detail in Section 2.1. To avoid calculating the inverse of Hessian matrix, some alternative versions of SGD are proposed with adapted learning rates from diagonal elements of an approximated Hessian, such as *SGD-QN* (Bordes et al., 2009) and *AdaGrad* (Duchi et al., 2011). Although such alternative procedures can achieve the same computation speed as the first-order methods, they are not useful for statistical inference because only part of the information matrix (*i.e.* Hessian's diagonal elements) is recorded and updated over iterations.

There are some online second-order methods such as Natural Gradient (NG) algorithm (Amari et al., 2000) and Online Newton Step (Hazan et al., 2007) that maintain complete information matrices over iterations. Similar to SGD, an outer product of the first gradients is used to approximate the negative Hessian, and its inverse is updated through the Sherman-Morrison formula. This updating scheme is widely used; see Vaits et al. (2013); Hao et al. (2016). However, this outer-product approximation to the Fisher information may not work well in general.

In the setting beyond the conventional likelihood framework, due to the failing of the Bartlett Identity (Song, 2007, Chapter 2), the Fisher information alone cannot provide a valid statistical inference. For online quasi-Newton methods, both Broyden-Fletcher-Goldfarb-Shanno (BFGS) (Nocedal and Wright, 1999) and limited memory BFGS (LBFGS) (Liu and Nocedal, 1989) algorithms have been modified for streaming data, respectively, termed as oBFGS and oLBFGS algorithms (Schraudolph et al., 2007; Bordes et al., 2009). But in these procedures, it is unclear whether estimated approximate Hessian is appropriate for statistical inference. A detailed comparison among these second-order online methods is available in Appendix A.1.

Although some relevant analytic expressions of the asymptotic variances have been derived in both explicit and implicit SGD (Toulis and Airold, 2017), the work of developing online confidence intervals remains unexplored due to the lack of suitable asymptotic results that may be directly applied to establish online inference. A recent paper by Fang (2019) proposed a perturbation-based resampling method to construct confidence intervals for AI-SGD. Even though this online bootstrap procedure can be parallelized to improve computational efficiency, as shown in the simulation studies later in the paper, it does not achieve desirable statistical efficiency and may produce misleading inference in the case of large regression parameters.

In addition to the SGD types of recursive algorithms, several cumulative updating methods have been proposed to specifically perform sequential updating of regression coefficient estimators, including the online least squares estimator (OLSE) for the linear model by Stengel (1994), the cumulative estimating equation (CEE) estimator and the cumulatively updated estimating equation (CUEE) estimator by Schifano et al. (2016) for nonlinear models. Even though CUEE is shown to have less estimation bias than CEE with finite sample sizes, its estimation consistency has been established upon a strong regularity condition: the total number of streaming datasets, *say*, $B$, needs to satisfy the order of $B = \mathcal{O}(n_j^k)$, with $k < 1/3$ for all$j$, where $n_j$ is the size of the $j$-th data batch (Lin and Xi, 2011; Schifano et al., 2016). This condition is also required by CEE for its estimation consistency. This implies a very strong restriction for these two methods; for example, their estimation consistency may not be guaranteed in the situation where streaming datasets arrive perpetually with $B \to \infty$. Our proposed renewable estimation method overcomes this unnatural restriction. Section 2.2 presents a more detailed review of these existing methods.

Streaming data analytics may be implemented in the so-called Lambda architecture (Marz and Warren, 2015). It is a realtime Big Data system of computing and storage with a synchronized processing of batch and stream data flows. The Lambda architecture consists of three layers: the speed layer, the batch layer, and the serving layer. Figure 1 shows a schematic outline as to how the speed and batch layers interact when a new data batch arrives. Transient and rough realtime views are captured at the speed layer using incremental algorithms, where previously stored views are updated with an incoming data batch to generate renewed views. Indeed, SGD is one of the most popular incremental algorithms widely used to process high-throughput streaming data via the Spark system (Bifet et al., 2015). The batch layer stores a constantly growing data and continuously recomputes the batch views when new data batch arrives. Despite latency, the batch layer refines results produced in the speed layer where estimation accuracy cannot be maintained consistently. Then the two view outputs are stored in the serving layer for queries. This architecture is flexible and applicable to a wide range of streaming data analytics in which the batch layer stores all sequentially accumulated raw data and produces reliable results via re-computations. Unfortunately, this powerful architecture has completely ignored the need of realtime statistical inference; for example, there are no gears in the system designed to sequentially compute and store Fisher information or as such, a critical piece required for statistical inference. To overcome, in this paper we propose to expand the speed layer by adding a new "inference layer", and name this new sub-architecture as "Rho architecture" (from the initial of Greek word "stream", $\rho\varepsilon\nu\mu\alpha$). Figure 2 in Section 3.2 displays the resulting expanded architecture allowing to conduct statistical inference with streaming data.

In the proposed Rho architecture, we aim to address three basic questions: (i) what types of summary statistics to be stored in the inference layer; (ii) how to update those summary

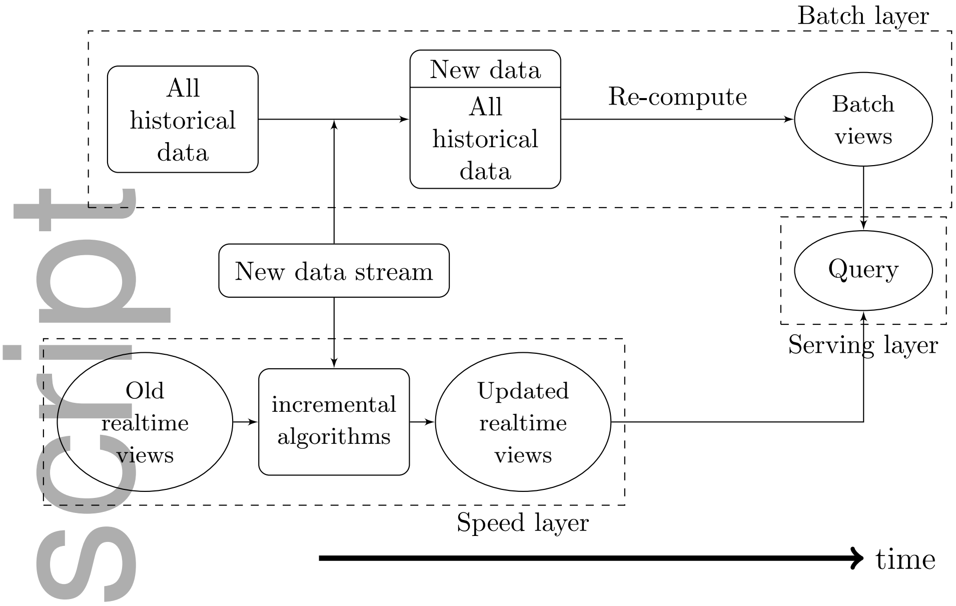

**Fig. 1.** Diagram concerning the flow of new data stream through the batch and speed layers in the Lambda architecture. Serving layer is responsible for indexing and exposing the views from batch and speed layers so that they can be queried.

statistics required for estimation and inference without use of previous raw data; and (iii) how to optimize the estimation efficiency of renewable estimation so it may be asymptotically equivalent to the MLE obtained from the entire dataset. Our goal is to fit a GLM (McCullagh and Nelder, 1983) $\mathbb{E}(y_i \mid \boldsymbol{x}_i) = g(\boldsymbol{x}_i^T\boldsymbol{\beta})$, $i = 1, \ldots, N_b$, where $g(\cdot)$ is a known link function and $N_b$ is the sample size of aggregated streaming data up to batch $b$, $N_b = \sum_{j=1}^{b} n_j$. At batch $b \geq 2$, a total of $N_b$ observations becomes available in a series of $b$ data batches, denoted by $D_1 = \{\boldsymbol{y}_1, \boldsymbol{X}_1\}, \ldots, D_b = \{\boldsymbol{y}_b, \boldsymbol{X}_b\}, \ldots$, where $\boldsymbol{y}$ and $\boldsymbol{X}$ are the generic notations of response variables and associated covariates. Under a fixed design, suppose each observation is drawn from $(y_i; \boldsymbol{x}_i) \sim f(y; \boldsymbol{x}, \boldsymbol{\beta}_0, \phi_0), i = 1, \ldots, N_b$ independently, where $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ is the true value of the parameter of interest and $\phi_0$ is the true value of a nuisance parameter. Let $D_b^\star = \{D_1, \ldots, D_b\}$ denote the cumulative data up to batch $b$. For convenience, slightly abusing the notation, we use $D_b$ (a single batch $b$) or $D_b^\star$ (an aggregation of $b$ batches) as the respective sets of indices for subjects involved. For a GLM, we may write out the associated log-likelihood function in the form of an ED model (Jørgensen, 1997):

$$\ell_{N_b}(\boldsymbol{\beta}, \phi; D_b^\star) = \sum_{i \in D_b^\star} \log f(y_i; \boldsymbol{x}_i, \boldsymbol{\beta}, \phi) = \sum_{i \in D_b^\star} \log a(y_i; \phi) - \frac{1}{2\phi} \sum_{i \in D_b^\star} d(y_i; \mu_i), \quad (2)$$

where $d(y_i; \mu_i)$ is the unit deviance function involving the mean parameter $\mu_i = \mathbb{E}(y_i \mid \boldsymbol{x}_i)$, and $a(\cdot)$ is a suitable normalizing factor depending only on the dispersion parameter $\phi > 0$. The systematic component of a GLM takes the form: $\mu_i = g(\boldsymbol{x}_i^T\boldsymbol{\beta})$, $i \in D_b^\star$. It is known that in the Gaussian linear model, the dispersion parameter $\phi$ is the variance parameter, and in both Bernoulli logistic and Poisson log-linear regression models, $\phi = 1$. Denote the (unit) score function by $\boldsymbol{U}(y_i; \boldsymbol{x}_i, \boldsymbol{\beta}) := \nabla_{\boldsymbol{\beta}} d(y_i; \mu_i)$. Then, the MLE $\hat{\boldsymbol{\beta}}_b^\star$ satisfying $\sum_{i \in D_b^\star} \boldsymbol{U}(y_i; \boldsymbol{x}_i, \boldsymbol{\beta}) = \boldsymbol{0}$ with the aggregated data $D_b^\star$ is the oracle estimator, which in general has no closed-form solution. It is often obtained numerically by certain iterative algorithms such as Newton-Raphson. Note that in the GLM the MLE $\hat{\boldsymbol{\beta}}_b^\star$ is derived with no involvement of nuisance parameter $\phi$ due to the so-called parameter orthogonality (Cox and Reid, 1987). For the detail of the MLE, refer to for example, McCullagh and Nelder (1983) and Song (2007, Chapter 2). Thus, unlike the case of the linear model where the MLE has an explicit closed-form expression, exact sequential updating procedures similar to (1) are generally unavailable for the GLMs.

The focus of this paper is to develop a new online framework in which both likelihood estimation and inference can be updated with current data and summary statistics of historical data. Our new contributions include: (i) we propose a Rho architecture as an expansion of the Spark's Lambda architecture for the purpose of online statistical inference; (ii) the proposed renewable estimator is shown to be asymptotically equivalent to the oracle MLE without strong condition $B = \mathcal{O}(n_j^k)$, $k < 1/3$; (iii) the $\ell_2$-norm difference between our renewable estimator and the oracle MLE vanishes as the total sample size increases; and (iv) being computationally advantageous, our method does not require a re-access to any old subject-level data after the completion of current updating step. Thus, our renewable estimation method is computationally efficient to address the challenge of data storage and data processing, which is particularly useful in the case where the number of data batches increases fast and/or perpetually. Also, our method provides a realtime interim inference based on the Wald test.

This paper is organized as follows. Section 2 gives a brief overview on existing methods to which the proposed method is compared. Section 3 presents our renewable estimation framework and incremental updating algorithm to compute renewable estimates. Section 4 includes some key large sample properties and hypothesis testing methods. Section 5 presents numerical implementation and some examples of commonly used GLMs. Section 6 presents simulation results of the proposed method with comparisons to the oracle MLE and existing online methods. Section 7 illustrates the proposed method by a real data application. Concluding remarks are provided in Section 8. All technical details are included in the appendix and the supplementary materials.

## 2. Existing Methods

There are two primary classes of online data analytics developed in the literature, including stochastic gradient descent algorithms and sequential estimation procedures. At an intermediary batch $b$, $\hat{\boldsymbol{\beta}}_b^\star$ denotes the oracle MLE estimator obtained with the entire cumulative dataset $D_b^\star$, and $\tilde{\boldsymbol{\beta}}_b$ denotes a renewable estimator with the same dataset $D_b^\star$. Throughout this paper, a hat "$\wedge$" over a symbol (e.g. $\hat{\boldsymbol{\beta}}$) denotes MLE, and "$\star$" in the superscript (e.g. $\hat{\boldsymbol{\beta}}_b^\star$) indicates a statistic derived from a cumulative dataset $D_b^\star$; otherwise, it is based on a single data batch (e.g. $\hat{\boldsymbol{\beta}}_b$ from $D_b$). Likewise, "$\sim$" over a symbol (e.g. $\tilde{\beta}$) denotes a quantity obtained sequentially by an incremental algorithm. For example, $\tilde{\boldsymbol{\beta}}$ denotes an estimator obtained by an online updating procedure (e.g. online LSE). For convenience, we list all necessary notations in Table A.2 in the appendix.

### 2.1. Stochastic Gradient Descent Algorithm

**Averaged Implicit SGD.** Toulis et al. (2014) proposed an averaged implicit stochastic gradient descent (AI-SGD) algorithm that is shown to be more stable than the explicit SGD algorithm. Later, Fang (2019) extended AI-SGD by adding a random weight $W_i^{(s)}$ to the gradient, resulting in the following implicit SGD procedure:

$$\boldsymbol{\beta}_i^{(\mathrm{s})\mathrm{im}} = \boldsymbol{\beta}_{i-1}^{\mathrm{im}} + \gamma_i W_i^{(s)} \boldsymbol{U}(y_i; \boldsymbol{x}_i, \boldsymbol{\beta}_i^{(\mathrm{s})\mathrm{im}}), \quad \boldsymbol{\beta}_i^{(\mathrm{s})\mathrm{aim}} \quad = \frac{1}{i}\sum_{k=1}^{i} \boldsymbol{\beta}_k^{(\mathrm{s})\mathrm{im}}, \quad i = 1, \ldots, N_b. \tag{3}$$

When fixing $W_i^{(s)} \equiv 1$, (3) gives the AI-SGD estimation. Using samples drawn from, *say*, $W_i^{(s)} \overset{i.i.d.}{\sim}$ Exponential(1), $s = 1, \ldots, S$, one obtains $S$ copies of $\boldsymbol{\beta}_i^{(\mathrm{s})\mathrm{aim}}$. Further using these replicates, one can assess the variability of $\boldsymbol{\beta}_i^{(\mathrm{s})\mathrm{aim}}$ and calculate the empirical standard error of the AI-SGD estimator for statistical inference. In Section 6, through simulation studies we compare our renewable estimation method with this AI-SGD method.

### *2.2. Sequential Updating Methods*

There are several sequential updating procedures in the literature, proposed by Lin and Xi (2011) and Schifano et al. (2016), among others. Here we present a brief introduction to this class of methods, and more details may be found in the Supplementary Material section S1.

**Online Least Squares Estimation (OLSE).** Consider a linear model $y_i = \boldsymbol{x}_i^T \boldsymbol{\beta}_0 + \epsilon_i$, with *i.i.d.* errors $\epsilon_i$'s, $i = 1, \ldots, N_b$. The LSE for the current single data batch $D_b$ is $\hat{\boldsymbol{\beta}}_b = (\boldsymbol{X}_b^T \boldsymbol{X}_b)^{-1} \boldsymbol{X}_b^T \boldsymbol{y}_b$. With initial $\tilde{\boldsymbol{\beta}}_1^{\text{olse}} = \hat{\boldsymbol{\beta}}_1$, the OLSE (Schifano et al., 2016), $\tilde{\boldsymbol{\beta}}_b^{\text{olse}}$, proceeds recursively according to the following decomposition:

$$\tilde{\boldsymbol{\beta}}_b^{\text{olse}} = \left( \sum_{j=1}^{b-1} \boldsymbol{X}_j^T \boldsymbol{X}_j + \boldsymbol{X}_b^T \boldsymbol{X}_b \right)^{-1} \left( \sum_{j=1}^{b-1} \boldsymbol{X}_j^T \boldsymbol{X}_j \tilde{\boldsymbol{\beta}}_{b-1}^{\text{olse}} + \boldsymbol{X}_b^T \boldsymbol{X}_b \hat{\boldsymbol{\beta}}_b \right), \ b = 2, 3, \ldots. \qquad (4)$$

**Online Estimating Equations.** Let $\boldsymbol{\beta}_0$ be a parameter value satisfying $\sum_{i \in D_b^\star} \mathbb{E}\{\boldsymbol{\psi}(y_i, \boldsymbol{x}_i; \boldsymbol{\beta}_0)\} = \mathbf{0}$, where $\boldsymbol{\psi}(\cdot)$ is an unbiased estimating function. Proposed first by Lin and Xi (2011) and adapted later to the sequential estimation setting by Schifano et al. (2016), a cumulative estimating equation (CEE) estimator, $\tilde{\boldsymbol{\beta}}_b^{\text{cee}}$, takes the following meta-estimation form:

$$\tilde{\boldsymbol{\beta}}_b^{\text{cee}} = \left( \tilde{\boldsymbol{A}}_{b-1}^{\text{cee}} + \boldsymbol{A}_b^{\text{cee}} \right)^{-1} \left( \tilde{\boldsymbol{A}}_{b-1}^{\text{cee}} \tilde{\boldsymbol{\beta}}_{b-1}^{\text{cee}} + \boldsymbol{A}_b^{\text{cee}} \hat{\boldsymbol{\beta}}_b \right), \quad \tilde{\boldsymbol{A}}_b^{\text{cee}} = \sum_{j=1}^{b} \boldsymbol{A}_j^{\text{cee}}, \ b = 1, 2, \ldots, \qquad (5)$$

with initial $\tilde{\boldsymbol{A}}_0^{\text{cee}} = \mathbf{0}_{p \times p}$, and $\boldsymbol{A}_b^{\text{cee}} = -\sum_{i \in D_b} \nabla_{\boldsymbol{\beta}} \boldsymbol{\psi}(y_i, \boldsymbol{x}_i; \hat{\boldsymbol{\beta}}_b)$ is the negative Hessian matrix of single data batch $D_b$.

It is easy to show that the bias of $\tilde{\boldsymbol{\beta}}_b^{\text{cee}}$ in (5) is of order $\mathcal{O}\left(\sum_{j=1}^{b} n_j^{-1/2}\right)$, which is $bn^{-1/2}$ in the case of equal batch size $n_j = n$ for all $j$. This suggests that for a small $n_j$, $b$ becomes a dominating factor in the bias, and consequently $\tilde{\boldsymbol{\beta}}_b^{\text{cee}}$ in (5) suffers an increased bias as $b \to \infty$. To reduce bias, a cumulatively updated estimating equation (CUEE) estimator is proposed by Schifano et al. (2016). See the related detail in the Supplementary Material section S1. It is worth pointing out that estimation consistency of CEE or CUEE is established under a strong regularity condition, $b = \mathcal{O}(n_j^k)$, for $k < 1/3$ and all $j$. This condition hardly holds for high throughput data streams, where $n_j$ is typically small while $b$ grows at a high rate. In this case, the theory of statistical inference is not yet available in the current literature.

## 3. Renewable Estimation

Let $\tilde{\boldsymbol{\beta}}_b$ be a renewable estimator, initialized by the MLE, $\tilde{\boldsymbol{\beta}}_1$ or $\hat{\boldsymbol{\beta}}_1$, from the first data batch $D_1$. For $b = 2, 3, \ldots$, a previous estimator $\tilde{\boldsymbol{\beta}}_{b-1}$ is sequentially updated to $\tilde{\boldsymbol{\beta}}_b$ when data batch $D_b$ arrives; after the updating, data batch $D_b$ is no longer accessible except estimate $\tilde{\boldsymbol{\beta}}_b$ and summary statistics $\boldsymbol{J}_b(D_b; \tilde{\boldsymbol{\beta}}_b)$ and $\tilde{\phi}_b$, which are carried forward in future calculations. Let $\boldsymbol{U}_b(D_b; \boldsymbol{\beta}) = \sum_{i \in D_b} \boldsymbol{U}(y_i; \boldsymbol{x}_i, \boldsymbol{\beta})$ be the score function of data batch $D_b$. Denote the single-batch negative Hessian by $\boldsymbol{J}_b(D_b; \boldsymbol{\beta}) := -\nabla_{\boldsymbol{\beta}} \boldsymbol{U}_b(D_b; \boldsymbol{\beta})$.

### *3.1. Method*

We begin with a simple scenario of two data batches $D_1$ and $D_2$, where $D_2$ arrives after $D_1$. We want to update the initial MLE $\hat{\boldsymbol{\beta}}_1$ (or $\hat{\boldsymbol{\beta}}_1^\star$) to a renewed MLE $\hat{\boldsymbol{\beta}}_2^\star$ without using any subject-level data but only some summary statistics from $D_1$. Here, MLE $\hat{\boldsymbol{\beta}}_1$ in a GLM satisfies the score equation, $\boldsymbol{U}_1(D_1; \hat{\boldsymbol{\beta}}_1) = \mathbf{0}$, and $\hat{\boldsymbol{\beta}}_2^\star$ satisfies the following aggregated score equation:

$$\boldsymbol{U}_1(D_1; \hat{\boldsymbol{\beta}}_2^\star) + \boldsymbol{U}_2(D_2; \hat{\boldsymbol{\beta}}_2^\star) = \mathbf{0}. \qquad (6)$$

Note that although the dispersion parameter $\phi$ is not involved in (6), it is needed in the calculation of the Fisher information. Solving (6) for $\hat{\boldsymbol{\beta}}_2^\star$ actually involves the use of subject-level data in both $D_1$ and $D_2$. To derive a renewable estimation, we take the first-order Taylor expansion of the first term in (6) around the MLE $\hat{\boldsymbol{\beta}}_1$,

$$\boldsymbol{U}_1(D_1;\hat{\boldsymbol{\beta}}_1)+\boldsymbol{J}_1(D_1;\hat{\boldsymbol{\beta}}_1)(\hat{\boldsymbol{\beta}}_1-\hat{\boldsymbol{\beta}}_2^{\star})+\boldsymbol{U}_2(D_2;\hat{\boldsymbol{\beta}}_2^{\star})+\mathcal{O}_p(\|\hat{\boldsymbol{\beta}}_2^{\star}-\hat{\boldsymbol{\beta}}_1\|^2)=\boldsymbol{0}. \tag{7}$$

Since $D_1$ and $D_2$ are independently sampled from the same underlying model with a common true parameter $\boldsymbol{\beta}_0$, when $\min\{n_1,n_2\}$ is large enough, under some mild regularity conditions, both $\hat{\boldsymbol{\beta}}_1$ and $\hat{\boldsymbol{\beta}}_2^{\star}$ are consistent estimators of $\boldsymbol{\beta}_0$ (e.g. Fahrmeir and Kaufmann (1985)). This implies that the error term $\mathcal{O}_p(\|\hat{\boldsymbol{\beta}}_2^{\star}-\hat{\boldsymbol{\beta}}_1\|^2)$ in (7) may be asymptotically ignored. Removing such term, we propose a new estimator $\tilde{\boldsymbol{\beta}}_2$ as a solution to the equation of the form:

$$\boldsymbol{U}_1(D_1;\hat{\boldsymbol{\beta}}_1)+\boldsymbol{J}_1(D_1;\hat{\boldsymbol{\beta}}_1)(\hat{\boldsymbol{\beta}}_1-\tilde{\boldsymbol{\beta}}_2)+\boldsymbol{U}_2(D_2;\tilde{\boldsymbol{\beta}}_2)=\boldsymbol{0}.$$

Since $\boldsymbol{U}_1(D_1;\hat{\boldsymbol{\beta}}_1)=\boldsymbol{0}$, the proposed estimator $\tilde{\boldsymbol{\beta}}_2$ satisfies the following estimating equation:

$$\boldsymbol{J}_1(D_1;\hat{\boldsymbol{\beta}}_1)(\hat{\boldsymbol{\beta}}_1-\tilde{\boldsymbol{\beta}}_2)+\boldsymbol{U}_2(D_2;\tilde{\boldsymbol{\beta}}_2)=\boldsymbol{0}. \tag{8}$$

Note that $\tilde{\boldsymbol{\beta}}_2$ in (8) approximates the oracle MLE $\hat{\boldsymbol{\beta}}_2^{\star}$ in (6) up to the second order asymptotic errors. Through (8), the initial $\hat{\boldsymbol{\beta}}_1$ is renewed by $\tilde{\boldsymbol{\beta}}_2$. Because of this, in this paper $\tilde{\boldsymbol{\beta}}_2$ is called *a renewable estimator* of $\boldsymbol{\beta}_0$, and equation (8) is termed as *an incremental estimating equation*. Numerically, it is rather straightforward to find $\tilde{\boldsymbol{\beta}}_2$ by, for example, the Newton-Raphson algorithm or Fisher scoring algorithm with $\phi=1$. Note that these two algorithms are equivalent in the GLM with a canonical link. That is, at the $(r+1)$-th iteration,

$$\tilde{\boldsymbol{\beta}}_2^{(r+1)}=\tilde{\boldsymbol{\beta}}_2^{(r)}+\left\{\boldsymbol{J}_1(D_1;\hat{\boldsymbol{\beta}}_1)+\boldsymbol{J}_2(D_2;\tilde{\boldsymbol{\beta}}_2^{(r)})\right\}^{-1}\left\{\boldsymbol{J}_1(D_1;\hat{\boldsymbol{\beta}}_1)(\hat{\boldsymbol{\beta}}_1-\tilde{\boldsymbol{\beta}}_2^{(r)})+\boldsymbol{U}_2(D_2;\tilde{\boldsymbol{\beta}}_2^{(r)})\right\},$$

where no subject-level data of $D_1$, but only the prior estimate $\hat{\boldsymbol{\beta}}_1$ and the prior negative Hessian $\boldsymbol{J}_1(D_1;\hat{\boldsymbol{\beta}}_1)$ are used in the above iterative algorithm. To speed up the calculations, we may avoid updating the negative Hessian $\boldsymbol{J}_2(D_2,\tilde{\boldsymbol{\beta}}_2^{(r)})$ at each iteration. Replacing $\tilde{\boldsymbol{\beta}}_2^{(r)}$ with $\hat{\boldsymbol{\beta}}_1$ leads to the following incremental updating algorithm:

$$\begin{aligned}\tilde{\boldsymbol{\beta}}_2^{(r+1)}&=\tilde{\boldsymbol{\beta}}_2^{(r)}+\left\{\sum_{j=1}^{2}\boldsymbol{J}_j(D_j;\hat{\boldsymbol{\beta}}_1)\right\}^{-1}\left\{\boldsymbol{J}_1(D_1;\hat{\boldsymbol{\beta}}_1)\left(\hat{\boldsymbol{\beta}}_1-\tilde{\boldsymbol{\beta}}_2^{(r)}\right)+\boldsymbol{U}_2\left(D_2;\tilde{\boldsymbol{\beta}}_2^{(r)}\right)\right\}\\&=\tilde{\boldsymbol{\beta}}_2^{(r)}+\left\{\boldsymbol{J}_1(\hat{\boldsymbol{\beta}}_1)+\boldsymbol{J}_2(\hat{\boldsymbol{\beta}}_1)\right\}^{-1}\tilde{\boldsymbol{U}}_2^{(r)},\end{aligned} \tag{9}$$

where $\tilde{\boldsymbol{U}}_2^{(r)}=\boldsymbol{J}_1(D_1;\hat{\boldsymbol{\beta}}_1)\left(\hat{\boldsymbol{\beta}}_1-\tilde{\boldsymbol{\beta}}_2^{(r)}\right)+\boldsymbol{U}_2\left(D_2;\tilde{\boldsymbol{\beta}}_2^{(r)}\right)$. In (9), $\tilde{\boldsymbol{\beta}}_2$ is iteratively solved by using the adjusted score function $\tilde{\boldsymbol{U}}_2^{(r)}$ and the aggregated negative Hessian $\left\{\boldsymbol{J}_1(\hat{\boldsymbol{\beta}}_1)+\boldsymbol{J}_2(\hat{\boldsymbol{\beta}}_1)\right\}$ evaluated at the previous estimate $\hat{\boldsymbol{\beta}}_1$. We name this algorithm (9) as *incremental updating algorithm*. Essentially, equation (9) presents a kind of gradient descent algorithm, so its solution will converge to the root of equation (8). Similar ideas have been used in the literature to speed up the calculation of Hessian matrix; see for example, Song et al. (2005). The difference between the proposed $\tilde{\boldsymbol{\beta}}_2$ and the oracle MLE $\hat{\boldsymbol{\beta}}_2^{\star}$ stems from an approximation to the score function $\boldsymbol{U}_1(D_1;\hat{\boldsymbol{\beta}}_2^{\star})$. As shown in Theorem 4.3, such distance vanishes at the rate of $1/N_2$, with $N_2=|D_2^{\star}|=n_1+n_2$. In practice, because the cumulative sample size $N_b=\sum_{j=1}^{b}n_j$ increases to infinity very fast, these two estimators, $\tilde{\boldsymbol{\beta}}_b$ and $\hat{\boldsymbol{\beta}}_b^{\star}$, are numerically very close, and eventually become the same. To run the algorithm (9), we extend the Spark Lambda architecture to store three key components $\left\{\hat{\boldsymbol{\beta}}_1,\boldsymbol{J}_1(D_1;\hat{\boldsymbol{\beta}}_1),\hat{\phi}_1\right\}$. Here, the initial $\hat{\phi}_1=\frac{1}{n_1-p}\sum_{i\in D_1}\frac{(y_i-\hat{\mu}_i)^2}{v(\hat{\mu}_i)}$ based on the Pearson residuals, where $\hat{\mu}_i=g(\boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}_1)$ and $v(\cdot)$ is the unit variance function.

Generalizing the above procedure to streaming datasets, we now propose a renewable estimation of $\boldsymbol{\beta}_0$ as follows. A renewable estimator $\tilde{\boldsymbol{\beta}}_b$ of $\boldsymbol{\beta}_0$ is defined as a solution to the following incremental estimating equation:

$$\sum_{j=1}^{b-1} \boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)(\tilde{\boldsymbol{\beta}}_{b-1}-\tilde{\boldsymbol{\beta}}_b)+\boldsymbol{U}_b(D_b;\tilde{\boldsymbol{\beta}}_b)=\mathbf{0}, \tag{10}$$

where $\hat{\boldsymbol{\beta}}_1 = \tilde{\boldsymbol{\beta}}_1$ at the initial data batch $D_1$. Note that when $b=2$, equation (10) reduces to equation (8). Let $\tilde{\boldsymbol{J}}_b = \sum_{j=1}^{b} \boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)$ denote the aggregated negative Hessian matrix. Solving equation (10) may be easily done by the following incremental updating algorithm:

$$\tilde{\boldsymbol{\beta}}_b^{(r+1)} = \tilde{\boldsymbol{\beta}}_b^{(r)} + \left\{\tilde{\boldsymbol{J}}_{b-1} + \boldsymbol{J}_b(D_b;\tilde{\boldsymbol{\beta}}_{b-1})\right\}^{-1}\tilde{\boldsymbol{U}}_b^{(r)}, \tag{11}$$

where the adjusted score $\tilde{\boldsymbol{U}}_b^{(r)} = \tilde{\boldsymbol{J}}_{b-1}(\tilde{\boldsymbol{\beta}}_{b-1} - \tilde{\boldsymbol{\beta}}_b^{(r)}) + \boldsymbol{U}_b(D_b;\tilde{\boldsymbol{\beta}}_b^{(r)})$ is updated over iterations. Again, algorithm (11) only uses subject-level data of current batch $D_b$ and summary statistics $\left\{\tilde{\boldsymbol{\beta}}_{b-1}, \tilde{\boldsymbol{J}}_{b-1}, \tilde{\phi}_{b-1}\right\}$ from historical data. Also, a consistent estimator of parameter $\phi$ is updated by $\tilde{\phi}_b = \frac{N_{b-1}-p}{N_b-p}\tilde{\phi}_{b-1} + \frac{n_b-p}{N_b-p}\hat{\phi}_b$, with $\hat{\phi}_b = \frac{1}{n_b-p}\sum_{i\in D_b}\frac{(y_i-\hat{\mu}_i)^2}{v(\hat{\mu}_i)}$.

### 3.2. *Rho Architecture*

Apache Spark is a unified data analytics platform for large-scale data processing. Built on a distributed computing paradigm, it offers high performance for both batch and streaming data. Its Lambda architecture is designed to achieve efficient communication and coordination between batch layer and speed layer to handle streaming data. To implement our proposed algorithm that provides both realtime estimation and statistical inference, we expand the speed layer in the Lambda architecture to accommodate inferential statistics, *i.e.* information matrices (in short "info.mats"), such as the Fisher information. As shown in Figure 2, the resulting Rho architecture consists of a speed layer and an inference layer responsible for inferential statistics updating. When a new data batch arrives, the speed layer updates the views (or estimates) in the GLMs with the utility of prior inferential statistics from the inference layer. Then, the updated views are sent back to the inference layer, where, together with the current data, realtime updates of information matrices are generated. The incremental updating algorithm in (11) is implemented in the Rho architecture.

### 3.3. *An example: Linear Model*

To see specific operational details discussed above, here we present the renewable estimation in the Gaussian linear model. Note that for the linear model, the proposed renewable estimation turns out to be identical to the online least squares estimation (OLSE) given in equation (4), with more details available in the Supplementary Material section S1.

EXAMPLE 1. *Consider data batch* $D_b = \{\boldsymbol{y}_b, \boldsymbol{X}_b\}$ *with outcome* $\boldsymbol{y}_b = (y_{b1},\ldots,y_{bn_b})^T$ *and covariates* $\boldsymbol{X}_b = (\boldsymbol{x}_{b1},\ldots,\boldsymbol{x}_{bn_b})^T$, *and* $\boldsymbol{y}_b|\boldsymbol{X}_b$ *are independently sampled from a Gaussian distribution with mean* $\boldsymbol{\mu}_b = (\mu_{b1},\ldots,\mu_{bn_b})^T$ *and variance* $\phi\boldsymbol{I}$ *such that* $\mu_{bi} = \mathbb{E}(y_{bi} \mid \boldsymbol{x}_{bi}) = \boldsymbol{x}_{bi}^T\boldsymbol{\beta}_0$ *and variance* $V(y_{bi} \mid \boldsymbol{x}_{bi}) = \phi_0$. *Here the variance function* $v(\mu_i) \equiv 1$. *Then, the score function and the corresponding negative Hessian for data batch* $D_b$ *are, respectively,* $\boldsymbol{U}_b(\boldsymbol{\beta}) = \boldsymbol{X}_b^T(\boldsymbol{y}_b - \boldsymbol{X}_b\boldsymbol{\beta})$, *and* $\boldsymbol{J}_b(\boldsymbol{\beta}) = \boldsymbol{X}_b^T\boldsymbol{X}_b$. *A closed-form expression for the renewable estimator of* $\boldsymbol{\beta}_0$ *is obtained directly by solving the incremental estimating equation* (10)*:*

$$\tilde{\boldsymbol{\beta}}_b = \left(\tilde{\boldsymbol{J}}_{b-1} + \boldsymbol{J}_b\right)^{-1}\left(\tilde{\boldsymbol{J}}_{b-1}\tilde{\boldsymbol{\beta}}_{b-1} + \boldsymbol{X}_b^T\boldsymbol{y}_b\right), \quad b = 1,2,\ldots.$$

*This* $\tilde{\boldsymbol{\beta}}_b$ *is calculated at the speed layer. By convention, the initials are* $\tilde{\boldsymbol{\beta}}_0 = \mathbf{0}_p$ *and* $\tilde{\boldsymbol{J}}_0 = \mathbf{0}_{p\times p}$.

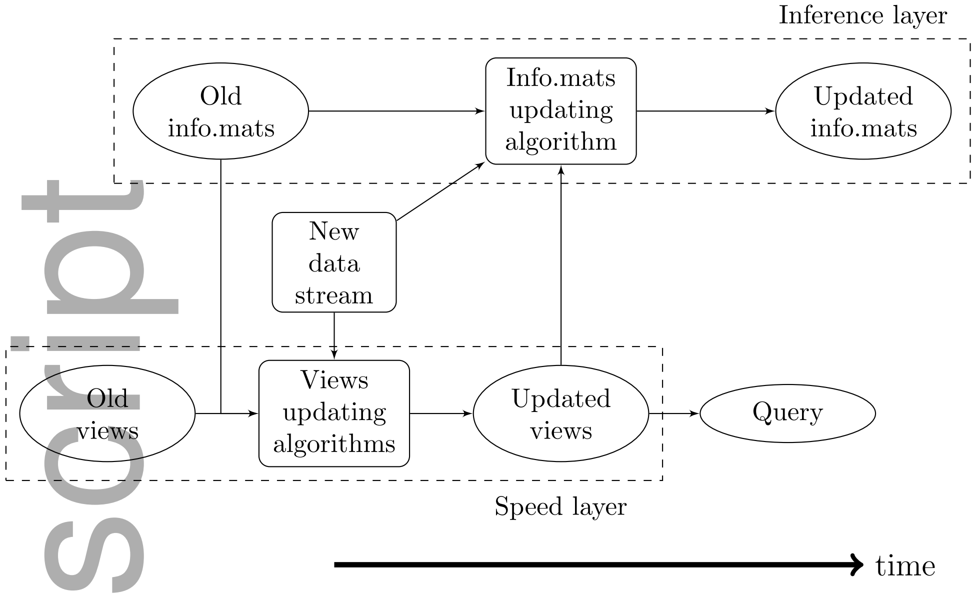

**Fig. 2.** Rho architecture: An expanded speed layer of the Lambda architecture with an addition of inference layer for incremental updating of quantities required by statistical inference.

*Moreover, an unbiased estimator of* $\phi_0$ *based on* $\tilde{\boldsymbol{\beta}}_b$ *takes the following recursive formula:*

$$\begin{aligned}\tilde{\phi}_b &= \frac{1}{N_b - p}\sum_{j=1}^{b}(\boldsymbol{y}_j - \boldsymbol{X}_j\tilde{\boldsymbol{\beta}}_b)^T(\boldsymbol{y}_j - \boldsymbol{X}_j\tilde{\boldsymbol{\beta}}_b) \\ &= \frac{1}{N_b - p}\left\{(N_{b-1} - p)\tilde{\phi}_{b-1} + \tilde{\boldsymbol{\beta}}_{b-1}^T\tilde{\boldsymbol{J}}_{b-1}\tilde{\boldsymbol{\beta}}_{b-1} + \boldsymbol{y}_b^T\boldsymbol{y}_b - \tilde{\boldsymbol{\beta}}_b^T\tilde{\boldsymbol{J}}_b\tilde{\boldsymbol{\beta}}_b\right\},\ b = 1, 2, \ldots .\end{aligned}$$

*The above* $\tilde{\phi}_b$ *is calculated and stored in the inference layer as part of the Fisher information calculation, given by* $\widetilde{Var}(\tilde{\boldsymbol{\beta}}_b) = \tilde{\phi}_b(\tilde{\boldsymbol{J}}_{b-1} + \boldsymbol{J}_b)^{-1}$*. Note that this estimated variance of* $\tilde{\boldsymbol{\beta}}_b$ *gives exactly the same standard error as that of the oracle MLE* $\hat{\boldsymbol{\beta}}_b^\star$*. The latter is obtained by fitting the linear model once with the entire data* $D_b^\star$*. So, the proposed renewable estimation does not lose any estimation efficiency, but is advantageous in data storage and computing speed.*

## 4. Large Sample Properties and Incremental Inference

In this section we first establish estimation consistency and asymptotic normality for the proposed renewable estimator, and then show its asymptotic equivalency to the oracle MLE. Also, we present the incremental inference based on the Wald statistic.

### 4.1. Large Sample Properties

For an arbitrary batch $b$, suppose $(y_i, \boldsymbol{x}_i)$ are *i.i.d.* samples from an exponential dispersion model with density $f(y; \boldsymbol{x}, \boldsymbol{\beta}, \phi)$, $i = 1, \ldots, N_b$, with mean $\mu_i = \mathbb{E}(y_i \mid \boldsymbol{x}_i) = g(\boldsymbol{x}_i^T\boldsymbol{\beta})$, $\boldsymbol{\beta} \in \Theta \subset \mathbb{R}^p$, and variance $\mathrm{V}(y_i \mid \boldsymbol{x}_i) = \phi v(\mu_i)$, $\phi > 0$ is the dispersion parameter, where $v(\cdot)$ is the known unit variance function. Let $\boldsymbol{\beta}_0$ and $\phi_0$ be the true parameters, respectively. Under the canonical link, denote $\boldsymbol{\mathcal{I}}_{N_b}(\boldsymbol{\beta}_0) = \sum_{i=1}^{N_b}\mathbb{E}\left\{\boldsymbol{U}_i\boldsymbol{U}_i^T\right\}/\phi = \sum_{i=1}^{N_b}\boldsymbol{x}_i v(\mu_i)\boldsymbol{x}_i^T$. Let $\mathcal{B}_{N_b}(\delta)$ be a neighborhood of $\boldsymbol{\beta}_0$, $\mathcal{B}_{N_b}(\delta) = \{\boldsymbol{\beta} : \|\boldsymbol{\mathcal{I}}_{N_b}^{T/2}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\| \leq \delta\} \in \Theta$, $\delta > 0$, where $\|\cdot\|$ is the $\ell_2$-norm. Here $\boldsymbol{\mathcal{I}}_{N_b}^{T/2}$ denotes the right Cholesky square root of $\boldsymbol{\mathcal{I}}_{N_b}(\boldsymbol{\beta}_0)$, according to $\boldsymbol{\mathcal{I}}_{N_b} = \boldsymbol{\mathcal{I}}_{N_b}^{1/2}\boldsymbol{\mathcal{I}}_{N_b}^{T/2}$. We postulate the following regularity conditions:

(C1) Divergence: the smallest eigenvalue of $\boldsymbol{\mathcal{I}}_{N_b}(\boldsymbol{\beta}_0)$ satisfies $\lambda_{\min}(\boldsymbol{\mathcal{I}}_{N_b}) \to \infty$, as $N_b \to \infty$.

(C2) $\boldsymbol{\mathcal{I}}_{N_b}(\boldsymbol{\beta})$ is positive-definite for all $\boldsymbol{\beta} \in \mathcal{B}_{N_b}(\delta)$.
(C3) The log-likelihood function $\ell(\boldsymbol{\beta}, \phi, \boldsymbol{x}; y)$ is twice continuously differentiable and $\boldsymbol{\mathcal{I}}_{N_b}(\boldsymbol{\beta})$ is Lipschitz continuous in $\Theta$.

Remark 1. *Under condition (C1), the neighborhood $\mathcal{B}_{N_b}(\delta)$ shrinks to a singleton $\boldsymbol{\beta}_0$, as $N_b \to \infty$. Condition (C2) is necessary for both consistency and asymptotic normality. Both (C1) and (C2) are the standard regularity conditions assumed by Fahrmeir and Kaufmann (1985). Different from the traditional MLE, the consistency for the renewable estimator requires the continuity assumption (C3) to be held over the whole parameter space $\Theta$, rather than over a neighborhood of $\boldsymbol{\beta}_0$. Since in the GLMs, the matrix $\boldsymbol{\mathcal{I}}_{N_b}(\boldsymbol{\beta})$ depends on $\boldsymbol{\beta}$ via the unit variance function $v(\cdot)$, the Lipschitz continuity condition automatically holds on a compact parameter space, which is sufficient for most applications.*

Theorem 4.1. *Under conditions (C1)-(C3), the renewable estimator $\tilde{\boldsymbol{\beta}}_b$ given in* (10) *is consistent, namely $\tilde{\boldsymbol{\beta}}_b \xrightarrow{p} \boldsymbol{\beta}_0$, as $N_b = \sum_{j=1}^{b} n_j \to \infty$.*

The proof of Theorem 4.1 is given in Section A.3 of the appendix.

Theorem 4.2. *Under conditions (C1)-(C3), the renewable estimator $\tilde{\boldsymbol{\beta}}_b$ is asymptotically normally distributed, that is,*

$$\sqrt{N_b}(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) \xrightarrow{d} \mathcal{N}\left(0, \boldsymbol{\Sigma}_0\right), \ \text{as } N_b = \sum_{j=1}^{b} n_j \to \infty,$$

*where $\boldsymbol{\Sigma}_0$ is the inverse of the Fisher information for a single observation at the true values.*

The proof of Theorem 4.2 is provided in Section A.4 of the appendix. It is interesting to notice that the asymptotic covariance matrix of the renewable estimator $\tilde{\boldsymbol{\beta}}_b$ given in Theorem 4.2 is the same as that of the oracle MLE $\hat{\boldsymbol{\beta}}_b^{\star}$. This implies that the proposed renewable estimator is fully efficient; see also Remark 2 below. With no need of historical subject-level data in the computation, using only the prior aggregated negative Hessian matrix stored in the Rho architecture, $\tilde{\boldsymbol{J}}_b = \sum_{j=1}^{b} \boldsymbol{J}_j(D_j; \tilde{\boldsymbol{\beta}}_j)$, we calculate the estimated asymptotic covariance matrix $\widetilde{\boldsymbol{\Sigma}_b}$ given by, $\widetilde{\boldsymbol{\Sigma}_b} = \left\{(N_b\tilde{\phi}_b)^{-1}\sum_{j=1}^{b} \boldsymbol{J}_j(D_j; \tilde{\boldsymbol{\beta}}_j)\right\}^{-1} = N_b\tilde{\phi}_b\tilde{\boldsymbol{J}}_b^{-1}$. It follows that the estimated variance matrix for $\tilde{\boldsymbol{\beta}}_b$ is given by

$$\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b) := \widetilde{\mathrm{Var}}(\tilde{\boldsymbol{\beta}}_b) = \frac{1}{N_b}\widetilde{\boldsymbol{\Sigma}_b} = \tilde{\phi}_b\tilde{\boldsymbol{J}}_b^{-1}. \tag{12}$$

Remark 2. *Because both SGD and AI-SGD may be regarded as special cases of the proposed renewable estimator, with $n_j = 1$ for all $j$, the result of Sakrison's asymptotic efficiency (Sakrison (1965)) remains true theoretically for AI-SGD (Toulis and Airoldi, 2015). Theorems 4.2 presents an extension of the efficiency theory for the GLMs with streaming data.*

The following theorem is the theoretical basis for the proposed renewable estimator $\tilde{\boldsymbol{\beta}}_b$, which is shown to be asymptotically equivalent to the oracle MLE $\hat{\boldsymbol{\beta}}_b^{\star}$.

Theorem 4.3. *Under conditions (C1)-(C3), the $\ell_2$-norm difference between the oracle MLE $\hat{\boldsymbol{\beta}}_b^{\star}$ and the proposed renewable estimator $\tilde{\boldsymbol{\beta}}_b$ vanishes at the rate of $N_b^{-1}$, namely*

$$\|\tilde{\boldsymbol{\beta}}_b - \hat{\boldsymbol{\beta}}_b^{\star}\|_2 = \mathcal{O}_p(1/N_b), \ \text{as } N_b \to \infty.$$

Theorem 4.3 implies that the renewable estimator achieves the optimal efficiency. The proof of Theorems 4.3 is included in Section A.5 of the appendix.

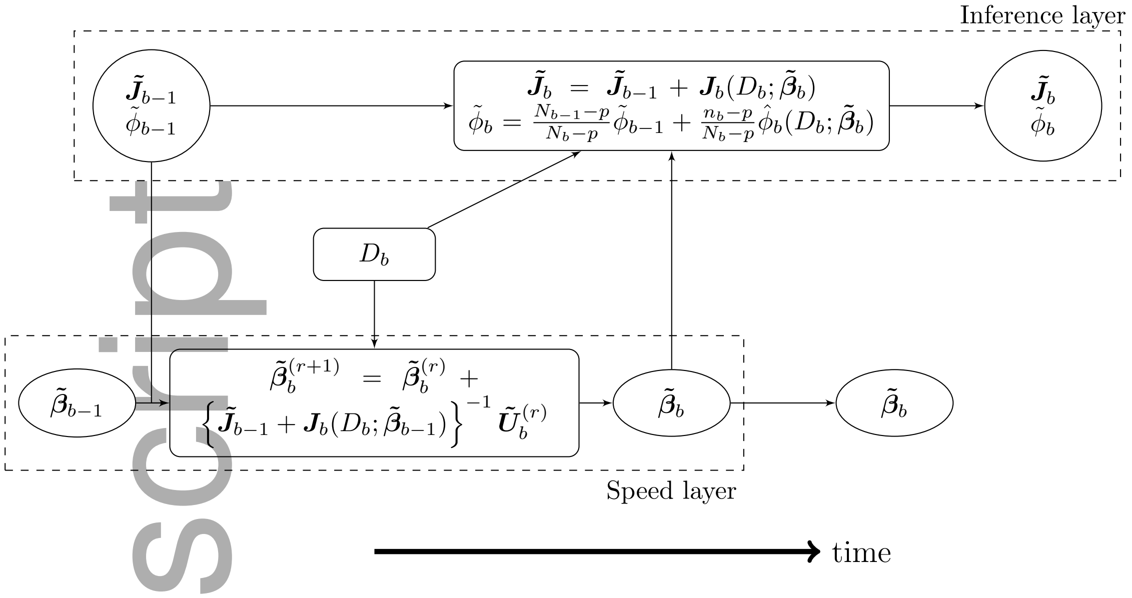
Inference layer
$\tilde{\boldsymbol{J}}_{b-1}$ $\tilde{\phi}_{b-1}$
$\tilde{\boldsymbol{J}}_b = \tilde{\boldsymbol{J}}_{b-1} + \boldsymbol{J}_b(D_b;\tilde{\boldsymbol{\beta}}_b)$
$\tilde{\phi}_b = \frac{N_{b-1}-p}{N_b-p}\tilde{\phi}_{b-1} + \frac{n_b-p}{N_b-p}\hat{\phi}_b(D_b;\tilde{\boldsymbol{\beta}}_b)$
$\tilde{\boldsymbol{J}}_b$ $\tilde{\phi}_b$
$D_b$
$\tilde{\boldsymbol{\beta}}_{b-1}$
$\tilde{\boldsymbol{\beta}}_b^{(r+1)} = \tilde{\boldsymbol{\beta}}_b^{(r)} + \left\{\tilde{\boldsymbol{J}}_{b-1} + \boldsymbol{J}_b(D_b;\tilde{\boldsymbol{\beta}}_{b-1})\right\}^{-1}\tilde{\boldsymbol{U}}_b^{(r)}$
$\tilde{\boldsymbol{\beta}}_b$
$\tilde{\boldsymbol{\beta}}_b$
Speed layer
time

**Fig. 3.** Diagram of the Rho architecture in which $\tilde{\boldsymbol{\beta}}_{b-1}$ is updated to $\tilde{\boldsymbol{\beta}}_b$ at the speed layer and $(\tilde{\boldsymbol{J}}_{b-1}, \tilde{\phi}_{b-1})$ are updated to $(\tilde{\boldsymbol{J}}_b, \tilde{\phi}_b)$ at the inference layer.

### *4.2. Incremental Inference*

The Wald test based on the asymptotic distribution of the renewable estimator in Theorem 4.2 is a straightforward approach to testing hypotheses of individual coefficients or of nested parameter sets. For $k < p$ and a pre-fixed null subvector $\boldsymbol{\beta}_1^{\text{null}}$, define the following null hypothesis parameter space $\Theta_{H_0} = \{(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2) = (\boldsymbol{\beta}_1^{\text{null}}, \beta_{k+1}, \ldots, \beta_p)\}$, a $(p-k)$-dimensional subspace of $\Theta$. The subvector $\tilde{\boldsymbol{\beta}}_{1b}$ of $\tilde{\boldsymbol{\beta}}_b$ corresponding to its first $k$ parameters follows asymptotically a $k$-dimensional marginal normal distribution, according to Theorem 4.2. Specifically, a suitable block-partition of the estimate $\tilde{\boldsymbol{\beta}}_b$ and its asymptotic variance matrix are given by, respectively, $\tilde{\boldsymbol{\beta}}_b = (\tilde{\boldsymbol{\beta}}_{1b}^T, \tilde{\boldsymbol{\beta}}_{2b}^T)^T$ and $\boldsymbol{\Sigma}_0 = [\boldsymbol{\Sigma}_{ij}]_{i,j=1,2}$, a two-by-two block matrix. Under the null hypothesis $H_0 : \boldsymbol{\beta}_1 = \boldsymbol{\beta}_1^{\text{null}}$, $\sqrt{N_b}\left(\tilde{\boldsymbol{\beta}}_{1b} - \boldsymbol{\beta}_1^{\text{null}}\right) \xrightarrow{d} \mathcal{N}_k(\boldsymbol{0}, \boldsymbol{\Sigma}_{11})$, as $N_b \to \infty$. This gives rise to the following asymptotic chi-square distribution with $k$ degrees of freedom. That is, under the null $H_0$,

$$\begin{aligned}\tilde{W}_b &= (\tilde{\boldsymbol{\beta}}_{1b} - \boldsymbol{\beta}_1^{\text{null}})^T \left\{\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b)_{11}\right\}^{-1} (\tilde{\boldsymbol{\beta}}_{1b} - \boldsymbol{\beta}_1^{\text{null}}) \\ &= (\tilde{\boldsymbol{\beta}}_{1b} - \boldsymbol{\beta}_1^{\text{null}})^T \left\{\left(\tilde{\phi}_b \tilde{\boldsymbol{J}}_b^{-1}\right)_{11}\right\}^{-1} (\tilde{\boldsymbol{\beta}}_{1b} - \boldsymbol{\beta}_1^{\text{null}}) \xrightarrow{d} \chi_k^2,\end{aligned} \tag{13}$$

where $\left(\tilde{\phi}_b \tilde{\boldsymbol{J}}_b^{-1}\right)_{11}$ is the $(1,1)$-block of matrix $\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b)$ in (12). Thus, a $100(1-\alpha)\%$ confidence ellipsoid for $\boldsymbol{\beta}_1$ is given by $\boldsymbol{\mathcal{C}} = \left\{\boldsymbol{\beta}_1 : (\tilde{\boldsymbol{\beta}}_{1b} - \boldsymbol{\beta}_1)^T \left\{\left(\tilde{\phi}_b \tilde{\boldsymbol{J}}_b^{-1}\right)_{11}\right\}^{-1} (\tilde{\boldsymbol{\beta}}_{1b} - \boldsymbol{\beta}_1) < \chi_k^2(\alpha)\right\}$.

It is worth pointing out that Rao's score test and Wilks' likelihood-ratio test are not discussed here because both methods require the renewable estimates of $\boldsymbol{\beta}$ under $H_0$. Unlike the above Wald test which is just a direct byproduct of Theorem 4.2, the other two tests involve constrained estimates under the null. The related estimation does not seem to follow incremental operations. Thus, incremental inference based on Rao's score test or Wilks' likelihood ratio test is an open problem in the setting of streaming data analysis.

## 5. Implementation

### *5.1. Rho architecture and pseudo code*

The proposed renewable analytics may be implemented in the Rho architecture in Figure 2. The work flow chart in Figure 3 facilitates the organization of the pseudo code for key numerical calculations, summarized by Algorithm 1.

**Algorithm 1:** Implementation of the renewable analytics in the Rho architecture.

```
1  Inputs: Model $p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\beta}_0,\phi_0)$, streaming datasets $D_1$,...,$D_b$,... ;
2  Outputs: $\tilde{\boldsymbol{\beta}}_b$ and $\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b)$, for $b=1,2,\dots$ ;
3  Initialize: Set initial values $\tilde{\boldsymbol{\beta}}_{\mathrm{init}}$, $\tilde{\phi}_0=0$ and $\tilde{\boldsymbol{J}}_0=\boldsymbol{0}_{p\times p}$ ;
4  for $b=1,2,\dots$ do
5      Read in dataset $D_b$ ;
6      At the inference layer, perform Cholesky decomposition of $\left\{\tilde{\boldsymbol{J}}_{b-1}+\boldsymbol{J}_b(D_b;\tilde{\boldsymbol{\beta}}_{b-1})\right\}$
         and cache the resulting factorizations ;
7      At the speed layer, with $\tilde{\boldsymbol{\beta}}_b^{(1)}=\tilde{\boldsymbol{\beta}}_{b-1}$, use the factorizations to run the following iterations
         $\tilde{\boldsymbol{\beta}}_b^{(r+1)}=\tilde{\boldsymbol{\beta}}_b^{(r)}+\left\{\tilde{\boldsymbol{J}}_{b-1}+\boldsymbol{J}_b(D_b;\tilde{\boldsymbol{\beta}}_{b-1})\right\}^{-1}\left\{\tilde{\boldsymbol{J}}_{b-1}(\tilde{\boldsymbol{\beta}}_{b-1}-\tilde{\boldsymbol{\beta}}_b^{(r)})+\boldsymbol{U}_b(D_b;\tilde{\boldsymbol{\beta}}_b^{(r)})\right\}$,
         until convergence ;
8      At the inference layer, update both $\tilde{\boldsymbol{J}}_b=\tilde{\boldsymbol{J}}_{b-1}+\boldsymbol{J}_b(D_b;\tilde{\boldsymbol{\beta}}_b)$ and
         $\tilde{\phi}_b=\frac{N_{b-1}-p}{N_b-p}\tilde{\phi}_{b-1}+\frac{n_b-p}{N_b-p}\hat{\phi}_b$, and then calculate $\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b)=\tilde{\phi}_b\tilde{\boldsymbol{J}}_b^{-1}$ ;
9      Save $\tilde{\boldsymbol{\beta}}_b$ at the speed layer, $\tilde{\boldsymbol{J}}_b$ and $\tilde{\phi}_b$ at the inference layers ;
10     Release data set $D_b$ from the memory ;
11 end
12 Return $\tilde{\boldsymbol{\beta}}_b$ and $\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b)$, for $b=1,2,\dots$.
```

(a) Line 1: all streaming datasets are modeled by a homogeneous GLM with a common true parameter $\boldsymbol{\beta}_0$. Such model automatically satisfies some of the regularity conditions given in Section 4.1, such as condition (C3).

(b) Line 2: outputs include renewable estimates of $\boldsymbol{\beta}$ and estimated asymptotic variances at each batch $b$.

(c) Line 3: set certain initial values for $\boldsymbol{\beta}_0$, *e.g.*, $\tilde{\boldsymbol{\beta}}_{\mathrm{init}}=\boldsymbol{0}$.

(d) Line 4: run through the online updating procedures along data streams.

(e) Line 6: at the inference layer, calculate the negative Hessian $\boldsymbol{J}_b(D_b;\tilde{\boldsymbol{\beta}}_{b-1})$ and communicate with the speed layer.

(f) Line 7: run the updating algorithm to renew $\tilde{\boldsymbol{\beta}}_{b-1}$ to $\tilde{\boldsymbol{\beta}}_b$, in which the cached factorizations are repetitively used in iterations.

(g) Line 8: at the inference layer, update both negative Hessian and dispersion parameter estimate with current batch $D_b$ under newly updated $\tilde{\boldsymbol{\beta}}_b$ from the speed layer.

### *5.2. Examples*

Unlike the first example of the Gaussian linear model in Section 3.3 where an exact decomposition of data batches is available, here we present two nonlinear GLMs in that the proposed renewable analytics are needed. They are popular logistic model for binary outcomes and log-linear model for count outcomes.

Example 2. *(Logistic model). Assume data batch $D_b=\{\boldsymbol{y}_b,\boldsymbol{X}_b\}$ with binary outcomes $\boldsymbol{y}_b=(y_{b1},\dots,y_{bn_b})^T$ and covariates $\boldsymbol{X}_b=(\boldsymbol{x}_{b1},\dots,\boldsymbol{x}_{bn_b})^T$, where $y_{bi}|\boldsymbol{x}_{bi}$ are independently sampled from a Bernoulli distribution with probability of success $\pi_{bi}=P(y_{bi}=1\mid\boldsymbol{x}_{bi})$, and the dispersion parameter $\phi=1$. A logistic model takes the form $g(\pi_{bi})=\log(\frac{\pi_{bi}}{1-\pi_{bi}})=\boldsymbol{x}_{bi}^T\boldsymbol{\beta}$. The score function and negative Hessian matrix (or the observed information matrix) for data batch $D_b$ are respectively given by*

$$\boldsymbol{U}_b(\boldsymbol{\beta})=\sum_{i=1}^{n_b}\boldsymbol{x}_{bi}\left\{y_{bi}-\frac{\exp(\boldsymbol{x}_{bi}^T\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_{bi}^T\boldsymbol{\beta})}\right\},\quad \textit{and}\quad \boldsymbol{J}_b(\boldsymbol{\beta})=\sum_{i=1}^{n_b}v_{bi}\boldsymbol{x}_{bi}\boldsymbol{x}_{bi}^T,$$

*where $v_{bi}(\pi_{bi})=\pi_{bi}(1-\pi_{bi})=\frac{\exp(\boldsymbol{x}_{bi}^T\boldsymbol{\beta})}{\{1+\exp(\boldsymbol{x}_{bi}^T\boldsymbol{\beta})\}^2}$ is the variance function. The renewable estimate*

*$\tilde{\boldsymbol{\beta}}_b$ and the aggregated observed information matrices $\tilde{\boldsymbol{J}}_b$ are updated according to the procedure given in Algorithm 1 under the Rho architecture in Figure 3.*

EXAMPLE 3. *(Poisson log-linear model). Consider $D_b = \{\boldsymbol{y}_b, \boldsymbol{X}_b\}$ with outcomes of counts $\boldsymbol{y}_b = (y_{b1}, \ldots, y_{bn_b})^T$ and covariates $\boldsymbol{X}_b = (\boldsymbol{x}_{b1}, \ldots, \boldsymbol{x}_{bn_b})^T$. Assume $y_{bi}|\boldsymbol{x}_{bi}$ are independently sampled from a Poisson distribution with mean $\mu_{bi} = \mathbb{E}(y_{bi}|\boldsymbol{x}_{bi})$ that is specified by a log-linear model $g(\mu_{bi}) = \log(\mu_{bi}) = \boldsymbol{x}_{bi}^T\boldsymbol{\beta}$. Here the dispersion parameter $\phi = 1$. The score function and negative Hessian matrix (or the observed information matrix) for data batch $D_b$ are given by, respectively, $\boldsymbol{U}_b(\boldsymbol{\beta}) = \sum_{i=1}^{n_b} \boldsymbol{x}_{bi}\left\{y_{bi} - \exp(\boldsymbol{x}_{bi}^T\boldsymbol{\beta})\right\}$, and $\boldsymbol{J}_b(\boldsymbol{\beta}) = \sum_{i=1}^{n_b} v_{bi}\boldsymbol{x}_{bi}\boldsymbol{x}_{bi}^T$, where $v_{bi} = \mu_{bi} = \exp(\boldsymbol{x}_{bi}^T\boldsymbol{\beta})$ is the variance function. Again, the renewable estimate $\tilde{\boldsymbol{\beta}}_b$ and the aggregated observed information matrices $\tilde{\boldsymbol{J}}_b$ are produced in the Rho architecture (Figure 3), respectively, at the speed layer and the inference layer via Algorithm 1.*

## 6. Simulation Experiments

### *6.1. Setup*

We conduct simulation experiments to assess the performance of the proposed renewable estimator and incremental inference in the settings of linear and logistic models. We compare our method with several leading methods in the current literature. They are (i) the oracle MLE obtained by processing the entire data once, (ii) AI-SGD, (iii) sequential estimation method of online LSE in the linear model, and (iv) sequential estimation method of CEE/CUEE for nonlinear GLMs.

Comparisons concern the aspects of parameter estimation, computational efficiency, and hypothesis testing. The evaluation criteria for parameter estimation include (a) absolute bias (A.bias), (b) averaged estimated standard error (ASE), (c) empirical standard error (ESE) and (d) coverage probability (CP). We use the MLE yielded from the R package `glm` as the gold standard in all comparisons. For AI-SGD method, we use the R package `sgd` with one-dimensional learning rate (Xu, 2011) and hyper-parameters, respectively, set at $\alpha = 1$, $\gamma_0 = 1$ and $c = 2/3$. Following Fang (2019), we set $S = 200$ bootstrap samples. Computational efficiency is also assessed by (e) computation time (C.Time) and (f) running time (R.Time). R.time accounts only for the data processing time, while C.time includes time spent on both loading data streams and processing data. Note that in the case of AI-SGD, one data point is run at one iteration, thus it is hard to capture the data loading time properly. In this case, we consider only R.time for AI-SGD.

In all the simulation experiments considered in Tables 1-3, we set a terminal point $B$. We generate the full dataset $D_B^\star$ with $N_B$ observations independently from the respective GLMs with the mean model $\mathbb{E}(y_i|\boldsymbol{x}_i) = g(\boldsymbol{x}_i^T\boldsymbol{\beta}_0)$, $i = 1, ..., N_B$. We set $\boldsymbol{\beta}_0 = (0.2, -0.2, 0.2, -0.2, 0.2)^T$, the intercept $\boldsymbol{x}_{i[1]} \equiv 1$, and $\boldsymbol{x}_{i[2:5]} \sim \mathcal{N}_4(\boldsymbol{0}, \boldsymbol{V}_4)$ independently where $\boldsymbol{V}_4$ being a $4 \times 4$ compound symmetry covariance matrix with correlation $\rho = 0.5$.

### *6.2. Evaluation of Parameter Estimation*

**Scenario 1: fixed $N_B$ but varying batch size $n_b$**

We begin with the comparison of four methods for the effect of data batch size $n_b$ on their performances of point estimation and computational efficiency. These methods include (a) MLE, (b) AI-SGD, (c) online LSE for the linear model, or CEE/CUEE for the logistic model, and (d) renewable estimation (Renew). We generate $B$ data streams consisting of $N_B = |D_B^\star| = 100,000$ independent observations, each batch with $n_b$ observations. Tables 1 and 2 report the evaluation criteria for the linear and logistic models, respectively, over 500 rounds of simulations. Additional simulation results in the linear, logistic and log-linear models with other varying batch sizes may be found in Tables S1, S2 and S3, respectively, in the Supplementary Material section S3.

**Bias and coverage probability.** In the linear model, due to the fact that the LSE is a linear function of data, it can be perfectly decomposed across data batches. Thus, MLE, online LSE

Table 1: Simulation results under the linear model are summarized over 500 replications, with fixed $N_B = 100,000$ and $p = 5$ with varying batch sizes $n_b$. "A.bias", "ASE", "ESE" and "CP" stand for the mean absolute bias, the averaged estimated standard error of the estimates, the empirical standard error, and the coverage probability, respectively. "C.Time" and "R.Time" respectively denote computation time and running time, and the unit of both is second.

| | AI-SGD | MLE | | | Online LSE | | | Renew | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $n_b$ | 1 | 1000 | 200 | 50 | 1000 | 200 | 50 | 1000 | 200 | 50 |
| A.bias$\times 10^{-3}$ | **13.48** | 3.17 | 3.17 | 3.17 | 3.17 | 3.17 | 3.17 | 3.17 | 3.17 | 3.17 |
| ASE$\times 10^{-3}$ | **15.08** | 3.83 | 3.83 | 3.83 | 3.83 | 3.83 | 3.83 | 3.83 | 3.83 | 3.83 |
| ESE$\times 10^{-3}$ | 17.24 | 3.94 | 3.94 | 3.94 | 3.94 | 3.94 | 3.94 | 3.94 | 3.94 | 3.94 |
| CP | 0.92 | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 |
| C.Time(s) | - | 0.56 | 1.68 | 5.91 | 0.08 | 0.19 | 0.66 | 0.12 | 0.34 | 1.27 |
| R.Time(s) | 0.14 | 0.32 | 0.30 | 0.29 | 0.02 | 0.07 | 0.28 | 0.07 | 0.24 | 0.95 |

Table 2: Simulation results, summarized from 500 replications, under the setting of $N_B = 100,000$ and $p = 5$ for the logistic model with varying batch size $n_b$.

| | AI-SGD | MLE | | CEE | | | CUEE | | | Renew | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $n_b$ | 1 | $10^5$ | 1000 | 200 | 50 | 1000 | 200 | 50 | 1000 | 200 | 50 |
| A.bias$\times 10^{-3}$ | **24.98** | 6.31 | 6.40 | **8.31** | **24.50** | 6.34 | 6.89 | **11.98** | 6.32 | 6.32 | 6.32 |
| ASE$\times 10^{-3}$ | **27.10** | 7.82 | 7.84 | 7.94 | 8.34 | 7.83 | 7.86 | 7.94 | 7.82 | 7.82 | 7.82 |
| ESE$\times 10^{-3}$ | 31.14 | 7.93 | 7.88 | 7.67 | 7.02 | 7.93 | 8.43 | **15.64** | 7.92 | 7.93 | 7.92 |
| CP | 0.92 | 0.95 | 0.94 | **0.88** | **0.12** | 0.95 | 0.92 | **0.74** | 0.95 | 0.95 | 0.95 |

and Renew are identical, leading to exactly the same bias and coverage probability, as shown in Table 1. It is easy to see that both bias and coverage probability in the linear model are not affected by data batch size $n_b$. From Table 2 with the logistic regression, our renewable estimation always exhibits similar performances to the oracle MLE, and appears quite robust to different $n_b$. In contrast, CEE appears numerically unstable; as batch size $n_b$ decreases to 200, its coverage probability drops down below 90%. Even though the CUEE is proposed to improve the CEE (Schifano et al., 2016), bias of CUEE appears much larger than that of the MLE as $n_b$ decreases to 50. In addition, CUEE has much larger empirical standard error than that of CEE as $n_b$ gets smaller. AI-SGD processes a single observation each time. So, its bias, estimated and empirical standard errors are not related to $n_b$, but all of them are constantly larger than those of the MLE or our renewable estimation. Even though the coverage probability of AI-SGD by the Fang's method is 0.92, close to the nominal level 0.95, it does not seem to be efficient as it has much larger standard errors than the MLE and the renewable method. See also the supplementary Tables S1 to S3.

**Computation time.** Two metrics are used to evaluate computational efficiency. "C.Time" in Tables 1 (see also in the supplementary Tables S2 and S3) refers to the total amount of time required by data loading and algorithm execution, while "R.Time" is the amount of time required only for algorithm execution. With an increased $B$, our renewable estimation method clearly outperforms the three competitors, MLE, CEE and CUEE. AI-SGD appears computationally very competitive, due to the fact that it avoids matrix inversion calculation in the algorithm. However, this high computing speed pays the price to significantly big estimation bias, leading to problematic statistical inference. As pointed out above, we are not able to evaluate data loading time for AI-SGD, since it passes one single data point at a time. The supplementary Figure S1 presents a pictorial summary of all the results obtained in the simulation scenario 1.

**Scenario 2: fixed batch size $n_b$ but varying $B$**

Now we turn to an interesting scenario where streaming datasets arrive at a high speed. For convenience, we fix batch size $n_b = 100$, but let $N_B$ increase from $10^3$ to $10^6$. Table 3 lists the summaries of simulation results under the logistic model.

**Bias and coverage probability.** When the batch size is as small as $n_b = 100$, increasing

Table 3: Comparison among different estimators in the logistic model with fixed batch size $n_b = 100$ and $p = 5$, $N_B$ increases from $10^3$ to $10^6$. Results are summarized from 500 replications.

| | $B = 10, N_B = 10^3$ | | | | | $B = 100, N_B = 10^4$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | MLE | AI-SGD | CEE | CUEE | Renew | MLE | AI-SGD | CEE | CUEE | Renew |
| A.bias$\times 10^{-3}$ | 61.59 | 63.18 | 58.71 | 60.78 | 60.97 | 19.59 | 24.14 | 20.80 | 19.93 | 19.55 |
| ASE$\times 10^{-3}$ | 78.70 | 58.34 | 81.07 | 79.38 | 79.15 | 24.73 | 28.40 | 25.53 | 24.93 | 24.76 |
| ESE$\times 10^{-3}$ | 77.32 | 78.63 | 73.05 | 76.30 | 76.56 | 24.50 | 30.23 | 22.99 | 24.81 | 24.44 |
| CP | 0.96 | **0.83** | 0.97 | 0.96 | 0.96 | 0.95 | **0.92** | 0.95 | 0.95 | 0.95 |
| C.Time(s) | 0.01 | - | 0.03 | 0.06 | 0.01 | 0.08 | - | 0.34 | 0.63 | 0.07 |
| R.Time(s) | 0.007 | 0.008 | 0.028 | 0.056 | 0.006 | 0.045 | 0.064 | 0.311 | 0.599 | 0.047 |
| | $B = 10^3, N_B = 10^5$ | | | | | $B = 10^4, N_B = 10^6$ | | | | |
| | MLE | AI-SGD | CEE | CUEE | Renew | MLE | AI-SGD | CEE | CUEE | Renew |
| A.bias$\times 10^{-3}$ | 6.23 | **23.44** | **12.63** | **7.66** | 6.22 | 1.92 | **23.44** | **12.43** | **4.67** | 1.92 |
| ASE$\times 10^{-3}$ | 7.82 | **27.94** | 8.07 | 7.88 | 7.82 | 2.47 | **27.94** | 2.55 | 2.49 | 2.47 |
| ESE$\times 10^{-3}$ | 7.78 | 29.39 | 7.31 | 9.42 | 7.78 | 2.42 | 29.39 | 2.28 | 5.98 | 2.42 |
| CP | 0.95 | 0.94 | **0.68** | **0.90** | 0.95 | 0.95 | 0.94 | **0** | **0.67** | 0.95 |
| C.Time(s) | 2.88 | - | 3.056 | 5.74 | 0.64 | 343.5 | - | 32.60 | 56.51 | 6.46 |
| R.Time(s) | 0.51 | 0.19 | 2.84 | 5.50 | 0.47 | 7.04 | 0.98 | 28.85 | 54.04 | 4.66 |

$N_B$ does not seem to help reduce the estimation bias of CEE or CUEE. In effect, their bias exacerbates as more data streams are processed, resulting in clearly problematic performances on statistical inference. When the number of data batches $B$ increases to 1000, the coverage probability by CEE or CUEE remains steadily below 90%, with no sign of improvement in response to increased volumes of data. It is striking to notice that when $B$ is further increased to $10^4$, the coverage probability of CUEE falls down to 67%, while CEE gives the worst 0% coverage probability. This confirms that when the condition $B = \mathcal{O}(n_j^k)$, $k < 1/3$, is violated, CEE/CUEE will not have valid asymptotic distributions for inference. In contrast, our proposed method confirms the large sample properties similar to those of the oracle MLE: the average absolute bias decreases rapidly as the total sample size accumulates, and the coverage probability stays robustly around 95%. For competitor AI-SGD, the estimated standard error is much smaller than the empirical one and coverage probability is only 83% when $N_B = 10^3$. When $N_B$ reaches $10^5$ the coverage probability gets improved to be around 95%. However, both bias and estimated standard errors are much larger than those of MLE or our renewable method, suggesting that AI-SGD does not provide an efficient inference. Moreover, its bias stops decreasing after a certain level. For example, its bias remains at $23.44 \times 10^{-3}$ when $N_B$ increases from $10^5$ to $10^6$ with no sign of further improvement. A similar phenomenon has been reported in the literature. According to Toulis and Airoldi (2015), once AI-SGD reaches a convergence phase, the subsequent estimates will jitter around the true parameter within a ball of slowly decreasing radius.

**Computation time.** Our renewable estimation method shows clearly advantageous as $N_B$ increases: the combined amount of time for data loading and algorithm execution only takes less than 10 seconds, whereas the oracle MLE, when processing a total of $10^6$ samples once, requires more than 5 minutes. This 35-fold faster computation by the proposed method does not sacrifice any estimation precision and inference power. In addition, the running time for our method and AI-SGD are comparable even under large sample size settings such as $N_B = 10^5$ and $10^6$. Once again, AI-SGD produces much larger bias and standard errors than our method. The extra small amount of time used by our method on updating info.mats at the inference layer is computationally worthwhile for achieving a valid and efficient statistical inference.

**Scenario 3: large $p$ with fixed $N_B$ and $B$**

To examine the scalability of our method when $p$ becomes large, we run simulations with $p = 1000, 2500$, in the logistic model. We set $N_B = 2 \times 10^5$, $B = 20$ and $n_b = 10^4$, and simulate $p$-element vectors of covariates from $\boldsymbol{x}_i \overset{i.i.d.}{\sim} \mathcal{N}(0, N_B^{-1}\boldsymbol{I}_p)$. Following Sur and Candés (2018), in order to guarantee the existence of MLE in such high dimensional setting, we generate the true values of $\boldsymbol{\beta}_0$ entrywise *i.i.d.* from $\mathcal{N}(10, 900)$ under $p = 1000$ and from $\mathcal{N}(10, 300)$ under $p = 2500$, respectively. The same criteria are used in the subsequent assessment and comparisons.

**Bias and coverage probability.** Table 4 summarizes the simulation results over 200 replications. Our renewable method has the same level of bias as the oracle MLE in this high-dimensional logistic regression. In this setting with $n_j \leq 10p$, both CEE and CUEE fail to provide reliable coverage probabilities due to severely large biases. AI-SGD has the largest bias, more than 10 times that of MLE, due largely to the fact that the AI-SGD updates may get trapped locally. Consequently, standard errors are not properly estimated by the Fang's perturbation resampling method, resulting in 0% coverage probability. According to Fang (2019), the resampling method may not be able to deal with high-dimensional large-scale data.

**Computation time.** For large $p = 1000$ or 2500, our renewable estimation method is at least 4-fold faster than the oracle MLE, and this computational efficiency gain repeats in the low dimension case ($p = 5$) shown in Table 3. Although AI-SGD runs faster than our renewable method, it is not applicable to the setting with very large $p$. The resulting severe bias hampers from producing any reliable estimation or valid inference.

Table 4: Comparisons among different estimators in the logistic model with fixed $N_B = 2 \times 10^5$, $n_b = 10^4$ and $B = 20$. The number of covariates, $p$, varies from 1000 to 2500.

| | | | $p = 1000$ | | |
|---|---|---|---|---|---|
| | AI-SGD | MLE | CEE | CUEE | Renew |
| A.bias | 25.799 | 2.176 | 3.880 | 2.242 | 2.152 |
| ASE | $1.70 \times 10^{-3}$ | 2.705 | 2.904 | 2.668 | 2.707 |
| ESE | $1.72 \times 10^{-3}$ | 2.715 | 2.358 | 2.616 | 2.673 |
| CP | **0** | 0.948 | 0.757 | 0.937 | 0.951 |
| C.Time(min) | - | 17.959 | 17.288 | 20.470 | 4.207 |
| R.Time(min) | 1.609 | 16.686 | 17.093 | 20.258 | 4.014 |
| | | | $p = 2500$ | | |
| | AI-SGD | MLE | CEE | CUEE | Renew |
| A.bias | 16.386 | 2.212 | 6.994 | 2.581 | 2.192 |
| ASE | $1.71 \times 10^{-3}$ | 2.728 | 3.475 | 2.523 | 2.789 |
| ESE | $1.72 \times 10^{-3}$ | 2.745 | 1.804 | 2.442 | 2.715 |
| CP | **0** | 0.946 | 0.561 | 0.874 | 0.954 |
| C.Time(min) | - | 126.407 | 122.528 | 149.411 | 31.451 |
| R.Time(min) | 4.737 | 123.904 | 122.037 | 148.924 | 30.917 |

In summary, the above simulation results clearly suggest that our proposed method can produce realtime robust and reliable estimation and inference. Its performances seen in the simulation studies are very similar to the oracle MLE that processes the entire data once, regardless of low or high dimension $p$, and regardless of volume and speed of streaming data. In contrast, we find that the existing online methods can only work in some cases. For example, AI-SGD only gives proper coverage probability when $B$ is large and $p$ is small, while CEE/CUEE produces valid inference when both $B$ and $p$ are small. Such evidence further demonstrates the usefulness of our method in interim analyses over the course of data streams. As far as computational efficiency concerns, the proposed method is clearly superior over existing methods when data streams arrive at a high speed. Note that the running time complexity of our method is $O\left(N_B p^2 + B p^3/3\right)$. When $p < n_b$, it reduces to $O(N_B p^2)$. This is a typical order of a second-order online method. When $N_B$ is fixed and $p$ is large, increasing batch size $n_b$ makes $B$ small, leading to a potential improvement in computational efficiency. This gain of computing speed has been repeatedly seen in both Tables 1 and 4, as well as the supplementary Tables S1-S3.

### 6.3. Evaluation of Hypothesis Testing

Now we evaluate the performance of the proposed incremental inference based on the Wald test available at the inference layer in the Rho architecture. We run a simulation study on the Wald test for $H_0 : \beta_{01} = 0.2$ vs. $H_A : \beta_{01} \neq 0.2$, where $\beta_{01}$ is the intercept parameter in the logistic model used in Tables 2 and 3. With the $\boldsymbol{\beta}^{\text{null}} = (0.2, -0.2, 0.2, -0.2, 0.2)^T$, set $\boldsymbol{\beta}_a = (\beta_{a1}, -0.2, 0.2, -0.2, 0.2)^T$ with $\beta_{a1}$ chosen to be a sequence of values from 0.205 to 0.250 with an increment of 0.005. We evaluate both the size (or type I error) and power (1−type II error) of the Wald test in equation (13) proposed in Section 4.2. Based on simulated data streams, with $N_B = 100,000$, each batch size $n_b = 200$, we calculated empirical type I error and power from 500 replications.

Under $H_0$, as shown in the (1,1)-th panel of Figure S2 in the Supplementary Material, the Q-Q plot of 500 replicates of the Wald test statistic stay closely along the 45° diagonal, indicating the validity of asymptotic $\chi^2_1$ distribution. In addition, we increase the number of coefficients in the test, and find that under $H_0$ the Wald statistics all behave approximately under the chi-square distribution; see the other plots of Figure S2. Supplementary Table S4 reports the empirical type I errors and power based on 500 replications, where the type I errors of the Wald test for $H_0 : \beta_{01} = 0.2$ by the MLE, AI-SGD and our proposed Wald test are very close to the nominal level of 0.05, while the Wald tests based on CEE and CUEE have poor type I error control. Figure 4 shows that the power of AI-SGD is steadily significantly lower than that of the proposed incremental Wald test or the MLE. In addition, CEE or CUEE has lower power when the parameter is to the true value 0.2, suggesting a poor local power.

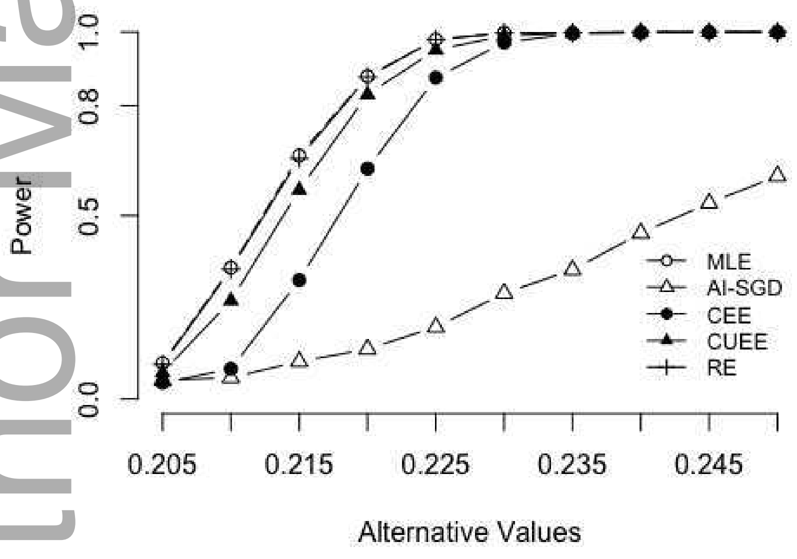

**Fig. 4.** Power curves of the Wald tests based on MLE, AI-SGD, CEE, CUEE and renewable estimation, under a sequence of alternative values for the intercept $\beta_1$.

## 7. Data Example

To show the usefulness of our proposed renewable estimation and inference in practice, we analyzed streaming data from the National Automotive Sampling System-Crashworthiness Data System (NASS CDS). Our primary interest was to evaluate the effectiveness of graduated driver licensing (GDL), which is a nationwide legislature on novice drivers of age 21 or younger under various conditions of vehicle operation. On the other hand, there are no restrictions on vehicle operation for older drivers (say, Age$\geq$ 65) in the current law. To assess the effect of driver's age on driving safety, we compared age groups with respect to the risk of fatal crash when an accident occurred. Three age groups were considered: "Age$<$ 21", "21 $\leq$Age $<$ 65", "Age$\geq$ 65" were coded as dummy variables in our analysis, with the middle age group as the reference. Since the number of young or old drivers involved in accidents was much smaller than those in the reference group, it was of interest to renew analysis results with more data being collected sequentially over time. Event of "Fatality" in a crash is a binary outcome of interest, which was analyzed using a logistic model. This outcome variable was created from the variable of Maximum Treatment in Accident (ATREAT) in the database, which indicated the most intensive treatment given to driver in an accident.

In this example, streaming data were formed by monthly accident data from the period of 7 years over January, 2009 to December, 2015, with $B = 84$ data batches and a total sample size $N_B = 23,184$ of recorded accidents in USA. We applied our proposed method to sequentially update parameter estimates and standard errors for the regression coefficients. We assumed the underlying risk of fatal crash across age groups was constant over the 7-year time window. Six additional confounding factors were included in the logistic model, including, Sex, Seat Belt Use, Light condition and Speed Limit.

As shown in Figure 5, the 95% pointwise confidence bands over the 84 batches became narrower for all regression coefficients as more data streams arrived. The top two panels display the traces plots of renewable estimates of the coefficients for young and old groups, respectively. The estimates for the young group stay below 0 over the 84-month period, meaning that the young group (Age$<$ 21) has lower adjusted odds of fatal crash than the reference group. This finding is consistent with the reported results in the literature that GDL is an effective policy to protect novice drivers from severe injuring (e.g. Chen et al. (2014)). In contrast, the trace plot for the old age group (Age$\geq$ 65) shows an upward trend and get stabilized when the sample size increases. This suggests that the adjusted odds of fatality in a vehicle crash for the old group becomes significant higher than the reference group when data accumulated large enough. This may suggest a need on policy modification on restrictive vehicle operation for old drivers.

Figure 6 shows the trends of $-\log_{10}(p)$, $p$-values of the incremental Wald test in the 10-base logarithm, for each regression coefficient over 84 months. Clearly, all the evidence against the null $H_0 : \beta_j = 0$ increases over time. "Seat Belt" turns out to have the strongest association to the odds of fatality in a crash among all covariates included in the model. This is an overwhelming confirmation to the enforcement of policy "buckle up" when sitting in a moving vehicle. In addition, to characterize the overall significance level for each covariate over the 84-month period, we proposed to calculate a summary statistic as of area under the $p$-value curve. Most of these curves have well separated patterns, so that the ranking of the overall significance by the calculated areas is well aligned with the ranking of $p$-values obtained at the end time of streaming data availability, namely December, 2015. It is interesting to note that "Traffic Control Function", "Light Condition" and "Sex" are among the weakest predictors.

Applying the proposed renewable estimation and inference to the above CDS data analysis enabled us to visualize time-course patterns of data evidence accrual as well as stability and reproducibility of inference. As shown clearly in Figure 5, at the early stage of data streams, due to limited sample sizes and possibly sampling bias, both parameter estimates and test power may be unstable and even misleading. These potential shortcomings can be convincingly overcome when estimates and inferential quantities were continuously updated along with data streams, which eventually reached stability and reliable conclusions. Table 5 reports the related analysis at the terminal time of these streaming data. Our proposed Rho architecture has made the above incremental analysis straightforward. As a matter of fact, this expanded architecture with an addition of inference layer has given rise to tremendous convenience in data storage and data analytics for processing high-throughput streaming data.

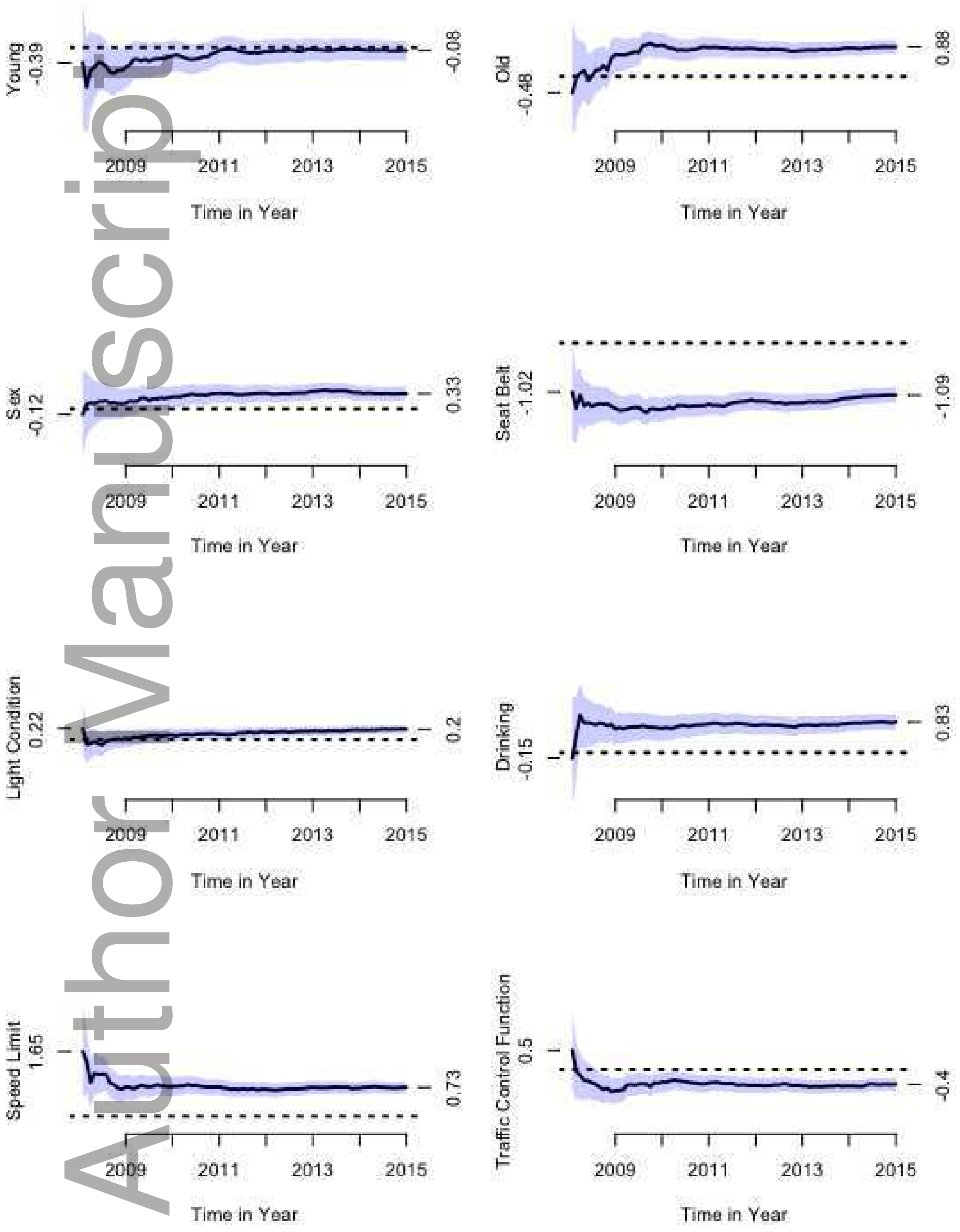

**Fig. 5.** Trace plot for the coefficients estimates and $95\%$ pointwise confidence bands of regression coefficients. Numerical numbers on two sides denote the estimated regression coefficients after the arrival of first and last batches. The dashed line is the "0" reference line.

Table 5: Results from the MLE method and the proposed renewable estimation method in logistic model with $N = 23,184, p = 9, B = 84$.

| | MLE | | | Renew | | |
|---|---|---|---|---|---|---|
| | Estimate | ASE | $p$-value | Estimate | ASE | $p$-value |
| Intercept | -4.284 | 0.174 | $3.91 \times 10^{-134}$ | -4.254 | 0.169 | $6.18 \times 10^{-140}$ |
| Young | -0.081 | 0.127 | 0.524 | -0.080 | 0.132 | 0.541 |
| Old | 0.889 | 0.104 | $1.16 \times 10^{-17}$ | 0.876 | 0.105 | $9.99 \times 10^{-17}$ |
| Sex | 0.343 | 0.079 | $1.60 \times 10^{-5}$ | 0.326 | 0.077 | $2.32 \times 10^{-5}$ |
| Seat Belt | -1.080 | 0.084 | $3.55 \times 10^{-38}$ | -1.085 | 0.081 | $2.87 \times 10^{-41}$ |
| Light Condition | 0.208 | 0.042 | $7.25 \times 10^{-7}$ | 0.202 | 0.042 | $1.24 \times 10^{-6}$ |
| Drinking | 0.835 | 0.106 | $2.42 \times 10^{-15}$ | 0.833 | 0.108 | $1.33 \times 10^{-14}$ |
| Speed Limit | 0.719 | 0.078 | $2.94 \times 10^{-20}$ | 0.734 | 0.077 | $2.19 \times 10^{-21}$ |
| Traffic Control Function | -0.414 | 0.085 | $1.18 \times 10^{-6}$ | -0.397 | 0.084 | $2.09 \times 10^{-6}$ |

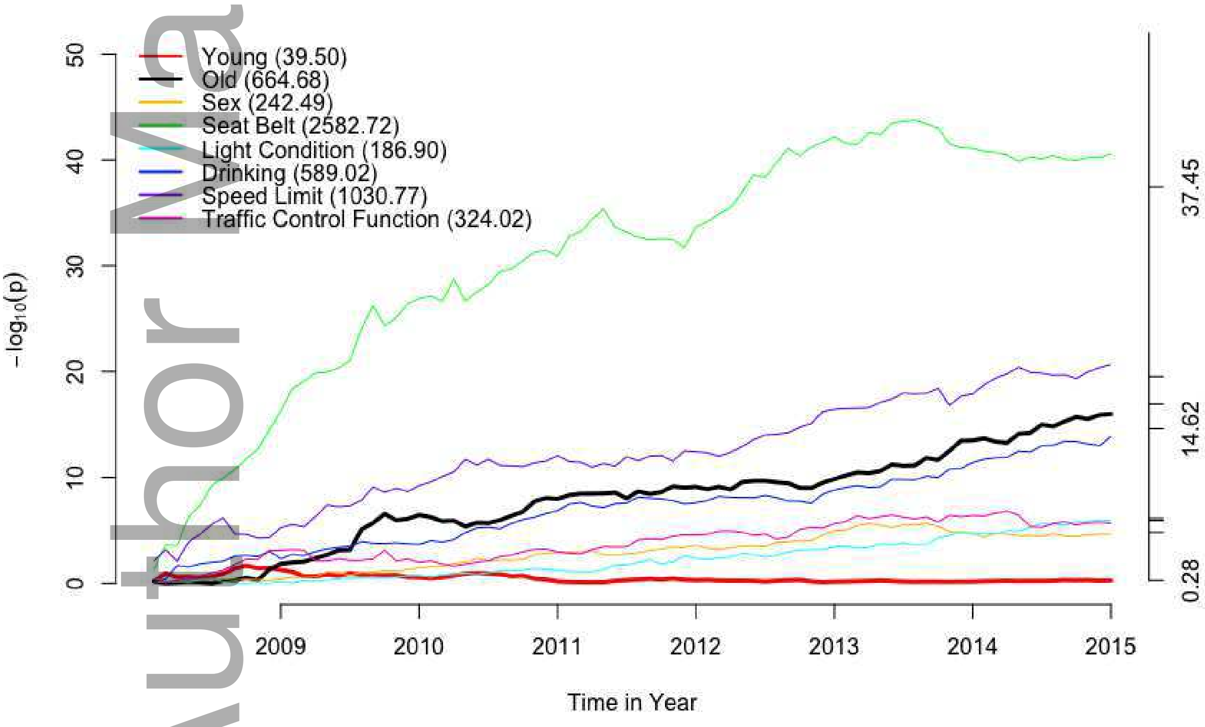

**Fig. 6.** Trace plot of $-log_{10}(p)$ over monthly data batches during January, 2009 to December, 2015, each for one regression coefficient. Numbers on the left y-axis are the negative logarithm $p$-values obtained by the proposed incremental Wald test and labels on the x-axis correspond to the last month of each year. On the right y-axis, the numerical numbers denote $-log_{10}(p)$ obtained by the oracle MLE. The values in the brackets next to covariate names denote respective areas under the $p$-value curves.

## 8. Concluding Remarks

Although a large number of statistical methods and computational recipes have been developed to address various challenges for Big Data analytics, such as the subsampling-based methods (Liang et al., 2013; Kleiner et al., 2014; Ma et al., 2015) and divide-and-conquer techniques (Lin and Xi, 2011; Guha et al., 2012; Chen and Xie, 2014; Tang et al., 2016; Zhou and Song, 2017), little is known about statistical inference in streaming data analyses under dynamic data storage and incremental updates. This paper has filled the gap with the proposed renewable estimation and incremental inference.

The renewable methodology is based primarily on a second-order approximation to the oracle MLE. It can sequentially renew both point estimation and asymptotic normality along data streams. We proposed a Rho architecture for implementation as an extension to the Apache Spark Lambda architecture, which adds an inference layer to carry out storage and updating of information matrices. Both proposed statistical methodology and computational algorithms have been justified theoretically and examined numerically in the setting of generalized linear models. Being a key methodology contribution, the incremental inference has shown to be statistically valid and efficient. It has no loss of estimation efficiency in comparison to the oracle MLE method, but is computationally much more efficient than the MLE.

Summary statistics involved in our proposed renewable framework behave similarly as the classical sufficient statistic does. Appendix A.6 presents an extension of the classical concept of sufficiency in this setting of renewable analytics, where only summary statistics of historical data are accessible. The proposed approximate sufficiency enables us to explain the renewable properties in terms of the sufficient statistic. This extension builds a useful theoretical connection between the classical theory of statistical sufficiency and modern online learning analytics. More details on the technical proofs are included in the Supplementary Material section S2.

Through various simulation studies, we demonstrate that our proposed method runs computationally faster than two existing methods CEE and CUEE. Our updating algorithm keeps using the same inversed Hessian matrix over all iterations, which is only computed once per data batch. It is worth pointing out once again that the estimation consistency of CEE or CUEE is established under a strong regularity condition concerning the ratio of batch size $n_b$ to the number of data batches $B$. Such condition may not hold in some real applications when data streams arrive perpetually. Our method has overcome this restriction and produces stable, reliable, efficient solutions to the three questions raised in the introduction section. Thus, our method is practically appealing. Reliability of statistical inference is of great importance in practice to handle data streams, such as Phase IV clinical trials where drug safety, side-effect and efficacy have to be assessed at the general population mobile health data analysis, as well as traditional sensor networks, web logs and computer network traffic (Gaber et al., 2005).

The proposed renewable analytics may be treated as a competitive alternative to currently popular parallel computation. Allocating memory has become a main focus in the development of Big Data analytics. The crucial technical challenge pertains to whether or not historical raw data, instead of summary statistics, are needed in iterative updates to search for the MLE. Some `R` packages such as `biglm` (Lumley, 2013) and `speedglm` (Enea et al., 2015) are proposed to address the problem of loading a large data set, and they have been shown to provide exactly the same results as the MLE from the R package `glm`. Both `biglm` and `speedglm` avoid reading in the entire big data set at once; instead calculating the needed sufficient statistics, $X^T W X$ and $XWZ$, in sequential increments and then summing them up in the Iterative Weighted Least Square (IWLS) algorithm. However, these two methods must use historical subject-level data in calculations. Thus, they are more expensive in data storage and computationally inefficient in comparison to our proposed method. From this perspective, our method could also serve as a powerful alternative to `biglm` and `speedglm`, and as well as to the parallel computing paradigm when analyzing very large static data.

The formulation of renewable analytics is under the context of generalized linear models where the log-likelihood functions have nice properties such as the twice continuous differentiability. Both theoretical and numerical experiences learned from the GLMs in this paper shed light on further generalization of such method to other important settings such as generalized estimating equations (GEEs), Cox regression, and quantile regression. In addition, our method is based on the assumption that data batches are all sampled from a homogeneous study population, which may be violated in some of practical studies. In this case of heterogeneous data streams, sequential updating procedures will be a challenging but useful methodology research topic, which is worth further exploration.

## Acknowledgements

The authors are grateful to editor, associate editor and reviewer for their constructive suggestions and insightful comments, which have led to a great improvement of this paper. This research was supported by the U.S. National Science Foundation Grant DMS1811734 and NIH Grant R01ES024732.

## Appendix

### A.1. Comparison of Second-order online methods

Table A1: In the column **Method**, "SGD" includes both first-order procedures and second-order procedures that are based only on the diagonal elements of an approximated Hessian matrix, not on the full estimated Hessian. In the column **Hessian matrix**, "Full" indicates whether the full $p \times p$ (approximated) Hessian matrix is used in an algorithm; "Exact" indicates whether the Hessian matrix is approximated or obtained by the second-order derivative of the log-likelihood function (i.e. no approximation). In the column **Inference**, "Yes" means the availability of statistical inference. See more details in the Appendix below.

| **Method** | **Computational cost per iteration** | **Tuning parameter** | **Hessian matrix** | | **Inference** |
|---|---|---|---|---|---|
| | | | Full | Exact | |
| SGD | $O(p)$ | Yes | No | No | No |
| Online Newton | $O(p^2)$ | Yes | Yes | No | No |
| Online BFGS | $O(p^2)$ | Yes | Yes | No | No |
| Online LBFGS | $O(\tau p)$, $\tau < p$ | Yes | No | No | No |
| Renewable | $O(n_b p^2 + p^3)$ | No | Yes | Yes | Yes |

### A.2. Notations for Existing Methods

Table A2: Summary of notations. The variances of online LSE (OLSE), CEE and CUEE are all given in the Supplementary Material section S1.

| **Method** | Estimator | Single-Batch Hessian | Aggregated Hessian | Variance |
|---|---|---|---|---|
| Oracle MLE | $\hat{\boldsymbol{\beta}}_b^\star$ | - | - | $\hat{\boldsymbol{V}}_b^\star$ |
| AI-SGD | $\boldsymbol{\beta}_{N_b}^{\text{aim}}$ | - | - | - |
| OLSE | $\tilde{\boldsymbol{\beta}}_b^{\text{olse}}$ | $\boldsymbol{X}_b^T \boldsymbol{X}_b$ | $\sum_{j=1}^b \boldsymbol{X}_j^T \boldsymbol{X}_j$ | $\tilde{\boldsymbol{V}}_b^{\text{olse}}$ |
| CEE | $\tilde{\boldsymbol{\beta}}_b^{\text{cee}}$ | $\boldsymbol{A}_b^{\text{cee}}$ | $\tilde{\boldsymbol{A}}_b^{\text{cee}}$ | $\tilde{\boldsymbol{V}}_b^{\text{cee}}$ |
| CUEE | $\tilde{\boldsymbol{\beta}}_b^{\text{cuee}}$ | $\boldsymbol{A}_b^{\text{cuee}}$ | $\tilde{\boldsymbol{A}}_b^{\text{cuee}}$ | $\tilde{\boldsymbol{V}}_b^{\text{cuee}}$ |
| Renewable | $\tilde{\boldsymbol{\beta}}_b$ | $\boldsymbol{J}_b$ | $\tilde{\boldsymbol{J}}_b$ | $\tilde{\phi}_b \tilde{\boldsymbol{J}}_b^{-1}$ |

### A.3. Proof of Consistency

Assume the conditions (C1)-(C3) given in Section 4.1 hold. The MLE of the cumulative dataset to time point $b$ is $\hat{\boldsymbol{\beta}}_b^\star = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\max}\, \ell_{N_b}(\boldsymbol{\beta}, \phi; D_b^\star)$. Under the condition $(C2)$, *i.e.*, $\boldsymbol{\mathcal{I}}_{N_b}(\boldsymbol{\beta})$

is positive-definite, there exists a unique solution to the score equation $\sum_{j=1}^{b} \boldsymbol{U}_j(D_j;\boldsymbol{\beta}) = 0$, which is the MLE $\hat{\boldsymbol{\beta}}_b^\star$ for this cumulative dataset.

Let $\boldsymbol{\beta}_0$ be the true parameter and $\tilde{\boldsymbol{\beta}}_b$ be the renewable estimator. Note that for the prior data batch $D_1$, we have $\tilde{\boldsymbol{\beta}}_1{=}\hat{\boldsymbol{\beta}}_1^\star{=}\hat{\boldsymbol{\beta}}_1$, which is consistent by the classical theory of MLE in the GLMs. Now we prove the consistency of $\tilde{\boldsymbol{\beta}}_b$ for an arbitrary $b \geq 2$ by the method of induction.

Define a function $f_b(\boldsymbol{\beta}) = -\frac{1}{N_b}\sum_{j=1}^{b-1} \boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}_{b-1}) + \frac{1}{N_b}\boldsymbol{U}_b(D_b;\boldsymbol{\beta})$. According to equation (10), the renewable estimator $\tilde{\boldsymbol{\beta}}_b$ satisfies

$$f_b(\tilde{\boldsymbol{\beta}}_b) = \boldsymbol{0}. \tag{a.1}$$

When $\tilde{\boldsymbol{\beta}}_{b-1}$ is consistent, we have

$$f_b(\boldsymbol{\beta}_0) = \frac{1}{N_b}\sum_{j=1}^{b-1} \boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)(\tilde{\boldsymbol{\beta}}_{b-1} - \boldsymbol{\beta}_0) + \frac{1}{N_b}\boldsymbol{U}_b(D_b;\boldsymbol{\beta}_0) = o_p(1). \tag{a.2}$$

Taking a difference between equations (a.2) and (a.1), we get

$$f_b(\boldsymbol{\beta}_0) - f_b(\tilde{\boldsymbol{\beta}}_b) = \frac{1}{N_b}\sum_{j=1}^{b-1} \boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) - \frac{1}{N_b}\boldsymbol{U}_b(D_b;\tilde{\boldsymbol{\beta}}_b) + \frac{1}{N_b}\boldsymbol{U}_b(D_b;\boldsymbol{\beta}_0) = o_p(1). \tag{a.3}$$

Then, taking the first-order Taylor expansion of term $\boldsymbol{U}_b(D_b;\tilde{\boldsymbol{\beta}}_b)$ in equation (a.3) around $\boldsymbol{\beta}_0$, we obtain

$$\boldsymbol{U}_b(D_b;\tilde{\boldsymbol{\beta}}_b) = \boldsymbol{U}_b(D_b;\boldsymbol{\beta}_0) - \left\{\boldsymbol{J}_b(D_b;\boldsymbol{\beta}_0) - \boldsymbol{J}_b(D_b;\boldsymbol{\beta}_0) + \boldsymbol{J}_b(D_b;\boldsymbol{\xi}_b)\right\}(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0), \tag{a.4}$$

where $\boldsymbol{\xi}_b$ lies in between $\tilde{\boldsymbol{\beta}}_b$ and $\boldsymbol{\beta}_0$. By the Lipschitz continuity in condition (C3), there exists $M(D_b) > 0$ such that

$$\|\boldsymbol{J}_b(D_b;\boldsymbol{\xi}_b) - \boldsymbol{J}_b(D_b;\boldsymbol{\beta}_0)\| \leq M(D_b)\|\boldsymbol{\xi}_b - \boldsymbol{\beta}_0\| \leq M(D_b)\|\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0\|. \tag{a.5}$$

Using (a.5), we rewrite (a.4) as

$$\boldsymbol{U}_b(D_b;\tilde{\boldsymbol{\beta}}_b) = \boldsymbol{U}_b(D_b;\boldsymbol{\beta}_0) - \boldsymbol{J}_b(D_b;\boldsymbol{\beta}_0)(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) + \mathcal{O}_p(n_b\|\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0\|^2). \tag{a.6}$$

Combining equations (a.3) and (a.6), we yield

$$f_b(\boldsymbol{\beta}_0) - f_b(\tilde{\boldsymbol{\beta}}_b) = \frac{1}{N_b}\left\{\sum_{j=1}^{b-1} \boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{J}_b(D_b;\boldsymbol{\beta}_0)\right\}(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) + O_p\left(\frac{n_b}{N_b}\|\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0\|^2\right) = o_p(1). \tag{a.7}$$

Under the assumption that $\tilde{\boldsymbol{\beta}}_j$ is consistent and $\tilde{\boldsymbol{\beta}}_j \in \mathcal{B}_{N_j}(\delta)$ for $j = 1, \ldots, b-1$, and by condition $(C2)$, we know $N_b^{-1}\left\{\sum_{j=1}^{b-1} \boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{J}_b(D_b;\boldsymbol{\beta}_0)\right\}$ is positive-definite. It follows that $\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0 \xrightarrow{p} \boldsymbol{0}$, as $N_b \to \infty$.

### *A.4. Proof of Asymptotic Normality*

(i) For the first data batch, with $b = 1$ and $n_1 = N_1$, the MLE $\hat{\boldsymbol{\beta}}_1^\star = \hat{\boldsymbol{\beta}}_1 = \tilde{\boldsymbol{\beta}}_1$ satisfies $\frac{1}{N_1}\boldsymbol{U}_1(D_1;\tilde{\boldsymbol{\beta}}_1) = \boldsymbol{0}$ and $\sqrt{N_1}(\tilde{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_0) \xrightarrow{d} \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_0)$, as $N_1 = n_1 \to \infty$. In addition, its score function has the following stochastic expression:

$$\frac{1}{N_1}\boldsymbol{U}_1(D_1;\boldsymbol{\beta}_0) = \frac{1}{N_1}\boldsymbol{J}_1(D_1;\hat{\boldsymbol{\beta}}_1)(\hat{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_0) + O_p\left(\frac{n_1}{N_1}\|\hat{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_0\|^2\right), \tag{a.8}$$

where we leave $\frac{n_1}{N_1} = 1$ in the expression for the convenience of mathematical arguments used in the subsequent proof.

(ii) Consider updating $\tilde{\boldsymbol{\beta}}_{b-1}$ to $\tilde{\boldsymbol{\beta}}_b$. The oracle MLE $\hat{\boldsymbol{\beta}}_b^\star$ for the cumulative dataset $D_b^\star$ satisfies: $\frac{1}{N_b}\sum_{j=1}^b \boldsymbol{U}_j(D_j;\hat{\boldsymbol{\beta}}_b^\star) = \boldsymbol{0}$. Taking the first-order Taylor expansion around $\boldsymbol{\beta}_0$ leads to

$$\frac{1}{N_b}\sum_{j=1}^b \boldsymbol{U}_j(D_j;\boldsymbol{\beta}_0) - \frac{1}{N_b}\sum_{j=1}^b \boldsymbol{J}_j(D_j;\boldsymbol{\beta}_0)(\hat{\boldsymbol{\beta}}_b^\star - \boldsymbol{\beta}_0) + O_p(\|\hat{\boldsymbol{\beta}}_b^\star - \boldsymbol{\beta}_0\|^2) = \boldsymbol{0}. \tag{a.9}$$

From the definition of $f_b(\boldsymbol{\beta})$, equations (a.1) and (a.7), we know that

$$\begin{aligned} f_b(\boldsymbol{\beta}_0) &= -\frac{1}{N_b}\sum_{j=1}^{b-1} \boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)(\boldsymbol{\beta}_0 - \tilde{\boldsymbol{\beta}}_{b-1}) + \frac{1}{N_b}\boldsymbol{U}_b(D_b;\boldsymbol{\beta}_0) \\ &= \frac{1}{N_b}\left\{\sum_{j=1}^{b-1} \boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{J}_b(D_b;\boldsymbol{\beta}_0)\right\}(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) + O_p\left(\frac{n_b}{N_b}\|\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0\|^2\right) = o_p(1). \end{aligned}$$

It follows that

$$\begin{aligned} -\frac{1}{N_b}\left\{\sum_{j=1}^{b-1} \boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{J}_b(D_b;\boldsymbol{\beta}_0)\right\}(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) + \frac{1}{N_b}\sum_{j=1}^{b-1} \boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)(\tilde{\boldsymbol{\beta}}_{b-1} - \boldsymbol{\beta}_0) + \frac{1}{N_b}\boldsymbol{U}_b(D_b;\boldsymbol{\beta}_0) \\ + O_p\left(\frac{n_b}{N_b}\|\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0\|^2\right) = \boldsymbol{0}. \end{aligned} \tag{a.10}$$

Similar to equation (a.8), at the $(b-1)$-th data batch, it is easy to show that

$$\frac{1}{N_{b-1}}\sum_{j=1}^{b-1} \boldsymbol{U}_j(D_j;\boldsymbol{\beta}_0) = \frac{1}{N_{b-1}}\sum_{j=1}^{b-1} \boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)(\tilde{\boldsymbol{\beta}}_{b-1} - \boldsymbol{\beta}_0) + O_p\left(\sum_{j=1}^{b-1}\frac{n_j}{N_{b-1}}\|\tilde{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_0\|^2\right). \tag{a.11}$$

Plugging equation (a.11) into equation (a.10), we obtain

$$\frac{1}{N_b}\sum_{j=1}^{b} \boldsymbol{U}_j(D_j;\boldsymbol{\beta}_0) - \frac{1}{N_b}\left\{\sum_{j=1}^{b-1} \boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{J}_b(D_b;\boldsymbol{\beta}_0)\right\}(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) + O_p\left(\sum_{j=1}^{b}\frac{n_j}{N_b}\|\tilde{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_0\|^2\right) = \boldsymbol{0}.$$

Since according to Theorem 4.1, all $\tilde{\boldsymbol{\beta}}_j$ are consistent for $j = 1, ..., b-1$, and by condition (C3), the Continuous Mapping Theorem implies that

$$\frac{1}{N_b}\sum_{j=1}^{b} \boldsymbol{U}_j(D_j;\boldsymbol{\beta}_0) - \frac{1}{N_b}\sum_{j=1}^{b} \boldsymbol{J}_j(D_j;\boldsymbol{\beta}_0)(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) + O_p\left(\sum_{j=1}^{b}\frac{n_j}{N_b}\|\tilde{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_0\|^2\right) = \boldsymbol{0}.$$

Furthermore, since $\tilde{\phi}_b$ is a consistent estimator of $\phi_0$ due to the weak law of large numbers (WLLN), we have $\frac{1}{N_b}\tilde{\phi}_b^{-1}\sum_{j=1}^b \boldsymbol{J}_j(D_j;\boldsymbol{\beta}_0) \xrightarrow{p} \boldsymbol{\Sigma}_0^{-1}$, $N_b \to \infty$. By condition (C2), $\boldsymbol{\mathcal{I}}_{N_b}^{-1}(\boldsymbol{\beta}_0)$ exists, and thus the Central Limit Theorem implies

$$\sqrt{N_b}(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) = \left\{\sum_{j=1}^{b} \boldsymbol{J}_j(D_j;\boldsymbol{\beta}_0)\right\}^{-1}\frac{1}{\sqrt{N_b}}\sum_{j=1}^{b} \boldsymbol{U}_j(D_j;\boldsymbol{\beta}_0) + o_p(1) \xrightarrow{d} \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_0),\ N_b \to \infty. \tag{a.12}$$

### A.5. Proof of Asymptotic Equivalency

Now we prove Theorem 4.3. The difference of two equations (a.9) and (A.4) suggests that

$$\frac{1}{N_b}\sum_{j=1}^{b} \boldsymbol{J}_j(D_j;\boldsymbol{\beta}_0)(\tilde{\boldsymbol{\beta}}_b - \hat{\boldsymbol{\beta}}_b^{\star}) = O_p\left(\sum_{j=1}^{b}\frac{n_j}{N_b}\|\tilde{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_0\|^2 + \|\hat{\boldsymbol{\beta}}_b^{\star} - \boldsymbol{\beta}_0\|^2\right) = O_p(1/N_b).$$

Theorem 4.2 or equation (a.12) implies that $\|\tilde{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_0\|^2 = O_p(1/N_j)$, $j = 1,\ldots,b$. By Condition (C2), it is easy to see that

$$\|\tilde{\boldsymbol{\beta}}_b - \hat{\boldsymbol{\beta}}_b^{\star}\|_2 = O_p(1/N_b).$$

### A.6. Approximate Sufficient Statistic

To understand what types of summary statistics suit for the recursive updating procedures in the proposed renewable analytics, we establish a new notion of approximate sufficient statistic. This is an extension of the classical concept of sufficiency in the connection to the second-order incremental updating procedures, where only summary statistics of historical raw data are accessible in the subsequent updates.

Definition 1. **(Approximate Sufficient Statistic)** *Let $D = \{d_i\}_{i=1}^n \overset{i.i.d.}{\sim} f(d;\boldsymbol{\beta_0},\phi_0)$ denote a set of random samples of size $n$, and $f_n(D;\boldsymbol{\beta}_0,\phi_0)$ is the joint pdf or pmf of $D$. Suppose that the nuisance parameter $\phi_0$ is unknown and consistently estimated by $\hat{\phi}_n$, namely $\hat{\phi}_n \xrightarrow{p} \phi_0$. Let $\mathcal{B}_n(\delta)$ be a neighborhood of $\boldsymbol{\beta}_0$ defined similarly to that given in Section 4.1. A statistic $S_n(D)$ is said to be an approximate sufficient statistic for $\boldsymbol{\beta}$, if there exist functions $g\left(S_n(D);\boldsymbol{\beta}\right)$ and $c_n(D;\hat{\phi}_n)$ such that for all samples in $D$ and all parameters $\boldsymbol{\beta} \in \mathcal{B}_n(\delta)$, $f_n(D;\boldsymbol{\beta},\hat{\phi}_n) = g_n(D;\boldsymbol{\beta},\hat{\phi}_n)c_n(D;\hat{\phi}_n)$, with $g_n(D;\boldsymbol{\beta},\hat{\phi}_n) = g\left(S_n(D);\boldsymbol{\beta}\right) + o_p(1)$. In particular, when the nuisance parameter $\phi_0$ is known, the factorization expression reduces to $f_n(D;\boldsymbol{\beta}) = g_n\left(D;\boldsymbol{\beta}\right)c_n(D)$, with $g_n(D;\boldsymbol{\beta}) = g\left(S_n(D);\boldsymbol{\beta}\right) + o_p(1)$.*

The above definition is well suited with the logistic model and Poisson model with $\phi_0 = 1$, as well as the linear model or gamma model with an unknown $\phi_0$. In the latter case, we replace the nuisance parameter $\phi_0$ with an unbiased/consistent estimator in the derivation of $S_n(D)$. Thus, $\boldsymbol{\beta}$ depends on data $D$ only through $S_n(D)$, approximately. In Supplementary Material section S2, we prove the summary statistics used in the proposed renewable analytics are approximate sufficient statistics in the framework of GLMs. Also, we present an interesting example of approximate sufficient statistic in the linear model where the factorization holds exactly.

## References

Amari, S.-I., Park, H. and Fukumizu, K. (2000) Adaptive method of realizing natural gradient learning for multilayer perceptrons. *Neural Computation*, **12**, 1399–1409.

Bifet, A., Maniu, S., Qian, J., Tian, G., He, C. and Fan, W. (2015) Streamdm: Advanced data mining in spark streaming. In *IEEE International Conference on Data Mining Workshop, ICDMW 2015, Atlantic City, NJ, USA, November 14-17, 2015*, 1608–1611.

Bordes, A., Bottou, L. and Gallinari, P. (2009) Sgd-qn: Careful quasi-newton stochastic gradient descent. *Journal of Machine Learning Research*, **10**, 1737–1754.

Bucak, S. S. and Gunsel, B. (2009) Incremental subspace learning via non-negative matrix factorization. *Pattern Recognition*, **42**, 788–797.

Cardot, H. and Degras, D. (2015) Online principal component analysis in high dimension: Which algorithm to choose? arXiv:1511.03688.

Chen, X. and Xie, M. (2014) A split-and-conquer approach for analysis of extraordinarily large data. *Statistica Sinica*, **24**, 1655–1684.

Chen, Y., Berrocal, V. J., Bingham, R. and Song, P. X. (2014) Analysis of spatial variations in the effectiveness of graduated driver's licensing (gdl) program in the state of michigan. *Spatial and Spatio-temporal Epidemiology*, **8**, 11–22.

Cox, D. R. and Reid, N. (1987) Paramater orthogonality and approximate conditional inference. *Journal of the Royal Statistical Society. Series B (Methodological)*, **49**, 1–39.

Duchi, J., Hazan, E. and Singer, Y. (2011) Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, **12**, 2121–2159.

Enea, M., Meiri, R. and Kalimi, T. (2015) *speedglm: Fitting linear and generalized linear models to large data sets.* URL: `https://cran.r-project.org/web/packages/speedglm/index.html`.

Fahrmeir, L. and Kaufmann, H. (1985) Consistency and asymptotic normality of the maximum likelihood estimator in generalized linear models. *The Annals of Statistics*, **13**, 342–368.

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

Gaber, M. M., Zaslavsky, A. and Krishnaswamy, S. (2005) Mining data streams: a review. *ACM SIGMOD Record*, **34**, 18–26.

Guha, S., Hafen, R., Rounds, J., Xia, J., Li, J., Xi, B. and Cleveland, W. S. (2012) Large complex data: divide and recombine (d&r) with rhipe. *Stat*, **1**, 53–67.

Hao, S., Zhao, P., Lu, J., Hoi, S. C. H., Miao, C. and Zhang, C. (2016) Soal: Second-order online active learning. In *International Conference on Data Mining*. Barcelona, Spain.

Hazan, E., Agarwal, A. and Kale, S. (2007) Logarithmic regret algorithms for online convex optimization. *Journal of Machine Learning Research*, **69**, 169–192.

Jørgensen, B. (1997) *The theory of dispersion models.* Chapman and Hall, London.

Kleiner, A., Talwalkar, A., Sarkar, P. and Jordan, M. I. (2014) A scalable bootstrap for massive data. *Journal of the Royal Statistical Society, Series B*, **76**, 795–816.

Liang, F., Cheng, Y., Song, Q., Park, J. and Yang, P. (2013) A resampling-based stochastic approximation method for analysis of large geostatistical data. *Journal of the American Statistical Association*, **108**, 325–339.

Lin, N. and Xi, R. (2011) Aggregated estimating equation estimation. *Statistics and Its Interface*, **4**, 73–83.

Liu, D. C. and Nocedal, J. (1989) On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, **45**, 503–528.

Lumley, T. (2013) *biglm: Bounded memory linear and generalized linear models.* URL: `https://cran.r-project.org/web/packages/biglm/index.html`.

Ma, P., Mahoney, M. W. and Yu, B. (2015) A statistical perspective on algorithm leveraging. *The Journal of Machine Learning Research*, **6**, 861–911.

Marz, N. and Warren, J. (2015) *Big Data: Principles and best practices of scalable realtime data systems.* Manning Publications.

McCullagh, P. and Nelder, J. (1983) *Generalized Linear Models.* Chapman and Hall, London.

Nion, D. and Sidiropoulos, N. D. (2009) Adaptive algorithms to track the parafac decomposition of third-order tensor. *IEEE Transactions on Signal Processing*, **57**, 2299–2310.

Nocedal, J. and Wright, S. J. (1999) *Numerical Optimization.* Springer-Verlag, New York.

Qamar, S., Guhaniyogi, R. and Dunson, D. B. (2014) Bayesian conditional density filtering. arXiv:1401.3632.

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

Sakrison, D. J. (1965) Efficient recursive estimation: application to estimating the parameter of a covariance function. *International journal of engineering science*, **3**, 461–483.

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.

Schraudolph, N. N., Yu, J. and Günter, S. (2007) A Stochastic Quasi-Newton Method for Online Convex Optimization. In *Proc. 11$^{th}$ Intl. Conf. Artificial Intelligence and Statistics (AIstats)* (eds. M. Meila and X. Shen), vol. 2 of *Workshop and Conference Proceedings*, 436–443. San Juan, Puerto Rico.

Song, P.-K., Fan, Y. and Kalbfleisch, J. (2005) Maximization by parts in likelihood inference. *Journal of the American Statistical Association (with Discussion)*, **100**, 1145–1158.

Song, P. X.-K. (2007) *Correlated data analysis.* Springer Series in Statistics.

Stengel, R. F. (1994) *Optimal control and estimation.* NY: Dover Publications Inc.

Sur, P. and Candés, E. J. (2018) A modern maximum-likelihood theory for high-dimensional logistic regression. arXiv:1803.06964v4.

Tang, L., Zhou, L. and Song, P. X.-K. (2016) Method of divide-and-combine in regularised generalised linear models for big data. arXiv:1611.06208.

Toulis, P. and Airold, E. M. (2017) Asymptotic and finite-sample properties of estimators based on stochastic gradients. *The Annals of Statistics*, **45**, 1694–1727.

Toulis, P. and Airoldi, E. M. (2015) Scalable estimation strategies based on stochastic approximations: classical results and new insights. *Statistics and computing*, **25**, 781–795.

Toulis, P., Rennie, J. and Airoldi, E. M. (2014) Statistical analysis of stochastic gradient methods for generalized linear models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, vol. 32.

Vaits, N., Moroshko, E. and Crammer, K. (2013) Second-order non-stationary online learning for regression. arXiv:1303.0140.

Xu, W. (2011) Towards optimal one pass large scale learning with averaged stochastic gradient descent. arXiv:1107.2490.

Zhou, L. and Song, P. X.-K. (2017) Scalable and efficient statistical inference with estimating functions in the mapreduce paradigm for big data. arXiv:1709.04389.