ROYAL STATISTICAL SOCIETY
DATA | EVIDENCE | DECISIONS

Journal of the Royal Statistical Society
Statistical Methodology
Series B

J. R. Statist. Soc. B (2020)
82, *Part* 1, *pp.* 69–97

Renewable estimation and incremental inference in generalized linear models with streaming data sets

Lan Luo and Peter X.-K. Song

University of Michigan, Ann Arbor, USA

[Received January 2018. Final revision October 2019]

Summary. The paper presents an incremental updating algorithm to analyse streaming data sets using generalized linear models. The method proposed 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 us 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; On-line learning; Spark computing platform; Stochastic gradient descent algorithm

1. Introduction

We consider a classical problem where a series of cross-sectional data sets becomes available sequentially. Such a type of data collection is pervasive in practice and is referred to as streaming data sets throughout this paper. Statistical analysis of streaming data sets has recently drawn considerable attention in the emerging field of ‘big data’ analytics due to the availability of modern powerful computing platforms such as Apache Spark (Bifet *et al.*, 2015). The key methodology that is relevant to such data analysis pertains to algorithms that enable us to update certain statistics of interest sequentially. For example, the sample mean may be recursively updated along data streams in which only previous sample means, instead of the entire historical subject level data, are needed. Specifically, consider two data sets arriving sequentially, where $D_1=(x_{11},\ldots,x_{1n_1})$ denotes the first data set of n_1 observations. Suppose that we want to update the sample mean when the second batch of data $D_2=(x_{21},\ldots,x_{2n_2})$ 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, i.e.

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

The defining feature in this operation is that the mean from the previous data, $\delta(D_1)$, rather than the data D_1 themselves, is used in the calculation. In this paper, a statistic that satisfies such a property is termed a *renewable estimator*. Indeed, the recursive operation that is exemplified

Address for correspondence: Peter X.-K. Song, Department of Biostatistics, University of Michigan, 1420 Washington Heights, Ann Arbor, MI 48109-2029, USA.
E-mail: pxsong@umich.edu

1369–7412/20/82069

in equation (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 equation (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 a sufficient statistic and is of critical importance in handling big data as far as computing memory and speed are concerned. This strategy has been widely advocated in the literature of on-line learning, incremental analytics, matrix or tensor decomposition and classification, and on-line Bayesian inference; see Bucak and Gunsel (2009), Cardot and Degras (2018), Nion and Sidiropoulos (2009) and Qamar *et al.* (2018), among others.

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

The interest in developing procedures allowing 'quick' updates of parameter estimates along with sequentially arriving 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 to a data sequence in the form of an open-ended set of independent observations, $y_i \sim^{\mathrm{IID}} f(y;\boldsymbol{\theta}_0)$, 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, i.e.

$$\boldsymbol{\theta}_i^{\mathrm{sgd}} = \boldsymbol{\theta}_{i-1}^{\mathrm{sgd}} + \gamma_i \mathbf{C}_i \nabla_{\boldsymbol{\theta}} \log\{f(y_i;\boldsymbol{\theta}_{i-1}^{\mathrm{sgd}})\},$$

where $\gamma_i > 0$ is a prespecified learning rate sequence such that $i\gamma_i \rightarrow \gamma$ as $i \rightarrow \infty$ and $\{\mathbf{C}_i\}$ 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 was later termed 'explicit SGD' by Toulis *et al.* (2014). Under the condition that $\gamma_i \mathbf{C}_i \rightarrow \mathcal{I}^{-1}(\boldsymbol{\theta}_0)$, $i \rightarrow \infty$, where $\mathcal{I}(\boldsymbol{\theta}_0)$ is the Fisher information matrix, this updating method enjoys some theoretical guarantees. For example, as $i \rightarrow \infty$, $\boldsymbol{\theta}_i^{\mathrm{sgd}} \rightarrow^{\mathrm{P}} \boldsymbol{\theta}_0$ with optimal asymptotic efficiency, namely, its asymptotic covariance matrix is $\mathcal{I}^{-1}(\boldsymbol{\theta}_0)$.

However, the SGD method is generally not robust to learning rate misspecification, and the algorithm may fail to converge if γ is too large. An improvement, which was called 'implicit SGD' by Toulis *et al.* (2014), is given by $\boldsymbol{\theta}_i^{\mathrm{im}}$ that appears on both sides of the updating equation, i.e.

$$\boldsymbol{\theta}_i^{\mathrm{im}} = \boldsymbol{\theta}_{i-1}^{\mathrm{im}} + \gamma_i \mathbf{C}_i \nabla_{\boldsymbol{\theta}} \log\{f(y_i;\boldsymbol{\theta}_i^{\mathrm{im}})\}.$$

According to a comparison of these two versions of SGD algorithms in GLMs, Toulis *et al.*

Table 1. Comparison of second-order on-line methods†

Method	*Computational cost per iteration*	*Tuning parameter*	*Hessian matrix*		*Inference*
			Full	*Exact*	
SGD	$O(p)$	Yes	No	No	No
On-line Newton	$O(p^2)$	Yes	Yes	No	No
oBFGS	$O(p^2)$	Yes	Yes	No	No
oLBFGS	$O(\tau p), \tau < p$	Yes	No	No	No
Renew	$O(n_b p^2 + p^3)$	No	Yes	Yes	Yes

†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 and '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 Appendix A.

(2014) concluded that implicit SGD appeared more robust to learning rate misspecification. To improve statistical efficiency, Toulis *et al.* (2014) further proposed averaged implicit SGD (AISGD); see the detail in Section 2.1. To avoid calculating the inverse of a 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. the Hessian's diagonal elements) is recorded and updated over iterations.

There are some on-line second-order methods such as the natural gradient algorithm (Amari *et al.*, 2000) and the on-line Newton step (Hazan *et al.*, 2007) that maintain complete information matrices over iterations. Similarly 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.* (2015) and 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, because of the failure of the Bartlett identity (Song (2007), chapter 2), the Fisher information alone cannot provide valid statistical inference. For on-line quasi-Newton methods, both the Broyden–Fletcher–Goldfarb–Shanno (Nocedal and Wright, 1999) and the limited memory Broyden–Fletcher–Goldfarb–Shanno (Liu and Nocedal, 1989) algorithms have been modified for streaming data, respectively termed oBFGS and oLBFGS (Schraudolph *et al.*, 2007; Bordes *et al.*, 2009). But, in these procedures, it is unclear whether the estimated approximate Hessian is appropriate for statistical inference. A detailed comparison between these second-order on-line methods is available in Table 1.

Although some relevant analytic expressions for the asymptotic variances have been derived in both explicit and implicit SGD (Toulis and Airold, 2017), the work of developing on-line confidence intervals remains unexplored because of the lack of suitable asymptotic results that may be directly applied to establish on-line inference. Recently Fang (2019) proposed a perturbation-based resampling method to construct confidence intervals for AISGD. Even though this on-line 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 algorithm, several cumulative updating methods have been proposed specifically to perform sequential updating of regression coefficient estimators, including the on-line 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 of Schifano *et al.* (2016) for non-linear models. Even though the CUEE estimator is shown to have less estimation bias than the CEE with finite sample sizes, its estimation consistency has been established on a strong regularity condition: the total number of streaming data sets, *say* B, needs to satisfy the order of $B=\mathcal{O}(n_j^k)$, with $k<\frac{1}{3}$ for all j, where n_j is the size of the jth data batch (Lin and Xi, 2011; Schifano *et al.*, 2016). This condition is also required by the 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 data sets 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 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. Fig. 1 shows a schematic outline of how the speed and batch layers interact when a new data batch arrives. Transient and rough realtime views are captured at the speed layer by using incremental algorithms, where previously stored views are updated with an incoming batch of data to generate renewed views. Indeed, SGD is one of the most popular incremental algorithms used to process high throughput streaming

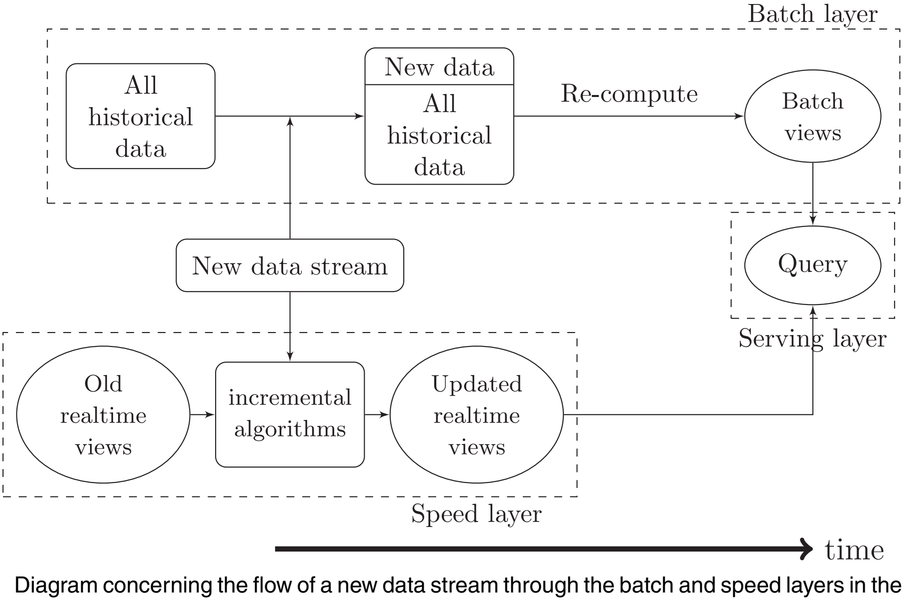

Fig. 1. Diagram concerning the flow of a new data stream through the batch and speed layers in the lambda architecture: the serving layer is responsible for indexing and exposing the views from the batch and speed layers so that they can be queried

data via the Spark system (Bifet *et al.*, 2015). The batch layer stores constantly growing data and continuously recomputes the batch views when a new batch of data arrives. Despite latency, the batch layer refines results that are produced in the speed layer where the accuracy of estimation 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 recomputations. Unfortunately, this powerful architecture has completely ignored the need for realtime statistical inference; for example, there are no gears in the system designed to compute and store Fisher information sequentially or as such, which is a critical piece required for statistical inference. To overcome this, in this paper we propose to expand the speed layer by adding a new 'inference layer', and we name this new subarchitecture 'rho architecture' (from the initial letter of the Greek word for 'stream': $\rho\varepsilon\nu\mu\alpha$). Fig. 2 in Section 3.2 displays the resulting expanded architecture enabling statistical inference to be conducted with streaming data.

In the proposed rho architecture, we aim to address three basic questions:

(a) what types of summary statistics are to be stored in the inference layer;
(b) how to update those summary statistics required for estimation and inference without the use of previous raw data;
(c) how to optimize the efficiency of estimation of renewable estimation so that it may be asymptotically equivalent to the MLE obtained from the entire data set.

Our goal is to fit a GLM (McCullagh and Nelder, 1983) $\mathbb{E}(y_i|\mathbf{x}_i)=g(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})$, $i=1,\ldots,N_b$, where $g(\cdot)$ is a known link function and N_b is the sample size of aggregated streaming data up to batch b, $N_b=\Sigma_{j=1}^{b}n_j$. At batch $b\geqslant 2$, a total of N_b observations becomes available in a series of b batches of data, denoted by $D_1=\{\mathbf{y}_1,\mathbf{X}_1\},\ldots,D_b=\{\mathbf{y}_b,\mathbf{X}_b\},\ldots,$ where $\mathbf{y}$ and $\mathbf{X}$ are the generic notations of the response variables and associated covariates. Under a fixed design, suppose that each observation is drawn from $(y_i;\mathbf{x}_i)\sim f(y;\mathbf{x},\boldsymbol{\beta}_0,\phi_0)$, $i=1,\ldots,N_b$, independently, where $\boldsymbol{\beta}_0\in\mathbb{R}^p$ is the true value of the parameter of interest and ϕ_0 is the true value of a nuisance parameter. Let $D_b^*=\{D_1,\ldots,D_b\}$ 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^* (an aggregation of b batches) as the sets of indices for subjects involved. For a GLM, we may write out the associated log-likelihood function in the form of an exponential dispersion model (Jørgensen, 1997):

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

where $d(y_i;\mu_i)$ is the unit deviance function involving the mean parameter $\mu_i=\mathbb{E}(y_i|\mathbf{x}_i)$, 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 $\mu_i=g(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})$, $i\in D_b^*$. It is known that, in the Gaussian linear model, the dispersion parameter ϕ is the variance parameter and, in both Bernoulli logistic and Poisson log-linear regression models, $\phi=1$. Denote the (unit) score function by $\mathbf{U}(y_i;\mathbf{x}_i,\boldsymbol{\beta}):=\nabla_{\boldsymbol{\beta}}d(y_i;\mu_i)$. Then, the MLE $\hat{\boldsymbol{\beta}}_b^*$ satisfying $\Sigma_{i\in D_b^*}\mathbf{U}(y_i;\mathbf{x}_i,\boldsymbol{\beta})=\mathbf{0}$ is the oracle estimator, which in general has no closed form solution. It is often obtained numerically by certain iterative algorithms such as the Newton–Raphson algorithm. Note that in the GLM the MLE $\hat{\boldsymbol{\beta}}_b^*$ is derived with no involvement of nuisance parameter ϕ because of so-called parameter orthogonality (Cox and Reid, 1987). For details 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 equation (1) are generally unavailable for GLMs.

The focus of this paper is to develop a new on-line framework in which both likelihood estimation and inference can be updated with current data and summary statistics of historical data. Our new contributions include the following:

(a) we propose a rho architecture as an expansion of the Spark lambda architecture for on-line statistical inference;
(b) the proposed renewable estimator is shown to be asymptotically equivalent to the oracle MLE without the strong condition $B = \mathcal{O}(n_j^k)$, $k < \frac{1}{3}$;
(c) the l_2-norm difference between our renewable estimator and the oracle MLE vanishes as the total sample size increases;
(d) being computationally advantageous, our method does not require reaccess to any old subject level data after the completion of the 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 batches of data increases fast and/or perpetually. Also, our method provides realtime interim inference based on the Wald test.

The paper is organized as follows. Section 2 gives a brief overview of existing methods to which the method proposed 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 with the oracle MLE and existing on-line 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 Appendix A and the on-line supplementary materials.

2. Existing methods

Two primary classes of on-line data analytics have been developed in the literature, including SGD algorithms and sequential estimation procedures. At an intermediary batch b, $\hat{\boldsymbol{\beta}}_b^*$ denotes the oracle MLE obtained with the entire cumulative data set D_b^*, and $\tilde{\boldsymbol{\beta}}_b$ denotes a renewable estimator with the same data set D_b^*. Throughout this paper, a circumflex over a symbol (e.g. $\hat{\boldsymbol{\beta}}$) denotes an MLE, and an asterisk in the superscript (e.g. $\hat{\boldsymbol{\beta}}_b^*$) indicates a statistic that is derived from a cumulative data set D_b^*; otherwise, it is based on a single batch of data (e.g. $\hat{\boldsymbol{\beta}}_b$ from D_b). Likewise, a tilde over a symbol (e.g. $\tilde{\beta}$) denotes a quantity that is obtained sequentially by an incremental algorithm. For example, $\tilde{\boldsymbol{\beta}}$ denotes an estimator obtained by an on-line updating procedure (e.g. OLSE). For convenience, we list all the necessary notation in Table 2.

2.1. Stochastic gradient descent algorithm: averaged implicit stochastic gradient descent

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

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

When fixing $W_i^{(s)} \equiv 1$, expression (3) gives the AISGD estimate. Using samples drawn from,

Table 2. Summary of notation†

Method	*Estimator*	*Single-batch Hessian*	*Aggregated Hessian*	*Variance*
Oracle MLE	$\hat{\boldsymbol{\beta}}_b^*$	—	—	$\hat{\mathbf{V}}_b^*$
AISGD	$\boldsymbol{\beta}_{N_b}^{\text{aim}}$	—	—	—
OLSE	$\tilde{\boldsymbol{\beta}}_b^{\text{olse}}$	$\mathbf{X}_b^{\text{T}}\mathbf{X}_b$	$\Sigma_{j=1}^b \mathbf{X}_j^{\text{T}}\mathbf{X}_j$	$\tilde{\mathbf{V}}_b^{\text{olse}}$
CEE	$\tilde{\boldsymbol{\beta}}_b^{\text{cee}}$	$\mathbf{A}_b^{\text{cee}}$	$\tilde{\mathbf{A}}_b^{\text{cee}}$	$\tilde{\mathbf{V}}_b^{\text{cee}}$
CUEE	$\tilde{\boldsymbol{\beta}}_b^{\text{cuee}}$	$\mathbf{A}_b^{\text{cuee}}$	$\tilde{\mathbf{A}}_b^{\text{cuee}}$	$\tilde{\mathbf{V}}_b^{\text{cuee}}$
Renew	$\tilde{\boldsymbol{\beta}}_b$	$\mathbf{J}_b$	$\tilde{\mathbf{J}}_b$	$\tilde{\phi}_b\tilde{\mathbf{J}}_b^{-1}$

†The variances of OLSE, CEE and CUEE are all given in the on-line supplementary material section S1.

say, $W_i^{(s)} \sim^{\text{IID}}$ exponential(1), $s=1,\ldots,S$, we obtain S copies of $\boldsymbol{\beta}_i^{(\text{s})\text{aim}}$. Further, using these replicates, we can assess the variability of $\boldsymbol{\beta}_i^{(\text{s})\text{aim}}$ and calculate the empirical standard error of the AISGD estimator for statistical inference. In Section 6, through simulation studies we compare our renewable estimation method with this AISGD 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 on-line supplementary material section S1.

2.2.1. On-line least squares estimation

Consider a linear model $y_i = \mathbf{x}_i^{\text{T}}\boldsymbol{\beta}_0 + \epsilon_i$, with independent and identically distributed (IID) errors ϵ_is, $i=1,\ldots,N_b$. The least squares estimator (LSE) for the current single data batch D_b is $\hat{\boldsymbol{\beta}}_b = (\mathbf{X}_b^{\text{T}}\mathbf{X}_b)^{-1}\mathbf{X}_b^{\text{T}}\mathbf{y}_b$. 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:

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

2.2.2. On-line estimating equations

Let $\boldsymbol{\beta}_0$ be a parameter value satisfying $\Sigma_{i\in D_b^*}\mathbb{E}\{\psi(y_i,\mathbf{x}_i;\boldsymbol{\beta}_0)\} = \mathbf{0}$, where $\psi(\cdot)$ 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), the CEE estimator $\tilde{\boldsymbol{\beta}}_b^{\text{cee}}$ takes the following meta-estimation form:

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

with initial $\tilde{\mathbf{A}}_0^{\text{cee}} = \mathbf{0}_{p\times p}$, and $\mathbf{A}_b^{\text{cee}} = -\Sigma_{i\in D_b}\nabla_{\boldsymbol{\beta}}\psi(y_i,\mathbf{x}_i;\hat{\boldsymbol{\beta}}_b)$ 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 expression (5) is of order $\mathcal{O}(\Sigma_{j=1}^b n_j^{-1/2})$, 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 $\tilde{\beta}_b^{\mathrm{cee}}$ in expression (5) suffers an increased bias as $b \to \infty$. To reduce bias, the CUEE estimator was proposed by Schifano *et al.* (2016). See the related detail in the on-line supplementary material section S1. It is worth pointing out that estimation consistency of the CEE or CUEE is established under a strong regularity condition, $b = \mathcal{O}(n_j^k)$, for $k < \frac{1}{3}$ and all j. This condition hardly holds for high throughput data streams, where n_j is typically small whereas 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 $\tilde{\beta}_b$ be a renewable estimator, initialized by the MLE $\tilde{\beta}_1$ or $\hat{\beta}_1$, from the first batch of data D_1. For $b = 2, 3, \ldots$, a previous estimator $\tilde{\beta}_{b-1}$ is sequentially updated to $\tilde{\beta}_b$ when data batch D_b arrives; after the updating, data batch D_b is no longer accessible except estimate $\tilde{\beta}_b$ and summary statistics $\mathbf{J}_b(D_b; \tilde{\beta}_b)$ and $\tilde{\phi}_b$, which are carried forward in future calculations. Let $\mathbf{U}_b(D_b; \beta) = \Sigma_{i \in D_b} \mathbf{U}(y_i; \mathbf{x}_i, \beta)$ be the score function of data batch D_b. Denote the single-batch negative Hessian by $\mathbf{J}_b(D_b; \beta) := -\nabla_\beta \mathbf{U}_b(D_b; \beta)$.

3.1. Method

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

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

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

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

Since D_1 and D_2 are independently sampled from the same underlying model with a common true parameter β_0, when $\min\{n_1, n_2\}$ is sufficiently large, under some mild regularity conditions, both $\hat{\beta}_1$ and $\hat{\beta}_2^*$ are consistent estimators of β_0 (e.g. Fahrmeir and Kaufmann (1985)). This implies that the error term $\mathcal{O}_p(\|\hat{\beta}_2^* - \hat{\beta}_1\|^2)$ in equation (7) may be asymptotically ignored. Removing such a term, we propose a new estimator $\tilde{\beta}_2$ as a solution to the equation of the form

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

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

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

$\tilde{\beta}_2$ in equation (8) approximates the oracle MLE $\hat{\beta}_2^*$ in equation (6) up to second-order asymptotic errors. Through equation (8), the initial $\hat{\beta}_1$ is renewed by $\tilde{\beta}_2$. Because of this, in this paper $\tilde{\beta}_2$ is called *a renewable estimator* of β_0, and equation (8) is termed *an incremental estimating equation*. Numerically, it is quite straightforward to find $\tilde{\beta}_2$ by, for example, the Newton–Raphson algorithm or Fisher scoring algorithm with $\phi = 1$. These two algorithms are equivalent in the GLM with a canonical link, i.e., at the $(r+1)$th iteration,

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

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

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

where $\tilde{\mathbf{U}}_2^{(r)} = \mathbf{J}_1(D_1;\hat{\beta}_1)(\hat{\beta}_1 - \tilde{\beta}_2^{(r)}) + \mathbf{U}_2(D_2;\tilde{\beta}_2^{(r)})$. In equation (9), $\tilde{\beta}_2$ is iteratively solved by using the adjusted score function $\tilde{\mathbf{U}}_2^{(r)}$ and the aggregated negative Hessian $\{\mathbf{J}_1(\hat{\beta}_1) + \mathbf{J}_2(\hat{\beta}_1)\}$ evaluated at the previous estimate $\hat{\beta}_1$. We name algorithm (9) the *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 a Hessian matrix; see, for example, Song *et al.* (2005). The difference between the proposed $\tilde{\beta}_2$ and the oracle MLE $\hat{\beta}_2^*$ stems from an approximation to the score function $\mathbf{U}_1(D_1;\hat{\beta}_2^*)$. As shown in theorem 3 in Section 4.1, such a distance vanishes at the rate of $1/N_2$, with $N_2 = |D_2^*| = n_1 + n_2$. In practice, because the cumulative sample size $N_b = \Sigma_{j=1}^{b} n_j$ increases to ∞ very fast, these two estimators, $\tilde{\beta}_b$ and $\hat{\beta}_b^*$, are numerically very close, and eventually become the same. To run algorithm (9), we extend the Spark lambda architecture to store three key components: $\{\hat{\beta}_1, \mathbf{J}_1(D_1;\hat{\beta}_1), \hat{\phi}_1\}$. 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 $\hat{\mu}_i = g(\mathbf{x}_i^{\mathrm{T}}\hat{\beta}_1)$ and $v(\cdot)$ is the unit variance function.

Generalizing the above procedure to streaming data sets, we now propose a renewable estimation of β_0 as follows. A renewable estimator $\tilde{\beta}_b$ of β_0 is defined as a solution to the following incremental estimating equation:

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

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

$$\tilde{\beta}_b^{(r+1)} = \tilde{\beta}_b^{(r)} + \{\tilde{\mathbf{J}}_{b-1} + \mathbf{J}_b(D_b;\tilde{\beta}_{b-1})\}^{-1}\tilde{\mathbf{U}}_b^{(r)}, \qquad (11)$$

where the adjusted score $\tilde{\mathbf{U}}_b^{(r)} = \tilde{\mathbf{J}}_{b-1}(\tilde{\beta}_{b-1} - \tilde{\beta}_b^{(r)}) + \mathbf{U}_b(D_b;\tilde{\beta}_b^{(r)})$ is updated over iterations. Again, algorithm (11) uses only subject level data of current batch D_b and summary statistics $\{\tilde{\beta}_{b-1}, \tilde{\mathbf{J}}_{b-1}, \tilde{\phi}_{b-1}\}$ from historical data. Also, a consistent estimator of parameter ϕ 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

$$\hat{\phi}_b = \frac{1}{n_b - p} \sum_{i \in D_b} \frac{(y_i - \hat{\mu}_i)^2}{v(\hat{\mu}_i)}.$$

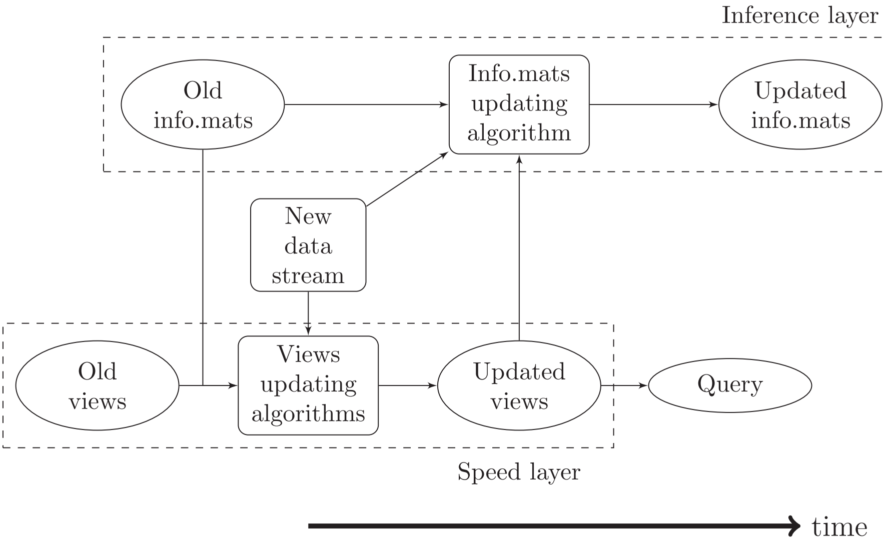

Fig. 2. Rho architecture: an expanded speed layer of the lambda architecture with an addition of an inference layer for incremental updating of quantities required by statistical inference

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 co-ordination between the 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 Fig. 2, the resulting rho architecture consists of a speed layer and an inference layer that is responsible for inferential statistics updating. When a new batch of data 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 equation (11) is implemented in the rho architecture.

3.3. An example: linear model

To see the specific operational details that were discussed above, here we present renewable estimation in the Gaussian linear model. For the linear model, the renewable estimation proposed turns out to be identical to OLSE given in equation (4), with more details available in the on-line supplementary material section S1.

3.3.1. Example 1

Consider data batch $D_b=\{\mathbf{y}_b,\mathbf{X}_b\}$ with outcome $\mathbf{y}_b=(y_{b1},\ldots,y_{bn_b})^{\mathrm{T}}$ and covariates $\mathbf{X}_b=(\mathbf{x}_{b1},\ldots,\mathbf{x}_{bn_b})^{\mathrm{T}}$, and $\mathbf{y}_b|\mathbf{X}_b$ are independently sampled from a Gaussian distribution with mean $\boldsymbol{\mu}_b=(\mu_{b1},\ldots,\mu_{bn_b})^{\mathrm{T}}$ and variance $\phi\mathbf{I}$ such that $\mu_{bi}=\mathbb{E}(y_{bi}|\mathbf{x}_{bi})=\mathbf{x}_{bi}^{\mathrm{T}}\boldsymbol{\beta}_0$ and variance $V(y_{bi}|\mathbf{x}_{bi})=\phi_0$. 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 $\mathbf{U}_b(\boldsymbol{\beta})=\mathbf{X}_b^{\mathrm{T}}(\mathbf{y}_b-\mathbf{X}_b\boldsymbol{\beta})$ and $\mathbf{J}_b(\boldsymbol{\beta})=\mathbf{X}_b^{\mathrm{T}}\mathbf{X}_b$. A closed form expression for the renewable estimator of $\boldsymbol\beta_0$ is obtained directly by solving the incremental estimating equation (10):

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

This $\tilde{\boldsymbol\beta}_b$ is calculated at the speed layer. By convention, the initials are $\tilde{\boldsymbol\beta}_0=\mathbf{0}_p$ and $\tilde{\mathbf{J}}_0=\mathbf{0}_{p\times p}$. Moreover, an unbiased estimator of ϕ_0 based on $\tilde{\boldsymbol\beta}_b$ takes the following recursive form:

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

The above $\tilde\phi_b$ is calculated and stored in the inference layer as part of the Fisher information calculation, given by $\widetilde{\mathrm{var}}(\tilde{\boldsymbol\beta}_b)=\tilde\phi_b(\tilde{\mathbf{J}}_{b-1}+\mathbf{J}_b)^{-1}$. This estimated variance of $\tilde{\boldsymbol\beta}_b$ gives exactly the same standard error as that of the oracle MLE $\hat{\boldsymbol\beta}_b^*$, which is obtained by fitting the linear model once with the entire data D_b^*. So, the proposed renewable estimator 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 equivalence 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 that $(y_i,\mathbf{x}_i)$ are IID samples from an exponential dispersion model with density $f(y;\mathbf{x},\boldsymbol\beta,\phi)$, $i=1,\ldots,N_b$, with mean $\mu_i=\mathbb{E}(y_i|\mathbf{x}_i)=g(\mathbf{x}_i^{\mathrm{T}}\boldsymbol\beta)$, $\boldsymbol\beta\in\Theta\subset\mathbb{R}^p$, and variance $V(y_i|\mathbf{x}_i)=\phi v(\mu_i)$, $\phi>0$, is the dispersion parameter, where $v(\cdot)$ is the known unit variance function. Let $\boldsymbol\beta_0$ and ϕ_0 be the true parameters. Under the canonical link, denote

$$\mathcal{I}_{N_b}(\boldsymbol\beta_0)=\sum_{i=1}^{N_b}\mathbb{E}(\mathbf{U}_i\mathbf{U}_i^{\mathrm{T}})/\phi=\sum_{i=1}^{N_b}\mathbf{x}_i v(\mu_i)\mathbf{x}_i^{\mathrm{T}}.$$

Let $\mathcal{B}_{N_b}(\delta)$ be a neighbourhood of $\boldsymbol\beta_0$, $\mathcal{B}_{N_b}(\delta)=\{\boldsymbol\beta:\|\mathcal{I}_{N_b}^{T/2}(\boldsymbol\beta-\boldsymbol\beta_0)\|\leqslant\delta\}\in\Theta$, $\delta>0$, where '$\|\cdot\|$' is the l_2-norm. Here $\mathcal{I}_{N_b}^{T/2}$ denotes the right Cholesky square root of $\mathcal{I}_{N_b}(\boldsymbol\beta_0)$, according to $\mathcal{I}_{N_b}=\mathcal{I}_{N_b}^{1/2}\mathcal{I}_{N_b}^{T/2}$. We postulate the following regularity conditions.

Condition 1 (divergence). The smallest eigenvalue of $\mathcal{I}_{N_b}(\boldsymbol\beta_0)$ satisfies $\lambda_{\min}(\mathcal{I}_{N_b})\rightarrow\infty$, as $N_b\rightarrow\infty$.

Condition 2. $\mathcal{I}_{N_b}(\boldsymbol\beta)$ is positive definite for all $\boldsymbol\beta\in\mathcal{B}_{N_b}(\delta)$.

Condition 3. The log-likelihood function $l(\boldsymbol\beta,\phi,\mathbf{x};y)$ is twice continuously differentiable and $\mathcal{I}_{N_b}(\boldsymbol\beta)$ is Lipschitz continuous in Θ.

Remark 1. Under condition 1, the neighbourhood $\mathcal{B}_{N_b}(\delta)$ shrinks to a singleton $\boldsymbol\beta_0$, as $N_b\rightarrow\infty$. Condition 2 is necessary for both consistency and asymptotic normality. Both condition 1 and condition 2 are the standard regularity conditions that were assumed by Fahrmeir and Kaufmann (1985). Different from the traditional MLE, the consistency for the renewable estimator requires the continuity assumption 3 to be held over the whole parameter space Θ, rather than over a neighbourhood of $\boldsymbol\beta_0$. Since, in the GLMs, the matrix $\mathcal{I}_{N_b}(\boldsymbol\beta)$ 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 1. Under conditions 1–3, the renewable estimator $\tilde{\beta}_b$ given in equation (10) is consistent, namely $\tilde{\beta}_b \to^{\mathrm{p}} \beta_0$, as $N_b = \Sigma_{j=1}^{b} n_j \to \infty$.

The proof of theorem 1 is given in Appendix A.1.

Theorem 2. Under conditions 1–3, the renewable estimator $\tilde{\beta}_b$ is asymptotically normally distributed, i.e.

$$\sqrt{N_b}(\tilde{\beta}_b - \beta_0) \xrightarrow{\mathrm{d}} \mathcal{N}(0, \boldsymbol{\Sigma}_0), \qquad \text{as } N_b = \sum_{j=1}^{b} n_j \to \infty,$$

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

The proof of theorem 2 is provided in Appendix A.2. It is interesting that the asymptotic covariance matrix of the renewable estimator $\tilde{\beta}_b$ that is given in theorem 2 is the same as that of the oracle MLE $\hat{\beta}_b^*$. This implies that the renewable estimator proposed is fully efficient; see also remark 2 below. With no need for historical subject level data in the computation, using only the prior aggregated negative Hessian matrix stored in the rho architecture, $\tilde{\mathbf{J}}_b = \Sigma_{j=1}^{b} \mathbf{J}_j(D_j; \tilde{\beta}_j)$, we calculate the estimated asymptotic covariance matrix $\tilde{\boldsymbol{\Sigma}}_b$ given by $\tilde{\boldsymbol{\Sigma}}_b = \{(N_b \tilde{\phi}_b)^{-1} \Sigma_{j=1}^{b} \mathbf{J}_j(D_j; \tilde{\beta}_j)\}^{-1} = N_b \tilde{\phi}_b \tilde{\mathbf{J}}_b^{-1}$. It follows that the estimated variance matrix for $\tilde{\beta}_b$ is given by

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

Remark 2. Because both SGD and AISGD 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 AISGD (Toulis and Airoldi, 2015). Theorem 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 $\tilde{\beta}_b$, which is shown to be asymptotically equivalent to the oracle MLE $\hat{\beta}_b^*$.

Theorem 3. Under conditions 1–3, the l_2-norm difference between the oracle MLE $\hat{\beta}_b^*$ and the proposed renewable estimator $\tilde{\beta}_b$ vanishes at the rate of N_b^{-1}, namely

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

Theorem 3 implies that the renewable estimator achieves optimal efficiency. The proof of theorem 3 is included in Appendix A.3.

4.2. Incremental inference

The Wald test based on the asymptotic distribution of the renewable estimator in theorem 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 β_1^{null}, define the following null hypothesis parameter space $\Theta_{H_0} = \{(\beta_1, \beta_2) = (\beta_1^{\mathrm{null}}, \beta_{k+1}, \ldots, \beta_p)\}$, a $(p-k)$-dimensional subspace of Θ. The subvector $\tilde{\beta}_{1b}$ of $\tilde{\beta}_b$ corresponding to its first k parameters follows asymptotically a k-dimensional marginal normal distribution, according to theorem 2. Specifically, a suitable block partition of the estimate $\tilde{\boldsymbol{\beta}}_b$ and its asymptotic variance matrix are given by respectively $\tilde{\boldsymbol{\beta}}_b = (\tilde{\boldsymbol{\beta}}_{1b}^{\mathrm{T}}, \tilde{\boldsymbol{\beta}}_{2b}^{\mathrm{T}})^{\mathrm{T}}$ and $\boldsymbol{\Sigma}_0 = [\boldsymbol{\Sigma}_{ij}]_{i,j=1,2}$: a 2×2 block matrix. Under the null hypothesis $H_0 : \boldsymbol{\beta}_1 = \boldsymbol{\beta}_1^{\mathrm{null}}$, $\sqrt{}N_b(\tilde{\boldsymbol{\beta}}_{1b} - \boldsymbol{\beta}_1^{\mathrm{null}}) \rightarrow^{\mathrm{d}} \mathcal{N}_k(\mathbf{0}, \boldsymbol{\Sigma}_{11})$, as $N_b \rightarrow \infty$. This gives rise to the following asymptotic χ^2-distribution with k degrees of freedom, i.e. under the null H_0,

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

where $(\tilde{\phi}_b \tilde{\mathbf{J}}_b^{-1})_{11}$ is the $(1, 1)$-block of matrix $\tilde{\mathbf{V}}(\tilde{\boldsymbol{\beta}}_b)$ in equation (12). Thus, a $100(1-\alpha)\%$ confidence ellipsoid for $\boldsymbol{\beta}_1$ is given by

$$\mathcal{C} = \{\boldsymbol{\beta}_1 : (\tilde{\boldsymbol{\beta}}_{1b} - \boldsymbol{\beta}_1)^{\mathrm{T}} \{(\tilde{\phi}_b \tilde{\mathbf{J}}_b^{-1})_{11}\}^{-1} (\tilde{\boldsymbol{\beta}}_{1b} - \boldsymbol{\beta}_1) < \chi_k^2(\alpha)\}.$$

It is worth pointing out that Rao's score test and Wilks's 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 by-product of theorem 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's likelihood ratio test is an open problem in the setting of streaming data analysis.

5. Implementation

5.1. Rho architecture and pseudocode

The renewable analytics proposed may be implemented in the rho architecture in Fig. 2. The workflow chart in Fig. 3 facilitates the organization of the pseudocode for key numerical calculations, summarized by algorithm 1 in Table 3.

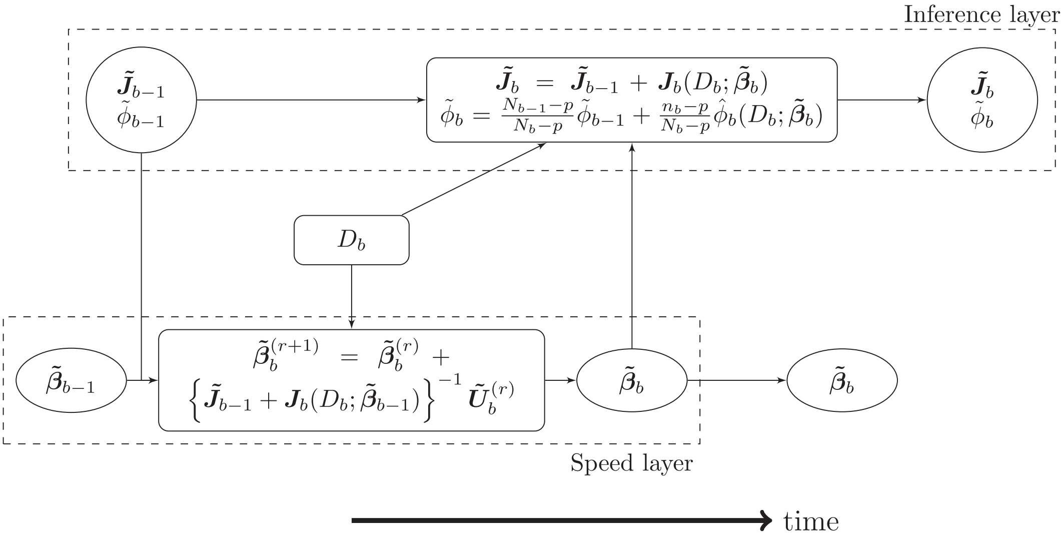
Inference layer
$\tilde{\boldsymbol{J}}_{b-1}$
$\tilde{\phi}_{b-1}$
$\tilde{\boldsymbol{J}}_b = \tilde{\boldsymbol{J}}_{b-1} + \boldsymbol{J}_b(D_b; \tilde{\boldsymbol{\beta}}_b)$
$\tilde{\phi}_b = \frac{N_{b-1}-p}{N_b-p} \tilde{\phi}_{b-1} + \frac{n_b-p}{N_b-p} \hat{\phi}_b(D_b; \tilde{\boldsymbol{\beta}}_b)$
$\tilde{\boldsymbol{J}}_b$
$\tilde{\phi}_b$
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 $\tilde{\beta}_{b-1}$ is updated to $\tilde{\beta}_b$ at the speed layer and $(\tilde{\mathbf{J}}_{b-1}, \tilde{\phi}_{b-1})$ are updated to $(\tilde{\mathbf{J}}_b, \tilde{\phi}_b)$ at the inference layer

Table 3. Algorithm 1: implementation of the renewable analytics in the rho architecture

```
1 Inputs: model $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}_0, \phi_0)$, streaming data sets $D_1, \ldots, D_b, \ldots$
2 Outputs: $\tilde{\boldsymbol{\beta}}_b$ and $\tilde{\mathbf{V}}(\tilde{\boldsymbol{\beta}}_b)$, for $b = 1, 2, \ldots$
3 Initialize: set initial values $\tilde{\boldsymbol{\beta}}_{\text{init}}$, $\tilde{\phi}_0 = 0$ and $\tilde{\mathbf{J}}_0 = \mathbf{0}_{p \times p}$
4 for $b = 1, 2, \ldots$ do
5     read in data set $D_b$
6     at the inference layer, perform Cholesky decomposition of $\{\tilde{\mathbf{J}}_{b-1} + \mathbf{J}_b(D_b; \tilde{\boldsymbol{\beta}}_{b-1})\}$ 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:
          $\tilde{\boldsymbol{\beta}}_b^{(r+1)} = \tilde{\boldsymbol{\beta}}_b^{(r)} + \{\tilde{\mathbf{J}}_{b-1} + \mathbf{J}_b(D_b; \tilde{\boldsymbol{\beta}}_{b-1})\}^{-1} \{\tilde{\mathbf{J}}_{b-1}(\tilde{\boldsymbol{\beta}}_{b-1} - \tilde{\boldsymbol{\beta}}_b^{(r)}) + \mathbf{U}_b(D_b; \tilde{\boldsymbol{\beta}}_b^{(r)})\}$, until
          convergence
8     at the inference layer, update both $\tilde{\mathbf{J}}_b = \tilde{\mathbf{J}}_{b-1} + \mathbf{J}_b(D_b; \tilde{\boldsymbol{\beta}}_b)$ 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 $\tilde{\mathbf{V}}(\tilde{\boldsymbol{\beta}}_b) = \tilde{\phi}_b \tilde{\mathbf{J}}_b^{-1}$
9     save $\tilde{\boldsymbol{\beta}}_b$ at the speed layer, and $\tilde{\mathbf{J}}_b$ 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 $\tilde{\mathbf{V}}(\tilde{\boldsymbol{\beta}}_b)$, for $b = 1, 2, \ldots$
```

(a) Line 1: all streaming data sets are modelled by a homogeneous GLM with a common true parameter $\boldsymbol{\beta}_0$. Such a model automatically satisfies some of the regularity conditions that were given in Section 4.1, such as condition 3.

(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. $\tilde{\boldsymbol{\beta}}_{\text{init}} = \mathbf{0}$.

(d) Line 4: run through the on-line updating procedures along data streams.

(e) Line 6: at the inference layer, calculate the negative Hessian $\mathbf{J}_b(D_b; \tilde{\boldsymbol{\beta}}_{b-1})$ 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 the negative Hessian and the 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 batches of data is available, here we present two non-linear GLMs in that the proposed renewable analytics are needed. They are the popular logistic model for binary outcomes and log-linear model for count outcomes.

5.2.1. *Example 2 (logistic model)*

Assume data batch $D_b = \{\mathbf{y}_b, \mathbf{X}_b\}$ with binary outcomes $\mathbf{y}_b = (y_{b1}, \ldots, y_{bn_b})^{\mathrm{T}}$ and covariates $\mathbf{X}_b = (\mathbf{x}_{b1}, \ldots, \mathbf{x}_{bn_b})^{\mathrm{T}}$, where $y_{bi}|\mathbf{x}_{bi}$ are independently sampled from a Bernoulli distribution with probability of success $\pi_{bi} = P(y_{bi} = 1|\mathbf{x}_{bi})$, and dispersion parameter $\phi = 1$. A logistic model takes the form

$$g(\pi_{bi}) = \log\left(\frac{\pi_{bi}}{1 - \pi_{bi}}\right) = \mathbf{x}_{bi}^{\mathrm{T}} \boldsymbol{\beta}.$$

The score function and negative Hessian matrix (or the observed information matrix) for data batch D_b are respectively given by

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

$$\mathbf{J}_b(\boldsymbol{\beta}) = \sum_{i=1}^{n_b} v_{bi} \mathbf{x}_{bi} \mathbf{x}_{bi}^{\mathrm{T}},$$

where $v_{bi}(\pi_{bi}) = \pi_{bi}(1-\pi_{bi}) = \exp(\mathbf{x}_{bi}^{\mathrm{T}}\boldsymbol{\beta})/\{1+\exp(\mathbf{x}_{bi}^{\mathrm{T}}\boldsymbol{\beta})\}^2$ is the variance function. The renewable estimate $\tilde{\boldsymbol{\beta}}_b$ and the aggregated observed information matrices $\tilde{\mathbf{J}}_b$ are updated according to the procedure given in algorithm 1 under the rho architecture in Fig. 3.

5.2.2. Example 3 (Poisson log-linear model)
Consider $D_b = \{\mathbf{y}_b, \mathbf{X}_b\}$ with outcomes of counts $\mathbf{y}_b = (y_{b1}, \ldots, y_{bn_b})^{\mathrm{T}}$ and covariates $\mathbf{X}_b = (\mathbf{x}_{b1}, \ldots, \mathbf{x}_{bn_b})^{\mathrm{T}}$. Assume that $y_{bi}|\mathbf{x}_{bi}$ are independently sampled from a Poisson distribution with mean $\mu_{bi} = \mathbb{E}(y_{bi}|\mathbf{x}_{bi})$ that is specified by a log-linear model $g(\mu_{bi}) = \log(\mu_{bi}) = \mathbf{x}_{bi}^{\mathrm{T}}\boldsymbol{\beta}$. 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 $\mathbf{U}_b(\boldsymbol{\beta}) = \Sigma_{i=1}^{n_b} \mathbf{x}_{bi}\{y_{bi} - \exp(\mathbf{x}_{bi}^{\mathrm{T}}\boldsymbol{\beta})\}$ and $\mathbf{J}_b(\boldsymbol{\beta}) = \Sigma_{i=1}^{n_b} v_{bi}\mathbf{x}_{bi}\mathbf{x}_{bi}^{\mathrm{T}}$, where $v_{bi} = \mu_{bi} = \exp(\mathbf{x}_{bi}^{\mathrm{T}}\boldsymbol{\beta})$ is the variance function. Again, the renewable estimate $\tilde{\boldsymbol{\beta}}_b$ and the aggregated observed information matrices $\tilde{\mathbf{J}}_b$ are produced in the rho architecture (Fig. 3) respectively at the speed layer and the inference layer via algorithm 1.

6. Simulation experiments

6.1. Set-up

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

(a) the oracle MLE obtained by processing the entire data once,
(b) AISGD,
(c) the sequential estimation method of the OLSE in the linear model and
(d) the sequential estimation method of the CEE or CUEE for non-linear GLMs.

Comparisons concern the aspects of parameter estimation, computational efficiency and hypothesis testing. The evaluation criteria for parameter estimation include

(a) the absolute bias Abias,
(b) the averaged estimated standard error ASE,
(c) the empirical standard error ESE and
(d) the coverage probability CP.

We use the MLE yielded by the R package `glm` as the gold standard in all comparisons. For the AISGD method, we use the R package `sgd` with one-dimensional learning rate (Xu, 2011) and hyperparameters set at $\alpha = 1$, $\gamma_0 = 1$ and $c = \frac{2}{3}$. Following Fang (2019), we set $S = 200$ bootstrap samples. Computational efficiency is also assessed by

(e) computation time CTime and
(f) running time RTime.

Rtime accounts only for the data processing time, whereas Ctime includes time that is spent on both loading data streams and processing data. In the case of AISGD, one data point is run at one iteration; thus it is difficult to capture the data loading time properly. In this case, we consider only Rtime for AISGD.

In all the simulation experiments that are considered in Tables 4–6, we set a terminal point B. We generate the full data set D_B^* with N_B observations independently from the respective GLMs with the mean model $\mathbb{E}(y_i|\mathbf{x}_i)=g(\mathbf{x}_i^{\mathrm{T}}\beta_0)$, $i=1,\ldots,N_B$. We set $\beta_0=(0.2,\,-0.2,0.2,\,-0.2,0.2)^{\mathrm{T}}$, the intercept $\mathbf{x}_{i[1]}\equiv 1$, and $\mathbf{x}_{i[2:5]}\sim\mathcal{N}_4(\mathbf{0},\mathbf{V}_4)$ independently where $\mathbf{V}_4$ is a 4×4 compound symmetry covariance matrix with correlation $\rho=0.5$.

6.2. Evaluation of parameter estimation

6.2.1. 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 are

(a) the MLE,
(b) AISGD,
(c) the OLSE for the linear model, or the CEE or CUEE for the logistic model and
(d) renewable estimation, Renew.

We generate B data streams consisting of $N_B=|D_B^*|=100000$ independent observations, each batch with n_b observations. Tables 4 and 5 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 on-line supplementary material section S3.

6.2.1.1. Bias and coverage probability. In the linear model, because the LSE is a linear function of data, it can be perfectly decomposed across data batches. Thus, the MLE, OLSE and Renew are identical, leading to exactly the same bias and coverage probability, as shown in Table 4. It is easy to see that neither the bias nor the coverage probability in the linear model is affected by data batch size n_b. From Table 5 with the regression, our renewable estimator always exhibits similar performances to the oracle MLE and appears quite robust to different n_b. In contrast, the CEE method appears numerically unstable; as the batch size n_b decreases to 200, its coverage probability drops below 90%. Even though the CUEE method is proposed to improve the CEE (Schifano *et al.*, 2016), the bias of the CUEE estimator appears much larger than that of the MLE as n_b decreases to 50. In addition, the CUEE method has much larger empirical standard error than that of the CEE estimator as n_b grows smaller. AISGD 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 estimator. Even though the coverage probability of AISGD by Fang's method is 0.92, which is 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 estimation method. See also the on-line supplementary Tables S1–S3.

6.2.1.2. Computation time. Two metrics are used to evaluate computational efficiency. CTime in Table 4 (see also in the supplementary Tables S2 and S3) refers to the total amount of time required by data loading and algorithm execution, whereas RTime is the amount of time that is required only for algorithm execution. With an increased B, our renewable estimation method clearly outperforms the three competitors, MLE, the CEE and the CUEE. AISGD appears computationally very competitive, because it avoids matrix inversion calculation in the algorithm. However, this high computing speed pays the price for significantly big estimation

Table 4. Simulation results under the linear model summarized over 500 replications, with fixed $N_B = 100000$ and $p = 5$ with varying batch sizes n_b†

	Results for AISGD,	*Results for MLE*			*Results for OLSE*			*Results for Renew*		
	$n_b = 1$	$n_b = 1000$	$n_b = 200$	$n_b = 50$	$n_b = 1000$	$n_b = 200$	$n_b = 50$	$n_b = 1000$	$n_b = 200$	$n_b = 50$
Abias×10^{-3}	*13.48*	3.17	3.17	3.17	3.17	3.17	3.17	3.17	3.17	3.17
ASE×10^{-3}	*15.08*	3.83	3.83	3.83	3.83	3.83	3.83	3.83	3.83	3.83
ESE×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
CTime (s)	—	0.56	1.68	5.91	0.08	0.19	0.66	0.12	0.34	1.27
RTime (s)	0.14	0.32	0.30	0.29	0.02	0.07	0.28	0.07	0.24	0.95

†‘Abias’, ‘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. ‘CTime’ and ‘RTime’ respectively denote computation time and running time.

Table 5. Simulation results, summarized from 500 replications, under the setting of $N_B = 100000$ and $p = 5$ for the logistic model with varying batch size n_b

	Results for AISGD,	*Results for MLE,*	*Results for CEE*			*Results for CUEE*			*Results for Renew*		
	$n_b = 1$	$n_b = 10^5$	$n_b = 1000$	$n_b = 200$	$n_b = 50$	$n_b = 1000$	$n_b = 200$	$n_b = 50$	$n_b = 1000$	$n_b = 200$	$n_b = 50$
Abias×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×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×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

Table 6. Comparison between different estimators in the logistic model with fixed batch size $n_b = 100$ and $p = 5$†

	Results for $B=10$ and $N_B=10^3$					*Results for $B=100$ and $N_B=10^4$*				
	MLE	*AISGD*	*CEE*	*CUEE*	*Renew*	*MLE*	*AISGD*	*CEE*	*CUEE*	*Renew*
Abias$\times 10^{-3}$	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
CTime (s)	0.01	—	0.03	0.06	0.01	0.08	—	0.34	0.63	0.07
RTime (s)	0.007	0.008	0.028	0.056	0.006	0.045	0.064	0.311	0.599	0.047
	Results for $B=10^3$ and $N_B=10^5$					*Results for $B=10^4$ and $N_B=10^6$*				
Abias$\times 10^{-3}$	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
CTime (s)	2.88	—	3.056	5.74	0.64	343.5	—	32.60	56.51	6.46
RTime (s)	0.51	0.19	2.84	5.50	0.47	7.04	0.98	28.85	54.04	4.66

†N_B increases from 10^3 to 10^6. Results are summarized from 500 replications.

bias, leading to problematic statistical inference. As pointed out above, we cannot evaluate the data loading time for AISGD, since it passes one single data point at a time. The on-line supplementary Fig. S1 presents a pictorial summary of all the results that were obtained in simulation scenario 1.

6.2.2. *Scenario 2: fixed batch size n_b but varying B*

Now we turn to an interesting scenario where streaming data sets 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 6 lists the summaries of simulation results under the logistic model.

6.2.2.1. Bias and coverage probability. When the batch size is as small as $n_b = 100$, increasing N_B does not seem to help to reduce the estimation bias of the CEE or CUEE. In effect, their bias is exacerbated as more data streams are processed, resulting in clearly problematic performances on statistical inference. When the number of batches of data B increases to 1000, the coverage probability by the CEE or CUEE methods remains steadily below 90%, with no sign of improvement in response to increased volumes of data. It is striking that, when B is further increased to 10^4, the coverage probability of the CUEE falls to 67%, whereas the CEE gives the worst 0% coverage probability. This confirms that when the condition $B = \mathcal{O}(n_j^k)$, $k < \frac{1}{3}$, is violated, the CEE or CUEE methods will not have valid asymptotic distributions for inference. In contrast, our proposed method confirms large sample properties which are 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 AISGD, the estimated standard error is much smaller than the empirical standard error and the coverage probability is only 83% when $N_B = 10^3$. When N_B reaches 10^5 the coverage probability improves to around 95%. However, both the bias and the estimated standard errors are much larger than those of

MLE or our renewable estimation method, suggesting that AISGD does not provide efficient inference. Moreover, its bias stops decreasing after a certain level. For example, its bias remains at 23.44×10^{-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 AISGD reaches a convergence phase, the subsequent estimates will jitter around the true parameter within a ball of slowly decreasing radius.

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

6.2.3. 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 $\mathbf{x}_i \sim^{\mathrm{IID}} \mathcal{N}(0, N_B^{-1}\mathbf{I}_p)$. Following Sur and Candés (2019), to guarantee the existence of the MLE in such high dimensional settings, we generate the true values of $\boldsymbol{\beta}_0$ entrywise IID from $\mathcal{N}(10, 900)$ under $p = 1000$ and from $\mathcal{N}(10, 300)$ under $p = 2500$. The same criteria are used in the subsequent assessment and comparisons.

6.2.3.1. Bias and coverage probability. Table 7 summarizes the simulation results over 200 replications. Our renewable estimation method has the same level of bias as the oracle MLE in this high dimensional logistic regression. In this setting with $n_j \leqslant 10p$, both the CEE and the CUEE methods fail to provide reliable coverage probabilities because of severely large biases. AISGD has the largest bias, more than 10 times that of the MLE, largely because the AISGD updates may become trapped locally. Consequently, standard errors are not properly estimated by 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.

6.2.3.2. Computation time. For large $p = 1000$ or $p = 2500$, our renewable estimation method is at least fourfold faster than the oracle MLE, and this computational efficiency is repeated in the low dimension case ($p = 5$) shown in Table 6. Although AISGD runs faster than our renewable estimation method, it is not applicable to the setting with very large p. The resulting severe bias hampers reliable estimation or valid inference.

In summary, these 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 on-line methods work only in some cases. For example, AISGD gives proper coverage probability only when B is large and p is small, whereas the CEE or CUEE produces valid inference when both B and p are small. Such evidence further demonstrates the

Table 7. Comparisons between the various estimators in the logistic model with fixed $N_B = 2\times 10^5$, $n_b = 10^4$ and $B = 20$†

	Results for the following methods				
	AISGD	*MLE*	*CEE*	*CUEE*	*Renew*
p=1000					
Abias	25.799	2.176	3.880	2.242	2.152
ASE	1.70×10^{-3}	2.705	2.904	2.668	2.707
ESE	1.72×10^{-3}	2.715	2.358	2.616	2.673
CP	*0*	0.948	0.757	0.937	0.951
CTime (min)	—	17.959	17.288	20.470	4.207
RTime (min)	1.609	16.686	17.093	20.258	4.014
p=2500					
Abias	16.386	2.212	6.994	2.581	2.192
ASE	1.71×10^{-3}	2.728	3.475	2.523	2.789
ESE	1.72×10^{-3}	2.745	1.804	2.442	2.715
CP	*0*	0.946	0.561	0.874	0.954
CTime (min)	—	126.407	122.528	149.411	31.451
RTime (min)	4.737	123.904	122.037	148.924	30.917

†The number of covariates, p, varies from 1000 to 2500.

usefulness of our method in interim analyses over the course of data streams. As far as computational efficiency is concerned, the method proposed is clearly superior to existing methods when data streams arrive at a high speed. Note that the running time complexity of our method is $O(N_B p^2 + Bp^3/3)$. When $p < n_b$, it reduces to $O(N_B p^2)$. This is a typical order for a second-order on-line method. When N_B is fixed and p is large, increasing the 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 4 and 7, as well as in 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 that is available in the inference layer in the rho architecture. We run a simulation study on the Wald test for $H_0:\beta_{01}=0.2$ *versus* $H_A:\beta_{01}\neq 0.2$, where β_{01} is the intercept parameter in the logistic model used in Tables 5 and 6. With $\boldsymbol{\beta}^{\text{null}} = (0.2, -0.2, 0.2, -0.2, 0.2)^{\text{T}}$, set $\boldsymbol{\beta}_a = (\beta_{a1}, -0.2, 0.2, -0.2, 0.2)^{\text{T}}$ with β_{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. On the basis of simulated data streams, with $N_B = 100000$, and each batch size $n_b = 200$, we calculated the empirical type I error and power from 500 replications.

Under H_0, as shown in the (1,1)-panel of Fig. S2 in the on-line supplementary material, the *Q–Q*-plot of 500 replicates of the Wald test statistic stays closely along the 45° diagonal, indicating the validity of an asymptotic χ^2_1-distribution. In addition, we increased the number of coefficients in the test and found that under H_0 the Wald statistics all behave approximately as a χ^2-distribution; see the other plots of Fig. 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, AISGD and our proposed Wald test are very close to the nominal

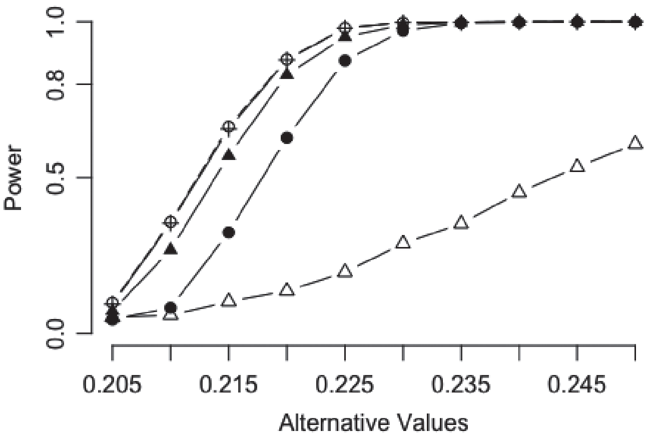

Fig. 4. Power curves of the Wald tests based on the MLE (○), AISGD (△), CEE (●), CUEE (▲) and renewable estimation (+), under a sequence of alternative values for the intercept β_1

level of 0.05, whereas the Wald tests based on the CEE and CUEE have poor type I error control. Fig. 4 shows that the power of AISGD is steadily significantly lower than that of the proposed incremental Wald test or the MLE. In addition, the CEE or CUEE has lower power when the parameter is close to the true value 0.2, suggesting poor local power.

7. Data example

To show the usefulness of our proposed renewable estimation and inference in practice, we analysed streaming data from the National Automotive Sampling System crashworthiness data system. Our primary interest was to evaluate the effectiveness of graduated driver licensing, which is nationwide legislature for novice drivers of age 21 years or younger under various conditions of vehicles operation. In contrast, there are no operating restrictions on operating vehicles for older drivers (say, Age$\geqslant$65 years) 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 a fatal crash when an accident occurred. Three age groups were considered: 'Age $<$ 21', '21 $\leqslant$Age $<$ 65' and 'Age$\geqslant$ 65' years were coded as dummy variables in our analysis, with the middle age group as the reference. Since the number of young or old drivers who are 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. The event 'Fatality' in a crash is a binary outcome of interest, which was analysed by using a logistic model. This outcome variable was created from the variable maximum treatment in accident, ATREAT, in the database, which indicated the most intensive treatment given to a 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 $B=84$ batches of data and a total sample size $N_B=23184$ of recorded accidents in the USA. We applied our proposed method to update sequentially parameter estimates and standard errors for the regression coefficients. We assumed that the underlying risk of a 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 Fig. 5, the 95% pointwise confidence bands over the 84 batches became narrower for all regression coefficients as more data streams arrived. Figs 5(a) and 5(b) display the trace

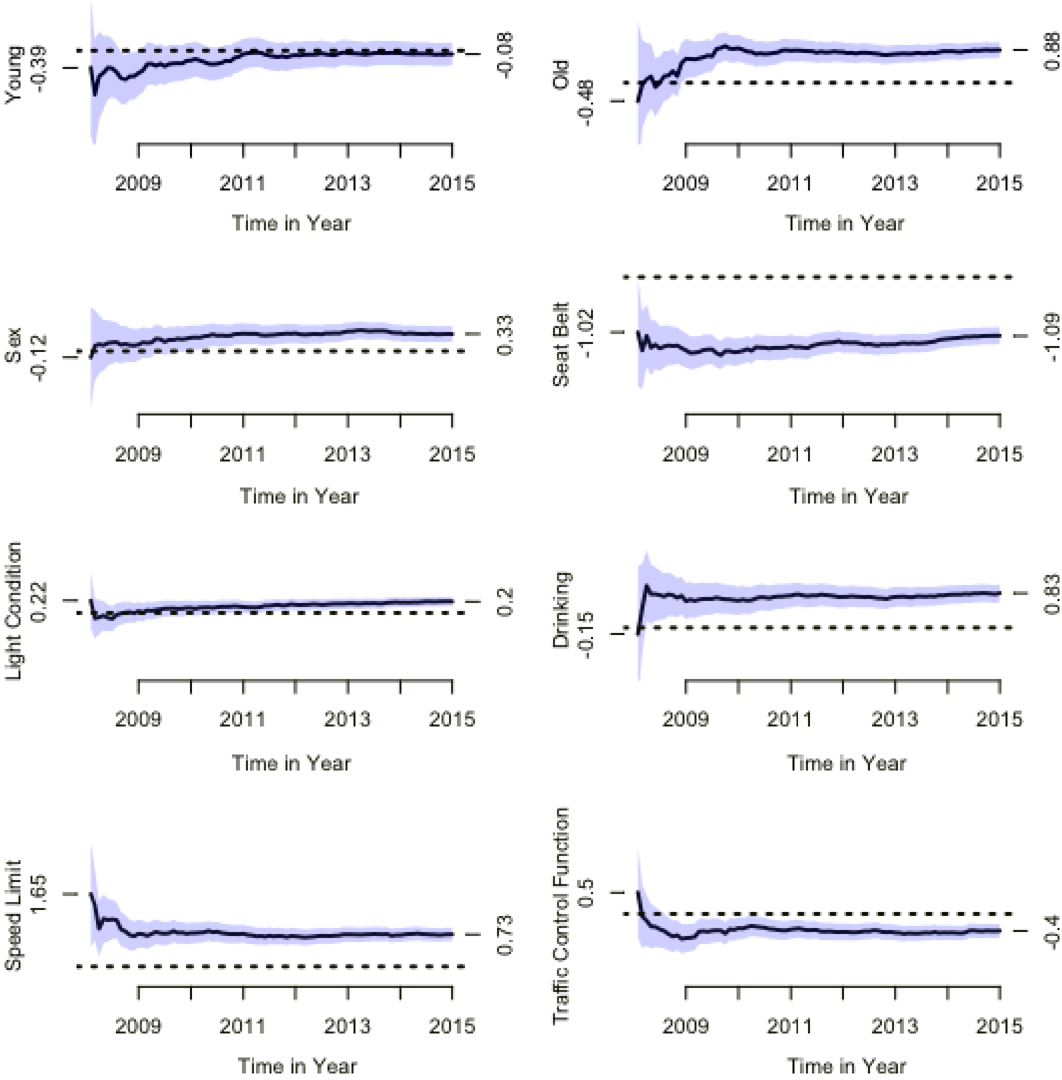

Fig. 5. Trace plots for the coefficient estimates and 95% pointwise confidence bands of regression coefficients (the numbers on each side denote the estimated regression coefficients after the arrival of the first and last batches): - - - - -, 0 reference line

plots of renewable estimates of the coefficients for the young and old age 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 a fatal crash than does the reference group. This finding is consistent with the reported results in the literature that graduated driver licensing is an effective policy to protect novice drivers from severe injury (e.g. Chen *et al.* (2014)). In contrast, the trace plot for the old age group (Age $\geqslant 65$) shows an upward trend and stabilizes when the sample size increases. This suggests that the adjusted odds of fatality in a vehicle crash for the old age group become significantly higher than for the reference group when data accumulated sufficiently large. This may suggest a need for a policy modification for a restrictive vehicle operation for old drivers.

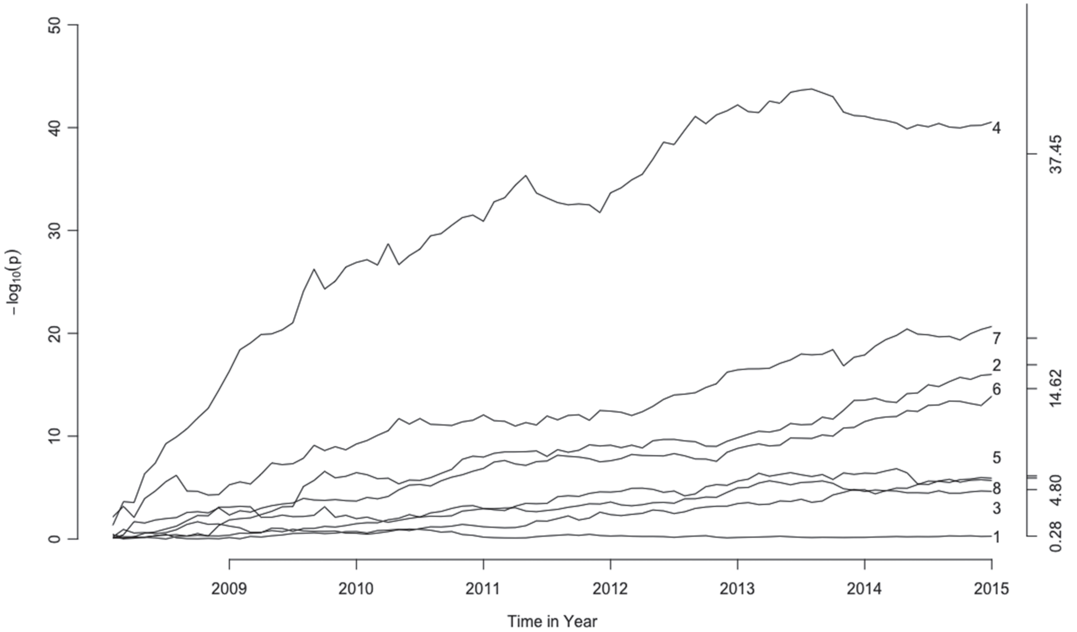

Fig. 6. Trace plot of $-\log_{10}(p)$ over monthly batches of data during January 2009–December 2015, each for one regression coefficient (numbers on the left-hand y-axis are the negative logarithm p-values obtained by the proposed incremental Wald test and labels on the x-axis correspond to the last month of each year; on the right-hand y-axis, the numerical numbers denote $-\log_{10}(p)$ obtained by the oracle MLE; the values in parentheses next to the covariate names denote the areas under the p-value curves): 1, Young (39.50); 2, Old (664.68); 3, Sex (242.49); 4, Seat belt (2582.72); 5, Light condition (186.90); 6, Drinking (589.02); 7, Speed limit (1030.77); 8, Traffic control function (324.02)

Fig. 6 shows the trends of $-\log_{10}(p)$, p-values of the incremental Wald test in the base 10 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 with the odds of fatality in a crash among all the covariates that were included in the model. This is an overwhelming confirmation for enforcement of the policy ‘buckle up’ when sitting in a moving vehicle. In addition, to characterize the overall level of significance for each covariate over the 84-month period, we proposed to calculate the summary statistic area under the p-value curve. Most of these curves have well-separated patterns, so the ranking of the overall significance by the areas calculated is well aligned with the ranking of p-values obtained at the end time of streaming data availability, namely December 2015. It is interesting that Traffic control function, Light condition and Sex are among the weakest predictors.

Applying the proposed renewable estimation and inference to the above crashworthiness data system 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 Fig. 5, at the early stage of data streams, because of 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 are continuously updated along with data streams, which eventually reached stability and reliable conclusions. Table 8 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 the inference layer has given rise to

Table 8. Results from the MLE method and the proposed renewable estimation method in the logistic model with $N=23184, p=9$ and $B=84$

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

tremendous convenience in data storage and data analytics for processing high throughput streaming data.

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) divide-and-conquer techniques (Lin and Xi, 2011; Guha *et al.*, 2012; Chen and Xie, 2014; Tang *et al.*, 2019; 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 estimation 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 the proposed statistical methodology and the computational algorithms have been justified theoretically and examined numerically in the setting of GLMs. Being a key methodology contribution, incremental inference has shown to be statistically valid and efficient. It has no loss of estimation efficiency in comparison with the oracle MLE method but is computationally much more efficient than the MLE.

Summary statistics that are involved in our proposed renewable estimation framework behave similarly to the classical sufficient statistic. Appendix A.4 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 estimation properties in terms of a sufficient statistic. This extension builds a useful theoretical connection between the classical theory of statistical sufficiency and modern on-line learning analytics. More details on the technical proofs are included in the on-line supplementary material section S2.

Through various simulation studies, we demonstrate that our proposed method runs computationally faster than two existing methods: the CEE and CUEE. Our updating algorithm keeps using the same inverse Hessian matrix over all the iterations, which is only computed once per batch of data. It is worth pointing out once again that the consistency of estimation of the 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 a condition may not hold in some real applications when data streams arrive perpetually. Our method has overcome this restriction and produces stable, reliable and efficient solutions to the three questions that were raised in Section 1. Thus, our method is appealing practically. Reliability of statistical inference is of great importance in practice to handle data streams, such as phase IV clinical trials where drug safety, side effects and efficacy must 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 estimation 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) have been 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 sufficient statistics needed, $X^{\mathrm{T}}WX$ and XWZ, in sequential increments and then summing them up in the iteratively weighted least square algorithm. However, these two methods must use historical subject level data in calculations. Thus, they are more expensive in data storage and are computationally inefficient in comparison with 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 analysing very large static data.

The formulation of renewable estimation analytics is in the context of GLMs where the log-likelihood functions have nice properties such as twice continuously differentiability. Both theoretical and numerical experiences learned from the GLMs in this paper shed light on further generalization of such methods to other important settings such as generalized estimating equations, Cox regression and quantile regression. In addition, our method is based on the assumption that batches of data are all sampled from a homogeneous study population, which may be violated in some 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 the Joint 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 US National Science Foundation grant DMS1811734 and National Institutes of Health grant R01ES024732.

Appendix A

A.1. Proof of consistency

Assume that conditions 1–3 given in Section 4.1 hold. The MLE of the cumulative data set to time point b is

$$\hat{\boldsymbol{\beta}}_b^* = \arg\max_{\beta\in\mathbb{R}^p} l_{N_b}(\boldsymbol{\beta}, \phi; D_b^*).$$

Under condition 2, i.e. $\mathcal{I}_{N_b}(\boldsymbol{\beta})$ is positive definite, there is a unique solution to the score equation $\Sigma_{j=1}^{b}\mathbf{U}_j(D_j;\boldsymbol{\beta})=0$, which is the MLE $\hat{\boldsymbol{\beta}}_b^*$ for this cumulative data set.

Let $\boldsymbol{\beta}_0$ be the true parameter and $\tilde{\boldsymbol{\beta}}_b$ be the renewable estimator. For the prior data batch D_1, we have $\tilde{\boldsymbol{\beta}}_1=\hat{\boldsymbol{\beta}}_1^*=\hat{\boldsymbol{\beta}}_1$, 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 $b\geqslant 2$ by the method of induction.

Define a function

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

According to equation (10), the renewable estimator $\tilde{\boldsymbol{\beta}}_b$ satisfies

$$f_b(\tilde{\boldsymbol{\beta}}_b)=\mathbf{0}. \tag{14}$$

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

$$f_b(\boldsymbol{\beta}_0)=\frac{1}{N_b}\sum_{j=1}^{b-1}\mathbf{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)(\tilde{\boldsymbol{\beta}}_{b-1}-\boldsymbol{\beta}_0)+\frac{1}{N_b}\mathbf{U}_b(D_b;\boldsymbol{\beta}_0)=o_p(1). \tag{15}$$

Taking the difference between equations (15) and (14), we obtain

$$f_b(\boldsymbol{\beta}_0)-f_b(\tilde{\boldsymbol{\beta}}_b)=\frac{1}{N_b}\sum_{j=1}^{b-1}\mathbf{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)-\frac{1}{N_b}\mathbf{U}_b(D_b;\tilde{\boldsymbol{\beta}}_b)+\frac{1}{N_b}\mathbf{U}_b(D_b;\boldsymbol{\beta}_0)=o_p(1). \tag{16}$$

Then, taking the first-order Taylor series expansion of term $\mathbf{U}_b(D_b;\tilde{\boldsymbol{\beta}}_b)$ in equation (16) around $\boldsymbol{\beta}_0$, we obtain

$$\mathbf{U}_b(D_b;\tilde{\boldsymbol{\beta}}_b)=\mathbf{U}_b(D_b;\boldsymbol{\beta}_0)-\{\mathbf{J}_b(D_b;\boldsymbol{\beta}_0)-\mathbf{J}_b(D_b;\boldsymbol{\beta}_0)+\mathbf{J}_b(D_b;\boldsymbol{\xi}_b)\}\,(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0), \tag{17}$$

where $\boldsymbol{\xi}_b$ lies in between $\tilde{\boldsymbol{\beta}}_b$ and $\boldsymbol{\beta}_0$. By the Lipschitz continuity in condition 3, there exists $M(D_b)>0$ such that

$$\|\mathbf{J}_b(D_b;\boldsymbol{\xi}_b)-\mathbf{J}_b(D_b;\boldsymbol{\beta}_0)\|\leqslant M(D_b)\|\boldsymbol{\xi}_b-\boldsymbol{\beta}_0\|\leqslant M(D_b)\|\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|. \tag{18}$$

Using inequality (18) we rewrite equation (17) as

$$\mathbf{U}_b(D_b;\tilde{\boldsymbol{\beta}}_b)=\mathbf{U}_b(D_b;\boldsymbol{\beta}_0)-\mathbf{J}_b(D_b;\boldsymbol{\beta}_0)(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)+\mathcal{O}_p(n_b\|\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|^2). \tag{19}$$

Combining equations (16) and (19) yields

$$f_b(\boldsymbol{\beta}_0)-f_b(\tilde{\boldsymbol{\beta}}_b)=\frac{1}{N_b}\left\{\sum_{j=1}^{b-1}\mathbf{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)+\mathbf{J}_b(D_b;\boldsymbol{\beta}_0)\right\}(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)+O_p\left(\frac{n_b}{N_b}\|\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|^2\right)=o_p(1). \tag{20}$$

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 2, we know that $N_b^{-1}\{\Sigma_{j=1}^{b-1}\mathbf{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)+\mathbf{J}_b(D_b;\boldsymbol{\beta}_0)\}$ is positive definite. It follows that $\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\rightarrow^{\mathrm{p}}\mathbf{0}$, as $N_b\rightarrow\infty$.

A.2. Proof of asymptotic normality

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

$$\frac{1}{N_1}\mathbf{U}_1(D_1;\boldsymbol{\beta}_0)=\frac{1}{N_1}\mathbf{J}_1(D_1;\hat{\boldsymbol{\beta}}_1)(\hat{\boldsymbol{\beta}}_1-\boldsymbol{\beta}_0)+O_p\left(\frac{n_1}{N_1}\|\hat{\boldsymbol{\beta}}_1-\boldsymbol{\beta}_0\|^2\right), \tag{21}$$

where we leave $n_1/N_1=1$ in the expression for the convenience of mathematical arguments that are used in the subsequent proof.

(b) Consider updating $\tilde{\boldsymbol{\beta}}_{b-1}$ to $\tilde{\boldsymbol{\beta}}_b$. The oracle MLE $\hat{\boldsymbol{\beta}}_b^*$ for the cumulative data set D_b^* satisfies $(1/N_b)\Sigma_{j=1}^{b}\mathbf{U}_j(D_j;\hat{\boldsymbol{\beta}}_b^*)=\mathbf{0}$. Taking the first-order Taylor series expansion around $\boldsymbol{\beta}_0$ leads to

$$\frac{1}{N_b}\sum_{j=1}^{b}\mathbf{U}_j(D_j;\boldsymbol{\beta}_0)-\frac{1}{N_b}\sum_{j=1}^{b}\mathbf{J}_j(D_j;\boldsymbol{\beta}_0)(\hat{\boldsymbol{\beta}}_b^*-\boldsymbol{\beta}_0)+O_p(\|\hat{\boldsymbol{\beta}}_b^*-\boldsymbol{\beta}_0\|^2)=\mathbf{0}. \tag{22}$$

From the definition of $f_b(\boldsymbol{\beta})$, equations (14) and (20), we know that

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

It follows that

$$-\frac{1}{N_b}\left\{\sum_{j=1}^{b-1}\mathbf{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j) + \mathbf{J}_b(D_b;\boldsymbol{\beta}_0)\right\}(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) + \frac{1}{N_b}\sum_{j=1}^{b-1}\mathbf{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)(\tilde{\boldsymbol{\beta}}_{b-1} - \boldsymbol{\beta}_0) + \frac{1}{N_b}\mathbf{U}_b(D_b;\boldsymbol{\beta}_0)$$
$$+ O_p\left(\frac{n_b}{N_b}\|\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0\|^2\right) = \mathbf{0}. \qquad (23)$$

Similarly to equation (21), at the $(b-1)$th data batch, it is easy to show that

$$\frac{1}{N_{b-1}}\sum_{j=1}^{b-1}\mathbf{U}_j(D_j;\boldsymbol{\beta}_0) = \frac{1}{N_{b-1}}\sum_{j=1}^{b-1}\mathbf{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)(\tilde{\boldsymbol{\beta}}_{b-1} - \boldsymbol{\beta}_0) + O_p\left(\sum_{j=1}^{b-1}\frac{n_j}{N_{b-1}}\|\tilde{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_0\|^2\right). \qquad (24)$$

Plugging equation (24) into equation (23), we obtain

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

Since, according to theorem 1, all $\tilde{\boldsymbol{\beta}}_j$ are consistent for $j = 1, \ldots, b-1$, and, by condition 3, the continuous mapping theorem implies that

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

Furthermore, since $\tilde{\phi}_b$ is a consistent estimator of ϕ_0 because of the weak law of large numbers, we have $(1/N_b)\tilde{\phi}_b^{-1}\Sigma_{j=1}^{b}\mathbf{J}_j(D_j;\boldsymbol{\beta}_0) \rightarrow^{\mathrm{p}} \boldsymbol{\Sigma}_0^{-1}$, $N_b \rightarrow \infty$. By condition 2, $\mathcal{I}_{N_b}^{-1}(\boldsymbol{\beta}_0)$ exists, and thus the central limit theorem implies that

$$\sqrt{N_b}(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) = \left\{\sum_{j=1}^{b}\mathbf{J}_j(D_j;\boldsymbol{\beta}_0)\right\}^{-1}\frac{1}{\sqrt{N_b}}\sum_{j=1}^{b}\mathbf{U}_j(D_j;\boldsymbol{\beta}_0) + o_p(1) \overset{\mathrm{d}}{\rightarrow} \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_0), \qquad N_b \rightarrow \infty. \qquad (25)$$

A.3. Proof of asymptotic equivalency

Now we prove theorem 3. The difference of the two equations (22) and (17) suggests that

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

Theorem 2 or equation (25) implies that $\|\tilde{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_0\|^2 = O_p(1/N_j)$, $j = 1, \ldots, b$. By condition 2, it is easy to see that

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

A.4. Approximate sufficient statistic

To understand what types of summary statistics are suitable 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 $D = \{d_i\}_{i=1}^{n} \sim^{\mathrm{IID}} f(d;\boldsymbol{\beta}_0,\phi_0)$ denote a set of random samples of size n, and $f_n(D;\boldsymbol{\beta}_0,\phi_0)$ is the joint probability density function or probability mass function of

D. Suppose that the nuisance parameter ϕ_0 is unknown and consistently estimated by $\hat{\phi}_n$, namely $\hat{\phi}_n \rightarrow^{\mathrm{p}} \phi_0$. Let $\mathcal{B}_n(\delta)$ be a neighbourhood 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 are functions $g\{S_n(D); \boldsymbol{\beta}\}$ 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 $g_n(D; \boldsymbol{\beta}, \hat{\phi}_n) = g\{S_n(D); \boldsymbol{\beta}\} + o_p(1)$. In particular, when the nuisance parameter ϕ_0 is known, the factorization expression reduces to $f_n(D; \boldsymbol{\beta}) = g_n\{D; \boldsymbol{\beta}\} c_n(D)$, with $g_n(D; \boldsymbol{\beta}) = g\{S_n(D); \boldsymbol{\beta}\} + o_p(1)$.

This definition is well suited to the logistic model and Poisson model with $\phi_0 = 1$, as well as the linear model or gamma model with an unknown ϕ_0. In the latter case, we replace the nuisance parameter ϕ_0 with an unbiased or consistent estimator in the derivation of $S_n(D)$. Thus, $\boldsymbol{\beta}$ depends on data D through $S_n(D)$ only, approximately. In the on-line supplementary material section S2, we prove that the summary statistics that are used in the proposed renewable analytics are approximate sufficient statistics in the framework of GLMs. Also, we present an interesting example of an 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. *Neurl Computn*, **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 *Proc. Int. Conf. Data Mining Wrkshp, Atlantic City, Nov. 14th–17th*, pp. 1608–1611. New York: Institute of Electrical and Electronics Engineers.

Bordes, A., Bottou, L. and Gallinari, P. (2009) Sgd-qn: careful quasi-Newton stochastic gradient descent. *J. Mach. Learn. Res.*, **10**, 1737–1754.

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

Cardot, H. and Degras, D. (2018) Online principal component analysis in high dimension: which algorithm to choose? *Int. Statist. Rev.*, **86**, 29–50.

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. *Spatl Spatio-temp. Epidem.*, **8**, 11–22.

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

Cox, D. R. and Reid, N. (1987) Paramater orthogonality and approximate conditional inference (with discussion). *J. R. Statist. Soc.* B, **49**, 1–39.

Duchi, J., Hazan, E. and Singer, Y. (2011) Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, **12**, 2121–2159.

Enea, M., Meiri, R. and Kalimi, T. (2015) speedglm: fitting linear and generalized linear models to large data sets. (Available from `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. *Ann. Statist.*, **13**, 342–368.

Fang, Y. (2019) Scalable statistical inference for averaged implicit stochastic gradient descent. *Scand. J. Statist.*, **46**, 987–1002.

Gaber, M. M., Zaslavsky, A. and Krishnaswamy, S. (2005) Mining data streams: a review. *ACM SIGMOD Rec.*, **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. *Int. Conf. Data Mining, Barcelona*.

Hazan, E., Agarwal, A. and Kale, S. (2007) Logarithmic regret algorithms for online convex optimization. *J. Mach. Learn. Res.*, **69**, 169–192.

Jørgensen, B. (1997) *The Theory of Dispersion Models*. London: Chapman and Hall.

Kleiner, A., Talwalkar, A., Sarkar, P. and Jordan, M. I. (2014) A scalable bootstrap for massive data. *J. R. Statist. Soc.* 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. *J. Am. Statist. Ass.*, **108**, 325–339.

Lin, N. and Xi, R. (2011) Aggregated estimating equation estimation. *Statist. Interfc.*, **4**, 73–83.

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

Lumley, T. (2013) biglm: bounded memory linear and generalized linear models. University of Auckland, Auckland. (Available from `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. *J. Mach. Learn. Res.*, **6**, 861–911.

Marz, N. and Warren, J. (2015) *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*. New York: Manning Publications.

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

Nion, D. and Sidiropoulos, N. D. (2009) Adaptive algorithms to track the PARAFAC decomposition of a third-order tensor. *IEEE Trans. Signl Process*, **57**, 2299–2310.

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

Qamar, S., Guhaniyogi, R. and Dunson, D. B. (2018) Bayesian conditional density filtering. *J. Computnl Graph. Statist.*, **27**, 657–672.

Robbins, H. and Monro, S. (1951) A stochastic approximation method. *Ann. Math. Statist.*, **22**, 400–407.

Sakrison, D. J. (1965) Efficient recursive estimation: application to estimating the parameter of a covariance function. *Int. J. Engng Sci.*, **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. *Proc. Mach. Learn. Res.*, **2**, 436–443.

Song, P. X.-K. (2007) *Correlated Data Analysis*. New York: Springer.

Song, P.-K., Fan, Y. and Kalbfleisch, J. (2005) Maximization by parts in likelihood inference (with discussion). *J. Am. Statist. Ass.*, **100**, 1145–1158.

Stengel, R. F. (1994) *Optimal Control and Estimation*. New York: Dover Publications.

Sur, P. and Candés, E. J. (2019) A modern maximum-likelihood theory for high-dimensional logistic regression. *Proc. Natn. Acad. Sci. USA*, **116**, 14516–14525.

Tang, L., Zhou, L. and Song, P. X.-K. (2019) Method of divide-and-combine in regularised generalised linear models for big data. *J. Multiv. Anal.*, to be published.

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

Toulis, P. and Airoldi, E. M. (2017) Asymptotic and finite-sample properties of estimators based on stochastic gradients. *Ann. Statist.*, **45**, 1694–1727.

Toulis, P., Rennie, J. and Airoldi, E. M. (2014) Statistical analysis of stochastic gradient methods for generalized linear models. *Proc. Mach. Learn. Res.*, **32**, 667–675.

Vaits, N., Moroshko, E. and Crammer, K. (2015) Second-order non-stationary online learning for regression. *J. Mach. Learn. Res.*, **16**, 1481–1517.

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

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

Supporting information

Additional ‘supporting information’ may be found in the on-line version of this article:

‘Supplementary material of “Renewable estimation and incremental inference in generalized linear models with streaming datasets”’.