Renewable Estimation and Incremental Inference with Streaming Health Datasets

by

Lan Luo

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Biostatistics)
in The University of Michigan
2020

Doctoral Committee:

Professor Peter X.K. Song, Chair
Professor Alfred O. Hero
Professor Bhramar Mukherjee
Assistant Professor Zhenke Wu

Lan Luo

luolsph@umich.edu

ORCID iD: 0000-0002-7901-2148

ACKNOWLEDGEMENTS

The past four years of my Ph.D. journey is by far the most precious time in my life. I feel so lucky that I have such chance to work on statistical methods to process streaming data, with a particular focus on statistical inference in regression analysis. It is definitely a gold period full of enjoyable, enriching and fulfilling experiences that will be a precious memory to me. It would not have been so fruitful without the help of many people, including my advisor, dissertation committee, colleagues, friends, and family.

First and foremost I want to express my deepest gratefulness to my advisor Dr. Peter X.-K. Song for his supervision and support. He is one of the most responsible mentors I have ever met. I cannot grow so fast without his instructions as a student from biology background with merely statistical training. Because of his passion about research, I am always highly motivated and gradually find that doing research is an enjoyable process full of fulfillment, and finally even choose the path in academia as my future career. Besides all his contributions of time, ideas and funding to make my Ph.D. experience production and stimulating, his enthusiasm and inspirational personalities generate a lot of positive impacts on me that will last for the rest of my life. To some extent, having been working with him during the past four years makes my Ph.D. journey meaningful, and even my whole life. I am also thankful for the excellent role model he has provided as such a wonderful mentor to students. I really wish that I would be able to maintain this as a life-long mentor relationship after graduation!

I would also like to thank Drs. Alfred O. Hero, Bhramar Mukherjee, and Zhenke Wu for serving as members of my dissertation committee and providing me invaluable suggestions on my dissertation. Especially, I am grateful to Dr. Hero for his domain expertise in electrical engineering and computer science such as online learning, for motivating me to formulate the third chapter of my dissertation in state space models that allows dynamic heterogeneity in data streams over time. This is an interesting and promising direction that I will continue working on after graduation. I am also appreciative to Dr. Mukherjee for her training on my data analysis skills dating back to a master course, and I benefit a lot from that experience in my later on collaborative and applied projects. Besides, she offered me a lot of encouragement and support whenever possible, and I felt so fortunate to have met such a successful female statistician in my early years. Dr. Wu, as a junior faculty in our department, also gave me some good suggestions on my thesis and offered me a chance to communicate my research idea with his group members.

The members of the Song Lab have contributed immensely to various projects that I have been working on. Our group has been a source of friendships as well as good advice and collaborations. Here is a list of their names: Margaret Banker, Emily Hector, Lu Tang, Xichen She, Ling Zhou and Yiwang Zhou. I am grateful to all the support they gave me such as manuscript proofreading and presentation rehearsals. I am especially grateful to Xichen, who collaborated with me on the YONO project and Ling, who generously helped me on the theoretical part in my first chapter. This is a warm family I will miss a lot especially those fall hikes, pot lucks and farewell dinners we had. Additionally, I would like to thank my department, the faculty and staff, fellow students and friends who kept company with me for six years and witnessed my transition from a biology student to statistics.

Lastly, I would like to thank my parents Shuihua Luo and Yongmei Liu for their understanding and unconditional love especially during those tough periods. Without

their support, I will not even have an opportunity to study in such top university. Thank you!

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

LIST OF APPENDICES

Appendix

ABSTRACT

New data collection and storage technologies have given rise to a new field of streaming data analytics, including real-time statistical methodology for online data analyses. Modeling and analysis of streaming health datasets become increasingly popular in the biomedical sciences and public health. Streaming data refers to high-throughput recordings with large volumes of observations gathered sequentially and perpetually over time. Such type of data includes national disease registry, mobile health, and disease surveillance, among others. This dissertation primarily concerns the development of fast real-time statistical estimation and inference for regression analysis, with a particular objective of optimizing both the streaming data storage and computational efficiency of statistical methods. Following the classical stochastic gradient descent (SGD) algorithm (*Sakrison*, 1965) and the seminal work of maximization-by-part (*Song et al.*, 2005), I develop a new regression analysis methodology that enjoys strong theoretical guarantees, including both asymptotic consistency and estimation efficiency, as well as fast computational speed.

The overarching objective of my dissertation is to develop a new methodology that allows to sequentially update parameter estimates and their standard errors along with data streams. The key technical novelty pertains to the fact that the proposed estimation method, termed as *renewable estimation* in my dissertation, uses current data and summary statistics of historical data, but no use of any historical subject-level data. This way of data operation in producing parameter estimates enables to optimize the steadily growing need in the space of streaming data storage

and sequential updates of repeated statistical analyses. To implement the renewable estimation, I utilize the powerful Lambda architecture in Apache Spark to design a new paradigm that includes an inference layer in addition to the existing speed layer. This expanded architecture is named as the Rho architecture in the dissertation, which accommodates inference-related statistics to facilitate sequential updating of quantities involved in estimation and inference.

The first project focuses on the renewable estimation in the setting of generalized linear models in which I develop a new sequential updating algorithm to calculate numerical solutions of parameter estimates and related inferential quantities. The proposed procedure aggregates both score functions and information matrices over streaming data batches through some approximate sufficient statistics. I show that the resulting estimation is asymptotically equivalent up to order 1/N in comparison to the maximum likelihood estimation (MLE) conducted with the entirely aggregated data once. An incremental Wald test is proposed to perform online statistical inference. I demonstrate this new methodology on the analysis of the National Automotive Sampling System-Crashworthiness Data System (NASS CDS) that aims to evaluate the effectiveness of graduated driver licensing (GDL) in the USA.

The second project focuses on a substantial extension of the first project to analyze streaming datasets with correlated outcomes, such as clustered data and longitudinal data. I establish the theoretical guarantees for the proposed renewable quadratic inference function (QIF) for dependent outcomes, and implement the proposed renewable QIF within the Rho architecture. Furthermore, I relax the homogeneous assumption in the first project and consider regime-switching regression models with a structural change-point. I propose a real-time hypothesis testing procedure based on a goodness-of-fit test statistic that is shown to achieve both proper type I error control and desirable change-point detection power.

The third project concerns data streams that involve both inter-data batch cor-

relation and dynamic heterogeneity, arising typically from various types of electronic health records (EHR) and mobile health data. This project is built in the framework of state space models in which the observed data stream is driven by a latent state process that may incorporate trend, seasonal, or time-varying covariate effects. In this setting, calculating the online MLE is challenge due to the involvement of high-dimensional integrals and complex covariance structures. In this project, I develop a Kalman filter to facilitate a multivariate online regression analysis (MORA) in the context of linear state space mixed models. MORA enables to renew both point estimates and standard errors of the fixed effects. We also apply the MORA method to analyze an EHR data example, adjusting for some heterogeneous batch-specific effects. The dissertation is closed with some summary remarks and future work.

CHAPTER I

Introduction

1.1 Motivation

New data collection and storage technologies have given rise to a new field of streaming data analytics, including real-time statistical methodology for online data analyses. Modeling and analysis of streaming health datasets become increasingly popular in the biomedical sciences and public health. Streaming data refers to high-throughput recordings with large volumes of observations gathered sequentially and perpetually over time. Such type of data has been often seen in national disease registry, mobile health, and disease surveillance, among others. One of the defining features for streaming data is that patient information is collected sequentially in real-time, and in some occasions the rate of data updates may be high. Unfortunately, most of currently available online learning methods focus only on point estimation, prediction and classification; statistical inference is lacking in the existing arsenal, which is essential to understand margin of error and uncertainty as part of statistical inference in data analysis. For example, in the analysis of phase IV clinical trials with streaming medical data, statistical inference (e.g. p-value) is needed to communicate with the clinical community and for drug side-effects monitoring. My dissertation aims to fill this gap of no statistical inference in regression analysis of streaming data.

My dissertation research has focused on the development of fast real-time data

analytics to perform estimation and inference in the regression analysis of streaming data, in both aspects of methodology and implementation via Spark's Lambda architecture. Regression analysis is regarded as the most widely used statistical tool in practice, so the proposed new methods and software in my dissertation will bring in a new useful toolbox to handle streaming health data analysis. In this dissertation, I plan to develop new methodologies that allow to sequentially update parameter estimates and associated standard errors along with data streams in generalized linear models with cross-sectional data and clustered data. To implement the proposed new methods, I will utilize the powerful Lambda architecture in Apache Spark (*Bifet et al.*, 2015) to design a new paradigm that includes an inference layer in addition to the existing speed layer. This expanded architecture is named as the *Rho architecture* in the dissertation, which accommodates inference-related statistics to facilitate sequential updating of the quantities involved in estimation and inference over data streams.

1.2 Statistical Challenges in Streaming Data Analyses

New technical issues arise when conventional statistical methods are applied for real-time data streams analysis. This section discusses a list of key issues that I will take into consideration when dealing with streaming data analysis. Despite being impossible to make a complete list of new technical challenges related to streaming data analysis, I present the most critical ones pertaining to regression analysis and its applications in biomedical studies. Moreover, the proposed methods in the subsequent chapters are going to address these listed challenges.

1.2.1 Real-time statistical inference

Since data streams arrive perpetually and are potentially unbounded in their sizes, it is rather challenging to either store or make inquiries from cumulatively growing

datasets. For example, a large-scale streaming database maintained by the Scientific Registry of Transplant Recipients (SRTR) is constantly updated, where every ten minutes new patients are added to the transplant waiting list. And, since the mid-2000s on average over 25,000 transplants have entered the data base yearly. Due to the lack of suitable data analytic methods, such data gathered and updated sequentially have been analyzed in a static fashion, which results in latency in the transition of data to knowledge. Also, conventional data analysis approaches with static data are often challenged by limitations in data storage and computational capacity when dealing with data that grow fast in volumes. Both analytic and computational challenges in the analysis of perpetually growing data call for reliable and efficient real-time statistical methodologies that promote timely processing of such data to generate new knowledge, so to improve clinical decision-making.

The Lambda architecture (*Marz and Warren*, 2015) is a real-time Big Data system of computing and storage with a synchronized processing of batch and stream data flows. This is the state-of-art paradigm widely used in industries to handle streaming data storage and analysis. My dissertation will utilize this platform to implement my newly proposed methods. The batch layer of this architecture provides accurate views with latency by processing all the raw data, while the speed layer calculates real-time but rough views of online data. The two views may be joined together to balance latency and throughput, and fault-tolerance. Unfortunately, so far this powerful architecture has ignored the need of real-time statistical inference. To address this weakness, in Chapters II I propose an expansion, called *Rho architecture*, that adds a new *inference layer* to compute and store inferential statistics.

1.2.2 Detection of abnormal data batches

The assumption of a fully homogeneous regression model in Project I may be violated. A sequence of streaming datasets is typically gathered by collecting individual

data batches over time. For example, streaming datasets became available quarterly from the National Automotive Sampling System-Crashworthiness Data System (NASS CDS), beginning in January, 2009. Analyzing the sequence of observational datasets based on an assumption of a common homogeneous statistical model may be just for a mathematical convenience. One of key complicating factors has to do with data contamination, which presents an issue of change-point in the regression analysis. Real-time data analysis with no quality control procedure would result in misleading conclusions. Thus, I plan to develop some adaptive and robust procedures to address such a data quality issue. Previous classical work in the field of online change-point detection have mostly deal with the simplest case in which both pre-change and post-change distributions have been fully specified; for example, Shewhart's control charts (*Amin et al.*, 1995), moving average control charts (*Amin and Search*, 1991), the CUSUM procedure (*Page*, 1954) and the Shiryayev-Roberts (SR) procedure (*Shiryayev*, 1963; *Roberts*, 1966), among others. For an unknown post-change distribution, one of popular approaches is rooted in the Generalized Likelihood Ratio (GLR) type procedure (*Goel and Wu*, 1971). However, GLR test statistic is not available when a quasi-likelihood approach is used to analyze streaming clustered data. Chapter III is devoted to the development of a new method in the framework of quadratic inference function (QIF), to detect abnormal data batches via a score-type test statistic for change-point detection.

1.2.3 Streaming dependence and dynamic heterogeneity

The assumption of independent data batches in projects I and II may be invalid in some practical studies. In some occasions, data streams may be a long time series in nature where both correlation and dynamic changes are present. For example, recent advances in health data collection technologies such as wearable devices have made mobile devices to collect individual streaming data by following individ-

ual subjects overtime. On one hand, this results in data batches that are essentially inter-correlated, leading to increasing complexity of data dependence; and on the other hand, data batches may not be homogeneous and there might be some quantities changing over time, leading to another technical challenge in streaming data analysis. Most of existing methods for long time series or longitudinal data analysis such as state space models have to be redeveloped to handle such data in an offline setting (*Jørgensen et al.*, 1999; *Song*, 2007). In project III, I will develop an online multivariate regression analysis procedure that allows to sequentially update regression analysis of serially dependent data batches. Additionally, the dynamic heterogeneity is modeled as a latent process, such as a stationary autoregressive process of order 1. The proposed new version of real-time regression analysis with heterogeneous streaming data will enjoy fast computational speed with little loss of statistical efficiency.

1.3 Summary of Objectives

Focusing on the key challenges presented above, I organize in this dissertation the methodology developments as follows.

Aim 1: To establish real-time estimation and inference methodologies in generalized linear models for independent cross-sectional data and quadratic inference functions for independent clustered data.

Aim 2: To propose an online hypothesis testing procedure to detect abnormal data batches in the framework of change-point detection with streaming clustered data.

Aim 3: To establish a multivariate online regression analysis method for real-time data streams with both inter-data batch correlation and heterogeneity.

Three projects are presented to address the above specific aims, respectively, in Chapter II, Chapter III, and Chapter IV. More details on backgrounds, literature re-

view, existing methodology and numerical illustrations can be found in the respective introduction sections of three chapters.

CHAPTER II

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

2.1 Introduction

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

easily updated with the new data batch D_2; that is,

$$\delta(D_1 \cup D_2) = \frac{1}{n_1 + n_2}\left(\sum_{i=1}^{n_1} x_{1i} + \sum_{i=1}^{n_2} x_{2i}\right) = \frac{1}{n_1 + n_2}\left(n_1\delta(D_1) + \sum_{i=1}^{n_2} x_{2i}\right). \quad (2.1)$$

The defining feature in (2.1) is that the mean estimate from the previous data, $\delta(D_1)$, rather than the dataset D_1 itself, is used in the calculation. In this paper, a statistic that satisfies such property is named as a *renewable estimator*. Indeed, the recursive operation exemplified in (2.1) works in general for many other statistics, such as sample moments and the least squares estimator in the linear model (*Stengel*, 1994). This is because all these statistics take certain linear functions of data, so that a decomposition similar to (2.1) between current and past data is feasible (see Section 2.3.3 for the details). Using only summary statistics from the previous data, instead of original historical raw data, is conceptually linked to sufficient statistic and is of critical importance in handling Big Data, as far as computing memory and speed are concerned. This strategy has been widely advocated in the literature of online learning, incremental analytics, matrix or tensor decomposition and classification, and online Bayesian inference; see *Bucak and Gunsel* (2009); *Cardot and Degras* (2015); *Nion and Sidiropoulos* (2009); *Qamar et al.* (2014), just to name a few.

Whether or not, and if so, to which extent, does the above renewability property seen in (2.1) for the case of sample mean hold in general? For example, can the maximum likelihood estimation (MLE), one of the most important statistical estimation and inference methods, may be updated sequentially in a similar fashion to the renewable procedure given in (2.1)? If not, how about MLE as a sufficient statistic? Answers to these questions are not trivial, because most of the maximum likelihood estimators are nonlinear functions of data, and often have no closed-form expressions, in that their MLE solutions can only be obtained by numerical iterative algorithms, such as Newton-Raphson. In this paper, we choose the class of generalized linear

models (GLMs) as an exemplary setting to illustrate the feasibility for an answer to the above questions. It is known that GLMs constitute a class of nonlinear models that play a central role in regression analysis, and the renewable estimation and incremental inference developed in such context will provide a useful arsenal to perform regression analysis of streaming datasets with both statistical and computational efficiency. In addition, in this setting of exponential dispersion models (*Jørgensen*, 1997), the connection between sufficient statistic and MLE may be established as part of solutions to the above questions.

The interest in developing procedures allowing "quick" updates of parameter estimates along with sequentially arrived data may be dated back five decades or so. In 1965, *Sakrison* (1965) proposed a seminal recursive estimation method that has become a very popular technique, namely the well-known *stochastic gradient descent* (SGD) algorithm that has been extensively used in the field of machine learning. The SGD method is applied for a data sequence that takes a form of an open-ending set of independent observations, $y_i \overset{i.i.d.}{\sim} f(y; \boldsymbol{\theta}_0)$, drawn from a certain statistical model $f(\cdot)$ with a fixed common unknown "true" parameter $\boldsymbol{\theta}_0$. An estimation of parameter $\boldsymbol{\theta}_0$ may be updated sequentially by a forward updating procedure, with a single data point y_i involved at each iteration: $\boldsymbol{\theta}_i^{\text{sgd}} = \boldsymbol{\theta}_{i-1}^{\text{sgd}} + \gamma_i \boldsymbol{C}_i \nabla_{\boldsymbol{\theta}} \log f(y_i; \boldsymbol{\theta}_{i-1}^{\text{sgd}})$, where $\gamma_i > 0$ is a prespecified learning rate sequence such that $i\gamma_i \to \gamma$ as $i \to \infty$ and $\{\boldsymbol{C}_i\}$ is a sequence of certain positive-definite matrices. Here $\nabla_{\boldsymbol{\theta}}$ denotes the gradient operation with respect to the model parameter $\boldsymbol{\theta}$. This updating procedure is later termed as "explicit SGD" in *Toulis et al.* (2014). Under the condition that $\gamma_i \boldsymbol{C}_i \to \boldsymbol{\mathcal{I}}(\boldsymbol{\theta}_0)^{-1}$, $i \to \infty$ where $\boldsymbol{\mathcal{I}}(\boldsymbol{\theta}_0)$ is the Fisher information matrix, this updating method enjoys the theoretical guarantees that as $i \to \infty$ the above SGD estimator $\boldsymbol{\theta}_i^{\text{sgd}}$ converges to the true parameter $\boldsymbol{\theta}_0$ with the optimal asymptotic efficiency. That is, the asymptotic covariance matrix is the inverse of the Fisher information matrix.

However, the SGD method is generally not robust to learning rate misspecification;

the algorithm may fail to converge if γ is too large. To overcome this, an improved recursive procedure is proposed by *Toulis et al.* (2014) where $\boldsymbol{\theta}_i^{\text{im}}$ appears in both sides of the updating equation, *i.e.*, $\boldsymbol{\theta}_i^{\text{im}} = \boldsymbol{\theta}_{i-1}^{\text{im}} + \gamma_i \boldsymbol{C}_i \nabla_{\boldsymbol{\theta}} \log f(y_i; \boldsymbol{\theta}_i^{\text{im}})$, called "implicit SGD". According to the comparison of these two versions of SGD algorithms in GLMs in the aspects of bias and empirical variance, *Toulis et al.* (2014) concluded that the implicit SGD appeared more robust to learning rate misspecification. To improve statistical efficiency, *Toulis et al.* (2014) further proposed the averaged implicit SGD (AI-SGD). SGD is also known as *a second-order procedure* when the learning rate is adapted according to diagonal elements of an approximated Hessian, such as *SGD-QN* (*Bordes et al.*, 2009) and *AdaGrad* (*Duchi et al.*, 2011). Although such second-order procedures can be computationally as fast as the first-order methods with $\boldsymbol{C}_i$ being the identity matrix trivially, it is not useful for statistical inference because only part of the information matrix (i.e. its diagonal elements) is updated over iterations.

There are some online second-order methods such as Natural Gradient (NG) algorithm (*Amari et al.*, 2000) and Online Newton Step (*Hazan et al.*, 2007) that maintain complete information matrices over iterations. Similar to SGD, an outer product of the first gradients is used to approximate the negative Hessian whose inverse is updated through the Sherman-Morrison formula. This updating scheme is widely used; see *Vaits et al.* (2013); *Hao et al.* (2016). However, this outer-product approximation to the Fisher information may not work well in general beyond the conventional likelihood framework due to the failing of the Bartlett Identity (*Song*, 2007, Chapter 2), and hence affects statistical inference. For online quasi-Newton methods, both Broyden-Fletcher-Goldfarb-Shanno (BFGS) (*Nocedal and Wright*, 1999) and limited memory BFGS (LBFGS) (*Liu and Nocedal*, 1989) algorithms have been modified for streaming data, respectively termed as oBFGS and oLBFGS algorithms (*Schraudolph et al.*, 2007; *Bordes et al.*, 2009). But it is unclear whether estimated approximate Hessian is appropriate for statistical inference. A detailed comparison among all these

second-order online methods is available in Table A.1 in the appendix.

Although analytic expressions have been derived for the asymptotic variances of both explicit and implicit SGD (*Toulis and Airold*, 2017), the issue of constructing confidence intervals online has remained unexplored. Recently, *Fang* (2019) proposed a perturbation-based resampling method to construct confidence intervals for AI-SGD. Even though this online-bootstrap procedure can be parallelized to improve computation efficiency, it is derived from a first-order SGD procedure that will lose statistical efficiency comparing to second-order procedures (*Toulis and Airold*, 2017). Consequently, with the same finite sample size, both bias and empirical standard error will be much larger than the one with second-order learning rate, let alone the oracle MLE. In essence, this procedure is not targeted to perform any valid interim statistical analysis and inference. It is important to note that interim statistical inference and decision making are essential in clinical studies and other related biomedical sciences (e.g., mobile health) where sequentially adaptive intervention to observed outcomes is of critical importance to increase treatment efficacy and to maximize ethical benefits. Besides, as pointed out by *Fang* (2019), it may not be used to conduct hypothesis testing which involves multiple comparisons. Furthermore, when p is very large, the AI-SGD procedure has very large bias but small empirical standard error. Even if the estimated standard error aligns with the empirical one, it may not provide valid inference. In addition to the SGD types of recursive algorithms, several cumulative updating methods have been proposed to specifically perform sequential updating of regression coefficient estimators, including the online least squares estimator (OLSE) for the linear model by *Stengel* (1994), the cumulative estimating equation (CEE) estimator and the cumulatively updated estimating equation (CUEE) estimator by *Schifano et al.* (2016) for estimating equations. Even though CUEE is shown to have less estimation bias than CEE with finite sample sizes, its estimation consistency has been established upon a strong regularity condition: the total number

of streaming datasets, *say*, B, needs to satisfy the order of $B = \mathcal{O}(n_j^k)$, with $k < 1/3$ for $j = 1, 2, \ldots, B$, where n_j is the sample size of the j-th streaming dataset (*Lin and Xi*, 2011; *Schifano et al.*, 2016). This strong condition is also required by CEE for its estimation consistency. This implies a very strong restriction for these two methods; for example, estimation consistency may not be guaranteed in the situation where streaming datasets arrive perpetually with $B \to \infty$. Our proposed renewable estimation method overcomes this unnatural restriction. Section 2.2 presents a more detailed review of these existing methods.

Streaming data analytics may be implemented in the so-called Lambda architecture (*Marz and Warren*, 2015). It is a real-time Big Data system of computing and storage with a synchronized processing of batch and stream data flows. The Lambda architecture consists of three layers: the speed layer, the batch layer, and the serving layer. Figure 2.1 shows a schematic outline as to how the speed and batch layers interact when a new data stream arrives. Transient and rough real-time views are captured at the speed layer using incremental algorithms, where previously stored views are updated with an incoming data stream to generate renewed views. In effect, SGD is one of the most popular incremental algorithms widely used in the Spark Streaming to process high-throughput streaming data via the Apache Spark System (*Bifet et al.*, 2015). The batch layer stores a constantly growing dataset and continuously recomputes the batch views when new data stream arrives. Despite latency, the batch layer refines results produced in the speed layer where estimation accuracy cannot be maintained consistently. Then the two view outputs are stored in the serving layer for queries. This architecture is flexible and applicable to a wide range of streaming data analysis in which the batch layer stores all sequentially accumulated raw data and produces reliable results via re-computations. Unfortunately, this powerful architecture has completely ignored the need of real-time statistical inference; for example, there are no gears in the system designed to sequentially compute and store Fisher

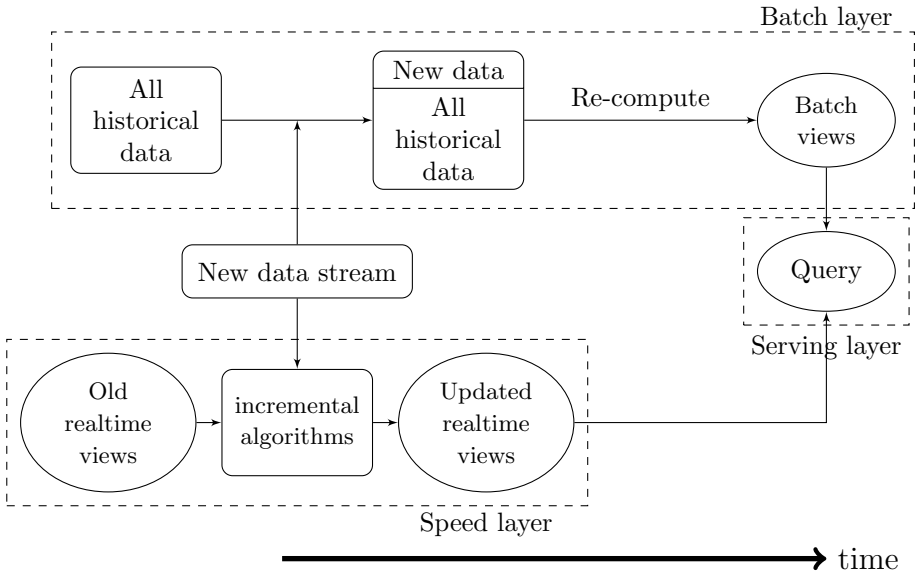

Figure 2.1: Diagram of the Lambda architecture.

information or as such, a critical piece required for statistical inference. To overcome, in this paper we propose to expand the speed layer by adding a new "inference layer", and this new sub-architecture is named as "Rho architecture" (named after the first letter of Greek word "stream", $\rho\varepsilon\nu\mu\alpha$), shown in Figure 2.2, so the resulting expanded architecture allows to conduct statistical inference in the setting of high-throughput streaming datasets.

In the proposed Rho architecture, we aim to address three basic questions for the new method: (i) what types of summary statistics to be stored in the inference layer; (ii) how to update those summary statistics required for estimation and inference without use of previous raw data; and (iii) how to optimize the renewable estimation method so as to achieve the asymptotically equivalent efficiency to that of MLE based on the entire dataset. In the setting of GLMs (*McCullagh and Nelder*, 1983) with data streams, our goal is to fit a regression model $\mathbb{E}(y_i \mid \boldsymbol{x}_i) = g(\boldsymbol{x}_i^T\boldsymbol{\beta})$ for subjects $i = 1, 2, \ldots, N_b$, where $g(\cdot)$ is a known link function and N_b is the sample size of aggregated streaming dataset up to data batch b, $N_b = \sum_{j=1}^{b} n_j$. Consider a time point $b \geq 2$ with a total of N_b samples arriving in a series of b data batches, denoted by $D_1 =$

$\{\boldsymbol{y}_1, \boldsymbol{X}_1\}, \ldots, D_b = \{\boldsymbol{y}_b, \boldsymbol{X}_b\}, \ldots$, where $\boldsymbol{y}$ and $\boldsymbol{X}$ are the generic notations of response variables and associated covariates. Under a fixed design, suppose each observation is drawn from $(y_i; \boldsymbol{x}_i) \sim f(y; \boldsymbol{x}, \boldsymbol{\beta}_0, \phi_0), i = 1, \ldots, N_b$ independently, where $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ is the true value of the parameter of interest and ϕ_0 is the true value of a nuisance parameter. Let $D_b^\star = \{D_1, \ldots, D_b\}$ denote the accumulated data up to data batch b. For convenience, slightly abusing the notation, we use D_b (a single data batch b) or $D_b^\star$ (an aggregation of b data batches) as respective sets of indices for subjects involved. For a GLM, we may write out the associated log-likelihood function in the form of exponential dispersion model (ED) (*Jørgensen*, 1997):

$$\ell_{N_b}(\boldsymbol{\beta}, \phi; D_b^\star) = \sum_{i \in D_b^\star} \log f(y_i; \boldsymbol{x}_i, \boldsymbol{\beta}, \phi) = \sum_{i \in D_b^\star} \log a(y_i; \phi) - \frac{1}{2\phi} \sum_{i \in D_b^\star} d(y_i; \mu_i), \quad (2.2)$$

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

unavailable for the GLMs.

In this chapter, we consider developing a new framework of renewable estimation and incremental inference, in which the MLE can be renewed with current data and summary statistics of historical data, but with no use of any historical subject-level data. To understand the nature of summary statistics, we introduce a notion of approximate sufficiency in the theory of renewable estimation. Different from the SGD, our new method enables to continuously update information matrices along streaming datasets, so to update statistical inference whenever a new data batch arrives. In Section 2.4.2, we present renewable Wald test statistic for hypothesis testing to conduct incremental inference within the proposed renewable methodology.

Our new methodology contributions include: (i) we propose a Rho architecture as a critical expansion of the Spark's Lambda architecture for streaming data analysis with the purpose of statistical inference, including both methods of renewable estimation and incremental inference; (ii) the proposed renewable estimator is shown to be asymptotically equivalent to the oracle MLE derived from the full cumulative data without strong condition $B = \mathcal{O}(n_j^k)$, $k < 1/3$; (iii) the ℓ_2-norm difference between our renewable estimator and the oracle MLE vanishes as the total sample size increases; and (iv) being computationally advantageous, our method does not require a re-access to any old subject-level data after the completion of current updating step. Thus, our renewable estimation method is computationally efficient to address the challenge of data storage and data processing, which is particularly useful in the case where the number of data batches increases fast and/or perpetually. In addition, our method provides a real-time interim inference based on the Wald test statistic without constraints on the relative scale of batch size and total number of batches, $B = \mathcal{O}(n_j^k)$, $k < 1/3$, which, as pointed above, presents significant restrictions in existing methods.

This paper is organized as follows. Section 2.2 gives a brief overview on several

existing methods: stochastic gradient descent (SGD), online LSE as well as CEE and CUEE. Section 2.3 presents our renewable estimation framework and incremental updating algorithm that is used to obtain renewable estimate. Section 2.4 includes the derivation of some key large sample properties, discussion on hypothesis testing methods, and approximate sufficient statistic. Section 2.5 presents numerical implementation and some examples of commonly used generalized linear models. Section 2.6 presents simulation results with comparisons of our proposed renewable methodology to the oracle MLE, SGD, and incremental estimators including online LSE, CEE and CUEE. Section 2.7 illustrates the proposed method by a real data analysis application. Some concluding remarks are provided in Section 2.8. All technical details are included in the appendix, including a comparison of computational complexity among second-order online methods, a table of notations for the sake of readability, the proofs of estimation consistency, asymptotic efficiency, the asymptotic equivalence between the renewable estimator and the oracle MLE, as well as properties of approximate sufficient statistics in the GLMs.

2.2 Existing Methods

We begin with some necessary notations that are also listed in Table A.2 in the appendix for perusal. At an intermediary time point b, $\hat{\boldsymbol{\beta}}_b^\star$ denotes the oracle MLE estimator based on the entire cumulative dataset $D_b^\star$, and $\tilde{\boldsymbol{\beta}}_b$ denotes a renewable estimator with the same dataset $D_b^\star$. Here $\hat{\boldsymbol{\beta}}_b^\star$ serves as the gold standard in all subsequent comparisons. Throughout this paper, $\hat{\boldsymbol{\beta}}$ denotes MLE, and "$\star$" in the superscript, e.g. $\hat{\boldsymbol{\beta}}_b^\star$ indicates a quantity derived from a cumulative dataset $D_b^\star$; otherwise, it is obtained from a single data batch, e.g.. $\hat{\boldsymbol{\beta}}_b$ from D_b. Likewise, $\tilde{\boldsymbol{\beta}}$ denotes an estimator obtained by an online updating procedure (e.g., online LSE, CEE, CUEE and our renewable estimator); for convenience, "$\sim$" over a symbol, *say*, $\tilde{a}$, denotes a quantity obtained cumulatively by an incremental algorithm using summary statistics of the

historical data. For example, $\tilde{\boldsymbol{U}}_2$ denotes the aggregated unit score function from the cumulative dataset $D_2^\star = D_1 \cup D_2$, while $\boldsymbol{U}_2$ denotes the one from a single data batch D_2 only.

2.2.1 Stochastic Gradient Descent Algorithm

Averaged Implicit SGD.

Toulis et al. (2014) proposed an averaged implicit stochastic gradient descent (AI-SGD) algorithm that took a cumulative average along iterations involving two steps: the first solves an implicit root $\boldsymbol{\beta}_i^{\text{im}}$, followed by a cumulative average, $\boldsymbol{\beta}_i^{\text{aim}}$.

$$\begin{aligned} \boldsymbol{\beta}_i^{\text{im}} &= \boldsymbol{\beta}_{i-1}^{\text{im}} + \gamma_i \boldsymbol{U}(y_i; \boldsymbol{x}_i, \boldsymbol{\beta}_i^{\text{im}}), \\ \boldsymbol{\beta}_i^{\text{aim}} &= \frac{1}{i} \sum_{k=1}^{i} \boldsymbol{\beta}_k^{\text{im}}, \quad i = 1, \ldots, N_b. \end{aligned} \tag{2.3}$$

It is shown that the AI-SGD in (2.3) not only works with flexible learning rate γ_i with desirable numerical stability, but also achieves the optimal Cramér-Rao bound under a strong convexity assumption. Throughout the paper, we compare our renewable estimation method to the AI-SGD method with the one-dimensional learning rate (*Xu*, 2011) and hyperparameters $\alpha = 1$, $\gamma_0 = 1$ and $c = 2/3$ are left to default values in the R package `sgd`.

Randomly Weighted AI-SGD. To approximate the sampling distribution of AI-SGD, *Fang* (2019) proposed a randomly weighted implicit stochastic gradient descent algorithm that adds a random weight to the gradient in implicit SGD procedure, followed by a cumulative average.

$$\begin{aligned} \boldsymbol{\beta}_i^{(\text{s})\text{im}} &= \boldsymbol{\beta}_{i-1}^{\text{im}} + \gamma_i W_i^s \boldsymbol{U}(y_i; \boldsymbol{x}_i, \boldsymbol{\beta}_i^{(\text{s})\text{im}}), \\ \boldsymbol{\beta}_i^{(\text{s})\text{aim}} &= \frac{1}{i} \sum_{k=1}^{i} \boldsymbol{\beta}_k^{(\text{s})\text{im}}, \quad i = 1, \ldots, N_b, \end{aligned} \tag{2.4}$$

where $W_i^{(s)} \overset{i.i.d.}{\sim}$ Exponential(1). At each i, we can obtain S copies of $\boldsymbol{\beta}_i^{(\mathrm{s})\mathrm{aim}}$, $s = 1, \ldots, S$, and then estimating the standard errors of $\boldsymbol{\beta}_i^{\mathrm{aim}}$ by the empirical standard errors of $\{\boldsymbol{\beta}_i^{(\mathrm{s})\mathrm{aim}}, s = 1, \ldots, S\}$. Following *Fang* (2019), we set $S = 200$ in our simulations.

2.2.2 Sequential Updating Methods

Online Least Squares Estimation. Consider a linear model $y_i = \boldsymbol{x}_i^T \boldsymbol{\beta}_0 + \epsilon_i$, with i.i.d. errors ϵ_i's, $i = 1, \ldots, N_b$, for cumulative dataset $D_b^\star$ to time point b. For the current single data batch D_b, the LSE (LSE) and its sum of squared errors (SSE) are denoted by $\hat{\boldsymbol{\beta}}_b = (\boldsymbol{X}_b^T \boldsymbol{X}_b)^{-1} \boldsymbol{X}_b^T \boldsymbol{y}_b$ and $SSE_b = SSE(\hat{\boldsymbol{\beta}}_b; D_b)$, respectively. Let $\tilde{\boldsymbol{\beta}}_b^{\mathrm{olse}}$ and $CMSE_b$ denote the online LSE (OLSE) and the cumulative mean squared error (CMSE) based on $D_b^\star$. With initial $\tilde{\boldsymbol{\beta}}_1^{\mathrm{olse}} = \hat{\boldsymbol{\beta}}_1$, the OLSE takes the following form of decomposition:

$$\tilde{\boldsymbol{\beta}}_b^{\mathrm{olse}} = \left(\sum_{j=1}^{b-1} \boldsymbol{X}_j^T \boldsymbol{X}_j + \boldsymbol{X}_b^T \boldsymbol{X}_b \right)^{-1} \left(\sum_{j=1}^{b-1} \boldsymbol{X}_j^T \boldsymbol{X}_j \tilde{\boldsymbol{\beta}}_{b-1}^{\mathrm{olse}} + \boldsymbol{X}_b^T \boldsymbol{X}_b \hat{\boldsymbol{\beta}}_b \right), \ b = 2, 3, \ldots. \tag{2.5}$$

The cumulative SSE (CSSE) takes a recursive procedure:

$$\begin{aligned} CSSE_b :=& SSE(\tilde{\boldsymbol{\beta}}_b^{\mathrm{olse}}; D_b^\star) \\ =& CSSE_{b-1} + SSE_b + \tilde{\boldsymbol{\beta}}_{b-1}^{\mathrm{olse}^T} \left(\sum_{j=1}^{b-1} \boldsymbol{X}_j^T \boldsymbol{X}_j \right) \tilde{\boldsymbol{\beta}}_{b-1}^{\mathrm{olse}} + \hat{\boldsymbol{\beta}}_b^T \boldsymbol{X}_b^T \boldsymbol{X}_b \hat{\boldsymbol{\beta}}_b \\ & - \tilde{\boldsymbol{\beta}}_b^{\mathrm{olse}^T} \left(\sum_{j=1}^{b} \boldsymbol{X}_j^T \boldsymbol{X}_j \right) \tilde{\boldsymbol{\beta}}_b^{\mathrm{olse}}, \ b = 2, 3, \ldots. \end{aligned} \tag{2.6}$$

The initial $CSSE_1 := SSE(\hat{\boldsymbol{\beta}}_1; D_1)$. It follows that the CMSE with $D_b^\star$ is $CMSE_b := MSE(\tilde{\boldsymbol{\beta}}_b^{\mathrm{olse}}; D_b^\star) = CSSE_b/(N_b - p)$.

Online Estimation with Estimating Equations. Let $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ be a parameter value satisfying $\sum_{i \in D_b^\star} \mathbb{E}\{\boldsymbol{\psi}(y_i, \boldsymbol{x}_i; \boldsymbol{\beta}_0)\} = \mathbf{0}$, where $\boldsymbol{\psi}(\cdot)$ is an unbiased estimating

function. This includes the unit score $\boldsymbol{U}$ and the scaled score $\boldsymbol{U}(\cdot)/\phi$ as special cases. The estimator and its variance with D_b are denoted by $\hat{\boldsymbol{\beta}}_b$ and $\boldsymbol{V}_b$ where $\boldsymbol{V}_b$ is the sandwich covariance matrix. A cumulative estimating equation (CEE) estimator, $\tilde{\boldsymbol{\beta}}_b^{\text{cee}}$, proposed by *Schifano et al.* (2016) is a sequentially updated estimate by the means of the following meta-type estimation, together with the corresponding cumulative negative Hessian matrix $\tilde{\boldsymbol{A}}_b^{\text{cee}}$:

$$\begin{aligned}\tilde{\boldsymbol{\beta}}_b^{\text{cee}} &= \left(\tilde{\boldsymbol{A}}_{b-1}^{\text{cee}} + \boldsymbol{A}_b^{\text{cee}}\right)^{-1}\left(\tilde{\boldsymbol{A}}_{b-1}^{\text{cee}}\tilde{\boldsymbol{\beta}}_{b-1}^{\text{cee}} + \boldsymbol{A}_b^{\text{cee}}\hat{\boldsymbol{\beta}}_b\right),\\ \tilde{\boldsymbol{A}}_b^{\text{cee}} &= \sum_{j=1}^{b}\boldsymbol{A}_j^{\text{cee}},\ b = 1, 2, \ldots,\end{aligned} \tag{2.7}$$

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

$$\begin{aligned}\tilde{\boldsymbol{V}}_b^{\text{cee}} \coloneqq \widetilde{\text{Var}}(\tilde{\boldsymbol{\beta}}_b^{\text{cee}}) = \left(\tilde{\boldsymbol{A}}_{b-1}^{\text{cee}} + \boldsymbol{A}_b^{\text{cee}}\right)^{-1}&\left\{\tilde{\boldsymbol{A}}_{b-1}^{\text{cee}}\tilde{\boldsymbol{V}}_{b-1}^{\text{cee}}\left(\tilde{\boldsymbol{A}}_{b-1}^{\text{cee}}\right)^T + \boldsymbol{A}_b^{\text{cee}}\boldsymbol{V}_b\left(\boldsymbol{A}_b^{\text{cee}}\right)^T\right\}\\ &\times\left\{\left(\tilde{\boldsymbol{A}}_{b-1}^{\text{cee}} + \boldsymbol{A}_b^{\text{cee}}\right)^{-1}\right\}^T,\ b = 1, 2, \ldots.\end{aligned} \tag{2.8}$$

It is easy to show that the bias of $\tilde{\boldsymbol{\beta}}_b^{\text{cee}}$ in (2.7) is of order $\mathcal{O}\left(\sum_{j=1}^{b} n_j^{-1/2}\right)$, which is $bn^{-1/2}$ for the case of equal batch size $n_j = n$ for all j. This suggests that for a small n_j, the dominance of b in the order of bias produces a cumulative bias, and consequently the meta estimator $\tilde{\boldsymbol{\beta}}_b^{\text{cee}}$ in (2.7) becomes increasingly biased over data batches. To reduce bias, a cumulatively updated estimating equation (CUEE) estimator is proposed by *Schifano et al.* (2016). The CUEE estimator and the corresponding

cumulative negative Hessian $\tilde{\boldsymbol{A}}_b^{\text{cuee}}$:

$$\begin{aligned}\tilde{\boldsymbol{\beta}}_b^{\text{cuee}} &= \left(\tilde{\boldsymbol{A}}_{b-1}^{\text{cuee}} + \boldsymbol{A}_b^{\text{cuee}}\right)^{-1} \\ &\quad \times \left\{\sum_{j=1}^{b-1} \boldsymbol{A}_j^{\text{cuee}} \check{\boldsymbol{\beta}}_j + \boldsymbol{A}_b^{\text{cuee}} \check{\boldsymbol{\beta}}_b + \sum_{j=1}^{b-1} \sum_{i \in D_j} \boldsymbol{\psi}_i(\check{\boldsymbol{\beta}}_j) + \sum_{i \in D_b} \boldsymbol{\psi}_i(\check{\boldsymbol{\beta}}_b)\right\}, \\ \tilde{\boldsymbol{A}}_b^{\text{cuee}} &= \sum_{j=1}^{b} \boldsymbol{A}_j^{\text{cuee}}, \ b = 1, 2, \ldots,\end{aligned} \tag{2.9}$$

with initial $\tilde{\boldsymbol{A}}_0 = \boldsymbol{A}_0 = \boldsymbol{0}_{p\times p}$, $\boldsymbol{A}_b^{\text{cuee}} = -\sum_{i\in D_b} \nabla_{\boldsymbol{\beta}} \boldsymbol{\psi}(y_i, \boldsymbol{x}_i; \check{\boldsymbol{\beta}}_b)$ is the negative Hessian of D_b evaluated at $\check{\boldsymbol{\beta}}_b$ which is an intermediary estimator similar to the CEE estimator. Similarly, initiated by $\tilde{\boldsymbol{V}}_0^{\text{cuee}} = \boldsymbol{0}_{p\times p}$, the recursively updated variance of $\tilde{\boldsymbol{\beta}}_b^{\text{cuee}}$ is

$$\begin{aligned}\tilde{\boldsymbol{V}}_b^{\text{cuee}} \coloneqq \widetilde{\text{Var}}(\tilde{\boldsymbol{\beta}}_b^{\text{cuee}}) &= \left(\tilde{\boldsymbol{A}}_{b-1}^{\text{cuee}} + \boldsymbol{A}_b^{\text{cuee}}\right)^{-1} \left\{\tilde{\boldsymbol{A}}_{b-1}^{\text{cuee}} \tilde{\boldsymbol{V}}_{b-1}^{\text{cuee}} \left(\tilde{\boldsymbol{A}}_{b-1}^{\text{cuee}}\right)^T + \boldsymbol{A}_b^{\text{cuee}} \boldsymbol{V}_b \left(\boldsymbol{A}_b^{\text{cuee}}\right)^T\right\} \\ &\quad \times \left\{\left(\tilde{\boldsymbol{A}}_{b-1}^{\text{cuee}} + \boldsymbol{A}_b^{\text{cuee}}\right)^{-1}\right\}^T, \ b = 1, 2, \ldots.\end{aligned} \tag{2.10}$$

As shown by *Schifano et al.* (2016), the CUEE estimator is less biased than the CEE estimator under finite sample sizes. Nevertheless, its estimation consistency is established under the same strong regularity condition as that required by the CEE estimator; that is, the number of data batches b is of order $\mathcal{O}(n_j^k)$, for $k < 1/3$ and each $j = 1, \ldots, b$. As pointed out above, this condition apparently is not valid for high throughput streaming data, where n_j is typically small, but b grows at a high rate. Consequently, in this case, the valid statistical inference is not yet available.

2.3 Renewable Estimation

Let $\tilde{\boldsymbol{\beta}}_b$ be a renewable estimator, with $\tilde{\boldsymbol{\beta}}_1$ being initialized by the MLE, namely, $\hat{\boldsymbol{\beta}}_1$, from the first data batch D_1. For $b = 2, 3, \ldots$, a previous estimator $\tilde{\boldsymbol{\beta}}_{b-1}$ is sequentially updated to $\tilde{\boldsymbol{\beta}}_b$ when data batch D_b arrives; after the updating, data batch

D_b is no longer accessible except estimate $\tilde{\boldsymbol{\beta}}_b$ and summary statistics $\boldsymbol{J}_b(D_b;\tilde{\boldsymbol{\beta}}_b)$ and $\tilde{\phi}_b$, which are carried forward in future calculations.

2.3.1 Method

We begin with a simple scenario of two data batches, where the second data batch D_2 arrives after the first data batch D_1. Similar to the operation given in (2.1), we are interested in updating the initial MLE $\hat{\boldsymbol{\beta}}_1$ (or $\hat{\boldsymbol{\beta}}_1^\star$) to a renewed MLE $\hat{\boldsymbol{\beta}}_2^\star$ without using any subject-level data but only some summary statistics from D_1.

The initial MLE $\hat{\boldsymbol{\beta}}_1$ in a GLM satisfies the unit score equation, $\boldsymbol{U}_1(D_1;\hat{\boldsymbol{\beta}}_1)=\mathbf{0}$. When D_2 arrives, we hope to obtain an updated MLE, $\hat{\boldsymbol{\beta}}_2^\star$, that satisfies the following aggregated unit score equation:

$$\boldsymbol{U}_1(D_1;\hat{\boldsymbol{\beta}}_2^\star)+\boldsymbol{U}_2(D_2;\hat{\boldsymbol{\beta}}_2^\star)=\mathbf{0}. \tag{2.11}$$

Let $\boldsymbol{U}_b(D_b;\boldsymbol{\beta})=\sum_{i\in D_b}\boldsymbol{U}(y_i;\boldsymbol{x}_i,\boldsymbol{\beta})$ be the unit score function of current data batch D_b, and the negative Hessian of the unit deviance $d(\cdot;\cdot)$ is denoted by $\boldsymbol{J}_b(D_b;\boldsymbol{\beta}):=-\sum_{i\in D_b}\partial^2\ d(y_i;\mu_i)/\partial\boldsymbol{\beta}^2,\ b=1,2,\ldots.$ Note that the dispersion parameter ϕ is not involved in estimation as the root of equation (2.11), but in the calculation of Fisher information. Additionally, solving (2.11) for $\hat{\boldsymbol{\beta}}_2^\star$ actually involves the use of subject-level data in both data batches D_1 and D_2. To derive a renewable version of estimation, we take the first-order Taylor expansion of the first term in (2.11) around MLE $\hat{\boldsymbol{\beta}}_1$,

$$\boldsymbol{U}_1(D_1;\hat{\boldsymbol{\beta}}_1)+\boldsymbol{J}_1(D_1;\hat{\boldsymbol{\beta}}_1)(\hat{\boldsymbol{\beta}}_1-\hat{\boldsymbol{\beta}}_2^\star)+\boldsymbol{U}_2(D_2;\hat{\boldsymbol{\beta}}_2^\star)+\mathcal{O}_p(\|\hat{\boldsymbol{\beta}}_2^\star-\hat{\boldsymbol{\beta}}_1\|^2)=\mathbf{0}. \tag{2.12}$$

Since D_1 and D_2 are independently sampled from the same underlying population with a common true parameter $\boldsymbol{\beta}_0$, when $\min\{n_1,n_2\}$ is large enough, under some mild regularity conditions, both $\hat{\boldsymbol{\beta}}_1$ and $\hat{\boldsymbol{\beta}}_2^\star$ are consistent estimators of $\boldsymbol{\beta}_0$ (e.g. *Fahrmeir*

and Kaufmann (1985)). This implies that the error term $\mathcal{O}_p(\|\hat{\boldsymbol{\beta}}_2^\star - \hat{\boldsymbol{\beta}}_1\|^2)$ in (2.12) may be asymptotically ignored. Removing such term, we propose a new estimator $\tilde{\boldsymbol{\beta}}_2$ as a solution to the equation of the form:

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

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

$$\boldsymbol{J}_1(D_1;\hat{\boldsymbol{\beta}}_1)(\hat{\boldsymbol{\beta}}_1 - \tilde{\boldsymbol{\beta}}_2) + \boldsymbol{U}_2(D_2;\tilde{\boldsymbol{\beta}}_2) = \mathbf{0}. \tag{2.13}$$

Note that $\tilde{\boldsymbol{\beta}}_2$ approximates the oracle MLE $\hat{\boldsymbol{\beta}}_2^\star$ up to the second order asymptotic errors. Through (2.13), the initial $\hat{\boldsymbol{\beta}}_1$ is renewed by $\tilde{\boldsymbol{\beta}}_2$. Because of this, $\tilde{\boldsymbol{\beta}}_2$ is *a renewable estimator* of $\boldsymbol{\beta}_0$, and equation (2.13) is termed as *an incremental estimating equation*. Numerically, it is rather straightforward to find $\tilde{\boldsymbol{\beta}}_2$ by, for example, the Newton-Raphson algorithm or Fisher scoring algorithm with $\phi = 1$. Note that these two algorithms are equivalent in this paper that concerns the GLM with a canonical link. That is, at the $(r+1)$-th iteration,

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

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

algorithm:

$$\begin{aligned}\tilde{\boldsymbol{\beta}}_2^{(r+1)} &= \tilde{\boldsymbol{\beta}}_2^{(r)} + \left\{\sum_{j=1}^{2} \boldsymbol{J}_j(D_j;\hat{\boldsymbol{\beta}}_1)\right\}^{-1} \left\{\boldsymbol{J}_1(D_1;\hat{\boldsymbol{\beta}}_1)\left(\hat{\boldsymbol{\beta}}_1 - \tilde{\boldsymbol{\beta}}_2^{(r)}\right) + \boldsymbol{U}_2\left(D_2;\tilde{\boldsymbol{\beta}}_2^{(r)}\right)\right\} \\ &= \tilde{\boldsymbol{\beta}}_2^{(r)} + \left\{\boldsymbol{J}_1(\hat{\boldsymbol{\beta}}_1) + \boldsymbol{J}_2(\hat{\boldsymbol{\beta}}_1)\right\}^{-1} \tilde{\boldsymbol{U}}_2^{(r)},\end{aligned} \tag{2.14}$$

where $\tilde{\boldsymbol{U}}_2^{(r)} = \boldsymbol{J}_1(D_1;\hat{\boldsymbol{\beta}}_1)\left(\hat{\boldsymbol{\beta}}_1 - \tilde{\boldsymbol{\beta}}_2^{(r)}\right) + \boldsymbol{U}_2\left(D_2;\tilde{\boldsymbol{\beta}}_2^{(r)}\right)$. In equation (2.14), $\tilde{\boldsymbol{\beta}}_2$ is iteratively solved by using the adjusted unit score function $\tilde{\boldsymbol{U}}_2$ and the aggregated negative Hessian $\left\{\boldsymbol{J}_1(\hat{\boldsymbol{\beta}}_1) + \boldsymbol{J}_2(\hat{\boldsymbol{\beta}}_1)\right\}$ evaluated at the previous estimate $\hat{\boldsymbol{\beta}}_1$. In this paper, we name this algorithm as *incremental updating algorithm*. It is interesting to note that equation (2.14) may be regarded as a kind of gradient descent algorithm, so its solution will converge to the root of equation (2.13). Similar ideas have been used in the literature to speed up the calculation of Hessian matrix; see for example, *Song et al.* (2005). The difference between the proposed renewable estimator $\tilde{\boldsymbol{\beta}}_2$ and the oracle $\hat{\boldsymbol{\beta}}_2^{\star}$ stems from an approximation to the unit score function $\boldsymbol{U}_1(D_1;\hat{\boldsymbol{\beta}}_2^{\star})$, which, as shown in Theorem II.6, vanishes at the rate of $1/N_2$, with $N_2 = |D_2^{\star}| = n_1 + n_2$. In practice, because accumulated sample size $N_b = \sum_{j=1}^{b} n_j$ increases to infinity very fast, these two estimators, $\tilde{\boldsymbol{\beta}}_b$ and $\hat{\boldsymbol{\beta}}_b^{\star}$, are numerically very close, and eventually become the same. To run the incremental updating algorithm (2.14), we extend the Spark Lambda architecture by designing the Rho architecture that stores three key components $\left\{\hat{\boldsymbol{\beta}}_1, \boldsymbol{J}_1(D_1;\hat{\boldsymbol{\beta}}_1), \hat{\phi}_1\right\}$. Here, the initial estimate of the dispersion parameter is given by $\hat{\phi}_1 = \frac{1}{n_1 - p}\sum_{i\in D_1} \frac{(y_i - \hat{\mu}_i)^2}{v(\hat{\mu}_i)}$ based on the Pearson residuals, where $\hat{\mu}_i = g(\boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}_1)$.

Generalizing the above procedure to streaming datasets, we now define a renewable estimation of $\boldsymbol{\beta}_0$ as follows. Let $\hat{\boldsymbol{\beta}}_b^{\star}$ be the oracle MLE of $\boldsymbol{\beta}_0$ with the accumulated data $D_b^{\star} = \cup_{j=1}^{b} D_j$ that satisfies the cumulative unit score equation: $\sum_{j=1}^{b} \boldsymbol{U}_j(D_j;\hat{\boldsymbol{\beta}}_b^{\star}) = \mathbf{0}$. We propose a renewable estimator $\tilde{\boldsymbol{\beta}}_b$ of $\boldsymbol{\beta}_0$ as a solution to the following incremental

estimating equation:

$$\sum_{j=1}^{b-1} \boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)(\tilde{\boldsymbol{\beta}}_{b-1}-\tilde{\boldsymbol{\beta}}_b)+\boldsymbol{U}_b(D_b;\tilde{\boldsymbol{\beta}}_b)=\mathbf{0}, \tag{2.15}$$

where $\hat{\boldsymbol{\beta}}_1=\tilde{\boldsymbol{\beta}}_1$ at the initial data batch D_1. Note that when $b=2$, equation (2.15) reduces to equation (2.13). Let $\tilde{\boldsymbol{J}}_b=\sum_{j=1}^{b}\boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)$ denote the aggregated negative Hessian matrix. Solving equation (2.15) may be easily done by the following incremental updating algorithm:

$$\tilde{\boldsymbol{\beta}}_b^{(r+1)}=\tilde{\boldsymbol{\beta}}_b^{(r)}+\left\{\tilde{\boldsymbol{J}}_{b-1}+\boldsymbol{J}_b(D_b;\tilde{\boldsymbol{\beta}}_{b-1})\right\}^{-1}\tilde{\boldsymbol{U}}_b^{(r)}, \tag{2.16}$$

where the adjusted unit score $\tilde{\boldsymbol{U}}_b^{(r)}=\tilde{\boldsymbol{J}}_{b-1}(\tilde{\boldsymbol{\beta}}_{b-1}-\tilde{\boldsymbol{\beta}}_b^{(r)})+\boldsymbol{U}_b(D_b;\tilde{\boldsymbol{\beta}}_b^{(r)})$. In equation (2.16), both the gradient and the adjusted unit score use the subject-level data of current batch D_b and summary statistics $\left\{\tilde{\boldsymbol{\beta}}_{b-1},\tilde{\boldsymbol{J}}_{b-1},\tilde{\phi}_{b-1}\right\}$ from historical data. In the end, a consistent estimator of the dispersion parameter ϕ is updated according to $\tilde{\phi}_b=\frac{N_{b-1}-p}{N_b-p}\tilde{\phi}_{b-1}+\frac{1}{N_b-p}\hat{\phi}_b$, where $\hat{\phi}_b$ takes the same form as $\hat{\phi}_1$ based on the Pearson residuals from the current data batch D_b and $\tilde{\boldsymbol{\beta}}_{b-1}$.

2.3.2 Rho Architecture

Apache Spark is known as a distributed computing system that allows the communication and coordination between batch and speed processing layers in the Lambda architecture. To implement our proposed algorithm that provides both real-time 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 Fisher information. Consequently, the proposed new Rho architecture consists of a speed layer and an inference layer responsible for inferential statistics updating, as shown in Figure 2.2. When a new data batch arrives, the speed layer updates the views (or estimates) in GLM with the utility of prior infer-

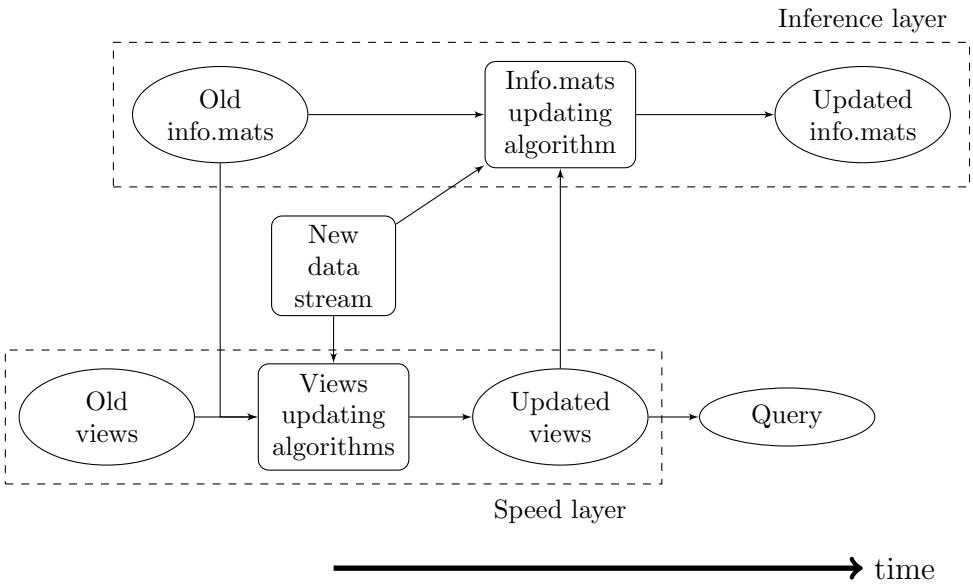

Figure 2.2: Diagram of the Rho architecture.

ential statistics from the inference layer. Then, the updated views are sent back to the inference layer, where, together with the current data, real-time updates of information matrices are generated. More specifically, in Figure 2.2, for GLM considered in this paper, the "views" are parameter estimates and "info.mats" correspond to inferential statistics, such as Fisher information matrices, required in the calculation of inferential quantities. The incremental algorithm in (2.16) can then be implemented in the proposed Rho architecture as shown in Figure 2.3.

2.3.3 An example: Linear Model

To see some specific operational details discussed above, here we present an example of the renewable estimation in the case of the Gaussian linear model. It is interesting to note that for the linear model, the proposed renewable estimation turns out to be identical to the online least squares estimation (OLSE) given in equations (2.5) and (2.6).

Example II.1. Consider data batch $D_b = \{\boldsymbol{y}_b, \boldsymbol{X}_b\}$ with outcome $\boldsymbol{y}_b = (y_{b1}, \ldots, y_{bn_b})^T$

and covariates $\boldsymbol{X}_b = (\boldsymbol{x}_{b1}, \ldots, \boldsymbol{x}_{bn_b})^T$. Suppose that $\boldsymbol{y}_b|\boldsymbol{X}_b$ are independently sampled from a Gaussian distribution with mean $\boldsymbol{\mu}_b = (\mu_{b1}, \ldots, \mu_{bn_b})^T$ and variance $\phi\boldsymbol{I}$ such that $\mu_{bi} = \mathbb{E}(y_{bi} \mid \boldsymbol{x}_{bi}) = \boldsymbol{x}_{bi}^T\boldsymbol{\beta}$ and $v(y_{bi} \mid \boldsymbol{x}_{bi}) = \phi$. Here the unit variance function $v(\mu_i) \equiv 1$. Then, the unit score function and the corresponding negative Hessian for data batch D_b are, respectively,

$$\boldsymbol{U}_b(\boldsymbol{\beta}) = \boldsymbol{X}_b^T(\boldsymbol{y}_b - \boldsymbol{X}_b\boldsymbol{\beta}), \quad \boldsymbol{J}_b(\boldsymbol{\beta}) = \boldsymbol{X}_b^T\boldsymbol{X}_b.$$

At the speed layer, we calculate the point estimation; a closed-form expression for the renewable estimator of $\boldsymbol{\beta}$ is obtained directly by solving the incremental estimating equation (2.15):

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

Here by convention, the initials are $\tilde{\boldsymbol{\beta}}_0 = \boldsymbol{0}_p$ and $\tilde{\boldsymbol{J}}_0 = \boldsymbol{0}_{p\times p}$. Moreover, an unbiased estimator of variance parameter ϕ based on $\tilde{\boldsymbol{\beta}}_b$ takes the following recursive formula:

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

This unbiased estimator of variance parameter ϕ can be renewed easily at the inference layer. In the linear model, the unbiased estimator $\tilde{\phi}_b$ is exact and stored in the inference layer as part of Fisher information calculation, which is given as follows:

$$\widetilde{\mathrm{Var}}(\tilde{\boldsymbol{\beta}}_b) = \tilde{\phi}_b(\tilde{\boldsymbol{J}}_{b-1} + \boldsymbol{J}_b)^{-1}.$$

Note that this estimated variance leads to exactly the same standard error as that given by the oracle MLE $\hat{\boldsymbol{\beta}}_b^\star$, which is obtained by fitting the linear model once

with the entire data $D_b^\star = \cup_{j=1}^b D_j$. So, the incremental estimation does not lose any estimation efficiency over the incremental updates, but is advantageous in data storage and computing speed.

2.4 Large Sample Properties, Inference and Sufficiency

In this section we first establish estimation consistency and asymptotic normality for the proposed renewable estimator, and then show its asymptotic equivalency to the oracle MLE. Also, we present the incremental inference based on the Wald statistic in the Rho architecture. To address the issue concerning which types of statistics suit for the renewable estimation framework, we discuss sufficient statistic and approximate sufficient statistic, as well as their connections to key summary statistics $\left\{\tilde{\boldsymbol{\beta}}_{b-1}, \tilde{\boldsymbol{J}}_{b-1}, \tilde{\phi}_{b-1}\right\}$ used in the Rho architecture.

2.4.1 Large Sample Properties

For an arbitrary time b, suppose $(y_i, \boldsymbol{x}_i)$ are $i.i.d.$ samples from an exponential dispersion model with density $f(y; \boldsymbol{x}, \boldsymbol{\beta}, \phi)$, $i = 1, \ldots, N_b$, with $\boldsymbol{\beta} \in \Theta \subset \mathbb{R}^p$ where the true parameter is $\boldsymbol{\beta}_0$, and ϕ is the dispersion parameter with true value ϕ_0. Under the canonical link, denote $\boldsymbol{\mathcal{I}}_{N_b}(\boldsymbol{\beta}_0) = \sum_{i=1}^{N_b} \mathbb{E}\left[\boldsymbol{U}_i \boldsymbol{U}_i^T\right] / \phi = \sum_{i=1}^{N_b} \boldsymbol{x}_i v(\mu_i) \boldsymbol{x}_i^T$ where $v(\cdot)$ is the known unit variance function. Let $\mathcal{B}_{N_b}(\delta)$ be a neighborhood of $\boldsymbol{\beta}_0$, namely

$$\mathcal{B}_{N_b}(\delta) = \{\boldsymbol{\beta} : \|\boldsymbol{\mathcal{I}}_{N_b}^{T/2}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\| \leq \delta\} \in \Theta, \quad \delta > 0, \tag{2.17}$$

where $\boldsymbol{\mathcal{I}}_{N_b}^{T/2}$ denotes the right Cholesky square root of $\boldsymbol{\mathcal{I}}_{N_b}(\boldsymbol{\beta}_0)$, according to $\boldsymbol{\mathcal{I}}_{N_b} = \boldsymbol{\mathcal{I}}_{N_b}^{1/2} \boldsymbol{\mathcal{I}}_{N_b}^{T/2}$, and $\|\cdot\|$ is the ℓ_2-norm.

We postulate the following regularity conditions:

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

(C2) $\boldsymbol{\mathcal{I}}_{N_b}(\boldsymbol{\beta})$ is positive-definite for all $\boldsymbol{\beta} \in \mathcal{B}_{N_b}(\delta)$.

(C3) The log-likelihood function $\ell(\boldsymbol{\beta}, \phi, \boldsymbol{x}; y)$ is twice continuously differentiable and $\boldsymbol{\mathcal{I}}_{N_b}(\boldsymbol{\beta})$ is Lipschitz continuous in Θ.

Remark II.2. Under condition (C1), the neighborhood $\mathcal{B}_{N_b}(\delta)$ shrinks to a singleton $\boldsymbol{\beta}_0$, as $N_b \to \infty$. Condition (C2) is necessary for both consistency and asymptotic normality. Both (C1) and (C2) are the standard regularity conditions assumed by *Fahrmeir and Kaufmann* (1985). Different from the traditional MLE, the consistency for the renewable estimator requires the continuity assumption (C3) to be held over the whole parameter space Θ, rather than over a neighborhood of $\boldsymbol{\beta}_0$. Since in the GLMs, the matrix $\boldsymbol{\mathcal{I}}_{N_b}(\boldsymbol{\beta})$ depends on $\boldsymbol{\beta}$ via the unit variance function $v(\mu_i)$ with $\mu_i = g(\boldsymbol{x}_i^T\boldsymbol{\beta})$, the Lipschitz continuity condition automatically holds on a compact parameter space, which is sufficient for most applications.

Theorem II.3. *Under conditions (C1)-(C3), the renewable estimator $\tilde{\boldsymbol{\beta}}_b$ given in* (2.15) *is consistent, namely $\tilde{\boldsymbol{\beta}}_b \xrightarrow{p} \boldsymbol{\beta}_0$, as $N_b = \sum_{j=1}^{b} n_j \to \infty$.*

The proof of Theorem II.3 is given in Section A.1 of the appendix.

Theorem II.4. *Under conditions (C1)-(C3), the renewable estimator $\tilde{\boldsymbol{\beta}}_b$ is asymptotically normally distributed, that is,*

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

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

The proof of Theorem II.4 is provided in Section A.2 of the appendix. It is interesting to notice that the asymptotic covariance matrix of the renewable estimator $\tilde{\boldsymbol{\beta}}_b$ given in Theorem II.4 is the same as that of the oracle MLE $\hat{\boldsymbol{\beta}}_b^{\star}$. This implies that the proposed renewable estimator is fully efficient; see also Remark II.5 below. With

no need of historical subject-level data in the computation, using the prior aggregated negative Hessian matrix stored in the Rho architecture, $\tilde{\boldsymbol{J}}_b = \sum_{j=1}^{b} \boldsymbol{J}_j(D_j; \tilde{\boldsymbol{\beta}}_j)$, we calculate the estimated asymptotic covariance matrix $\widetilde{\boldsymbol{\Sigma}_b}$ as follows:

$$\widetilde{\boldsymbol{\Sigma}_b} = \left\{ (N_b \tilde{\phi}_b)^{-1} \sum_{j=1}^{b} \boldsymbol{J}_j(D_j; \tilde{\boldsymbol{\beta}}_j) \right\}^{-1} = N_b \tilde{\phi}_b \tilde{\boldsymbol{J}}_b^{-1}.$$

It follows that the estimated variance matrix for $\tilde{\boldsymbol{\beta}}_b$ is given by

$$\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b) \coloneqq \widetilde{\mathrm{Var}}(\tilde{\boldsymbol{\beta}}_b) = \frac{1}{N_b} \widetilde{\boldsymbol{\Sigma}_b} = \tilde{\phi}_b \tilde{\boldsymbol{J}}_b^{-1}. \tag{2.18}$$

Remark II.5. Because both SGD and AI-SGD may be regarded as special cases of the proposed renewable estimator, with $n_j = 1$ for $j = 1, \ldots, b$, the result of Sakrison's asymptotic efficiency established by *Sakrison* (1965) remains true theoretically for AI-SGD (*Toulis and Airoldi*, 2015). Theorems II.4 presents an extension of the statistical efficiency result for the GLMs with streaming datasets.

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

Theorem II.6. *Under conditions (C1)-(C3), the ℓ_2-norm difference between the oracle MLE $\hat{\boldsymbol{\beta}}_b^{\star}$ and the proposed renewable estimator $\tilde{\boldsymbol{\beta}}_b$ vanishes at the rate of N_b^{-1}, namely*

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

Theorem II.6 implies that the renewable estimator achieves the optimal efficiency and its distance to the MLE $\hat{\boldsymbol{\beta}}_b^{\star}$ vanishes in the order of $\mathcal{O}_p(1/N_b)$. The proof of Theorems II.6 is included in Section A.3 of the appendix.

2.4.2 Incremental Inference

The Wald test based on the asymptotic distribution of the renewable estimator in Theorem II.4 is a straightforward approach to testing hypotheses of individual coefficients or of nested parameter sets. For $k < p$ and a pre-fixed null subvector $\boldsymbol{\beta}_1^{\text{null}}$, define the following null hypothesis parameter space Θ_{H_0}:

$$\Theta_{H_0} = \{(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2) = (\boldsymbol{\beta}_1^{\text{null}}, \beta_{k+1}, \ldots, \beta_p)\}, \tag{2.19}$$

where Θ_{H_0} is a $(p-k)$-dimensional subspace of Θ. The subvector $\tilde{\boldsymbol{\beta}}_{1b}$ of $\tilde{\boldsymbol{\beta}}_b$ corresponding to its first k parameters follows an asymptotically k-dimensional marginal normal distribution, according to Theorem II.4. Specifically, a suitable block-partition of the estimate $\tilde{\boldsymbol{\beta}}_b$ and its asymptotic variance matrix are given by, respectively,

$$\tilde{\boldsymbol{\beta}}_b = (\tilde{\boldsymbol{\beta}}_{1b}^T, \tilde{\boldsymbol{\beta}}_{2b}^T)^T, \text{ and } \boldsymbol{\Sigma}_0 = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}.$$

Under the null hypothesis $H_0 : \boldsymbol{\beta}_1 = \boldsymbol{\beta}_1^{\text{null}}$, $\sqrt{N_b}\left(\tilde{\boldsymbol{\beta}}_{1b} - \boldsymbol{\beta}_1^{\text{null}}\right) \xrightarrow{d} \mathcal{N}_k(\mathbf{0}, \boldsymbol{\Sigma}_{11})$, as $N_b \to \infty$, which gives rise to the following asymptotic chi-square distribution with k degrees of freedom. That is, under the null H_0,

$$\begin{aligned} \tilde{W}_b &= (\tilde{\boldsymbol{\beta}}_{1b} - \boldsymbol{\beta}_1^{\text{null}})^T \left\{ \tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b)_{11} \right\}^{-1} (\tilde{\boldsymbol{\beta}}_{1b} - \boldsymbol{\beta}_1^{\text{null}}) \\ &= (\tilde{\boldsymbol{\beta}}_{1b} - \boldsymbol{\beta}_1^{\text{null}})^T \left\{ \left(\tilde{\phi}_b \tilde{\boldsymbol{J}}_b^{-1} \right)_{11} \right\}^{-1} (\tilde{\boldsymbol{\beta}}_{1b} - \boldsymbol{\beta}_1^{\text{null}}) \overset{asy}{\sim} \chi_k^2, \end{aligned} \tag{2.20}$$

where $\left(\tilde{\phi}_b \tilde{\boldsymbol{J}}_b^{-1} \right)_{11}$ is the $(1,1)$-block of matrix $\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b)$ in (2.18). Consequently, a $100(1-\alpha)\%$ confidence ellipsoid for subvector $\boldsymbol{\beta}_1$ is given by

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

It is worth pointing out that Rao's Score test and Wilks's likelihood-ratio test are not discussed here because both methods require the renewable estimates of $\boldsymbol{\beta}$ under H_0. Unlike the Wald test statistic which is just a direct byproduct of both estimate $\tilde{\boldsymbol{\beta}}_b$ and estimated asymptotic covariance matrix $\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b)$, 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.

2.5 Implementation

2.5.1 Rho architecture and pseudo code

The proposed renewable estimation and incremental inference may be implemented according to the Rho architecture in Figure 2.2. The work flow chart in Figure 2.3 facilitates the organization of the pseudo code for related numerical calculations.

Algorithm 2.4 lists the pseudo code for the implementation of the algorithm in Figure 2.3. Some explanations for the pseudo code are given below.

1. Line 1: the GLM considered in this paper belongs to the family of exponential dispersion (ED) models (*Jørgensen*, 1997) and all streaming datasets are supposed to be governed by the same model with a common true parameter $\boldsymbol{\beta}_0$. The ED models automatically satisfy some of the regularity conditions given in Section II.3, such as condition (C3).

2. Line 2: the outputs are renewable estimates of the regression coefficients and the corresponding estimated asymptotic variances at each time point b, and the latter is needed for statistical inference.

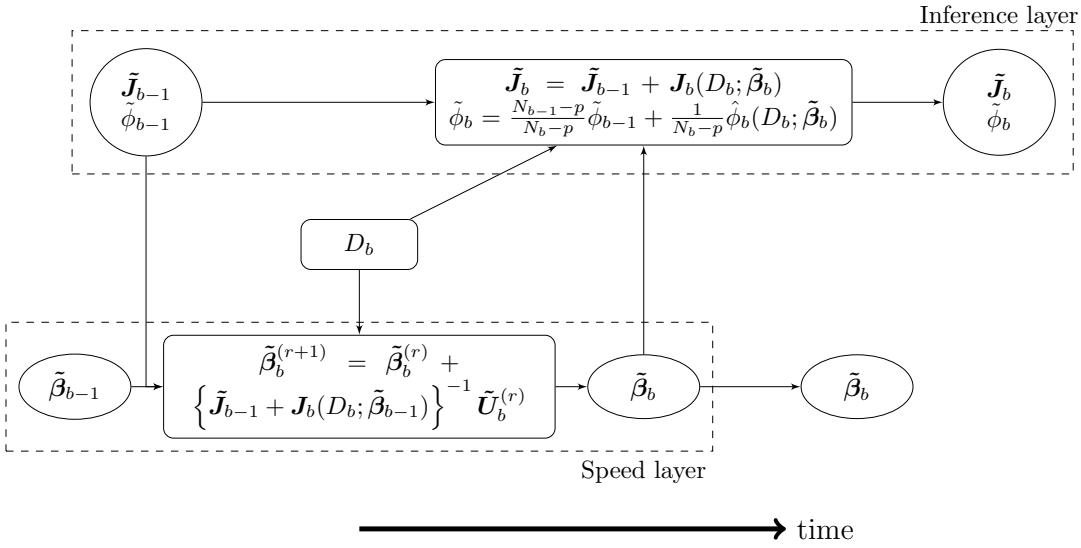

Figure 2.3: Implementation of GLM in the Rho architecture.

3. Line 3: set certain initial values for regression coefficients, *e.g.*, $\tilde{\boldsymbol{\beta}}_{\text{init}} = \mathbf{0}$.

4. Line 4: run through the online updating procedure along dataset streams.

5. Line 6: at the inference layer, utilize the prior estimate $\tilde{\boldsymbol{\beta}}_{b-1}$ and current data batch D_b to calculate the negative Hessian $\boldsymbol{J}_b(D_b; \tilde{\boldsymbol{\beta}}_{b-1})$ that is temporary and only used for the operation of incremental updating iterations. There is a communication between the speed layer and the inference layer to carry out the updating algorithm at the speed layer.

6. Line 7-8: run the incremental updating algorithm to update from $\tilde{\boldsymbol{\beta}}_{b-1}$ to $\tilde{\boldsymbol{\beta}}_b$. Note that with a given b this gradient matrix is fixed through all iterations, in which the cached factorizations are repetitively used in iterative steps;

7. Line 9: at the inference layer, update both negative Hessian matrix and dispersion parameter estimate with the current data batch D_b and newly updated $\tilde{\boldsymbol{\beta}}_b$ from the speed layer. These quantities are needed to perform statistical inference.

Algorithm 1: Incremental learning with the generalized linear model for the implementation of the Rho architecture.

Inputs: Model $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta}_0, \phi_0)$, sequentially arrived datasets $D_1, \ldots, D_b, \ldots$;

Outputs: $\tilde{\boldsymbol{\beta}}_b$ and $\text{diag}\{\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b)\}$, for $b = 1, 2, \ldots$;

Initialize: Certain initial values $\tilde{\boldsymbol{\beta}}_{\text{init}}$, $\tilde{\phi}_0 = 0$ and $\tilde{\boldsymbol{J}}_0 = \boldsymbol{0}_{p\times p}$;

for $b = 1, 2, \ldots$ **do**

  Read in dataset D_b;

  At the inference layer, perform Cholesky decomposition to $\left\{\tilde{\boldsymbol{J}}_{b-1} + \boldsymbol{J}_b(D_b; \tilde{\boldsymbol{\beta}}_{b-1})\right\}$ and cache the resulting factorizations;

  At the speed layer, with $\tilde{\boldsymbol{\beta}}_b^{(1)} = \tilde{\boldsymbol{\beta}}_{b-1}$, use the factorizations to run the following iterations until convergence

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

  At the inference layer, update both $\tilde{\boldsymbol{J}}_b = \tilde{\boldsymbol{J}}_{b-1} + \boldsymbol{J}_b(D_b; \tilde{\boldsymbol{\beta}}_b)$ and $\tilde{\phi}_b = \frac{N_{b-1}-p}{N_b-p}\tilde{\phi}_{b-1} + \frac{1}{N_b-p}\hat{\phi}_b$, and then calculate $\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b) = \tilde{\phi}_b \tilde{\boldsymbol{J}}_b^{-1}$

  Save $\tilde{\boldsymbol{\beta}}_b$ at the speed layer, $\tilde{\boldsymbol{J}}_b$ and $\tilde{\phi}_b$ at the inference layers;

  Release data set D_b from the memory.

end

Return $\tilde{\boldsymbol{\beta}}_b$ and $\text{diag}\{\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b)\}$ for $b = 1, 2, \ldots$.

Figure 2.4: Pseudo code for the implementation of renewable GLM.

2.5.2 Examples

In addition to the first example of linear model discussed in Section 2.3.3, here we present the renewable estimation and its implementation in two popular generalized linear models: logistic model for binary outcomes and log-linear model for count outcomes.

Example II.7. (Logistic model). Assume data batch $D_b = \{\boldsymbol{y}_b, \boldsymbol{X}_b\}$ with binary outcomes $\boldsymbol{y}_b = (y_{b1}, \ldots, y_{bn_b})^T$ and covariates $\boldsymbol{X}_b = (\boldsymbol{x}_{b1}, \ldots, \boldsymbol{x}_{bn_b})^T$, where $y_{bi}|\boldsymbol{x}_{bi}$ are independently sampled from a Bernoulli distribution with probability of success $\pi_{bi} = P(y_{bi} = 1 \mid \boldsymbol{x}_{bi})$, and the dispersion parameter $\phi = 1$. A logistic regression

model takes the form $g(\pi_{bi}) = \log(\frac{\pi_{bi}}{1-\pi_{bi}}) = \boldsymbol{x}_{bi}^T\boldsymbol{\beta}$. The (unit) score function and negative Hessian matrix (also the observed information matrix) for data batch D_b are respectively given by

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

where $v_{bi}(\pi_{bi}) = \pi_{bi}(1-\pi_{bi}) = \frac{\exp(\boldsymbol{x}_{bi}^T\boldsymbol{\beta})}{\{1+\exp(\boldsymbol{x}_{bi}^T\boldsymbol{\beta})\}^2}$ is the variance function. The renewable estimate $\tilde{\boldsymbol{\beta}}_b$ and the aggregated observed information matrices $\tilde{\boldsymbol{J}}_b$ are updated according to the procedure given in Algorithm 2.4 and the Rho architecture in Figure 2.3.

Example II.8. (Poisson log-linear model). Assume data batch $D_b = \{\boldsymbol{y}_b, \boldsymbol{X}_b\}$, with outcomes of counts $\boldsymbol{y}_b = (y_{b1}, \ldots, y_{bn_b})^T$ and covariates $\boldsymbol{X}_b = (\boldsymbol{x}_{b1}, \ldots, \boldsymbol{x}_{bn_b})^T$, and $y_{bi}|\boldsymbol{x}_{bi}$ are independently sampled from a Poisson distribution with mean $\mu_{bi} = \mathbb{E}(y_{bi}|\boldsymbol{x}_{bi})$ that is specified by a log-linear model $g(\mu_{bi}) = \log(\mu_{bi}) = \boldsymbol{x}_{bi}^T\boldsymbol{\beta}$. Here the dispersion parameter $\phi = 1$. The (unit) score function and negative Hessian matrix (also the observed information matrix) for data batch D_b are given by, respectively,

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

where $v_{bi} = \mu_{bi} = \exp(\boldsymbol{x}_{bi}^T\boldsymbol{\beta})$ is the variance function. Again, the renewable estimate $\tilde{\boldsymbol{\beta}}_b$ and the aggregated observed information matrices $\tilde{\boldsymbol{J}}_b$ are produced in the Rho architecture, respectively, at the speed layer and the inference layer, also presented in Algorithm 2.4 and Figure 2.3.

2.6 Simulation Experiments

2.6.1 Setup

We conduct several simulation experiments to assess the numerical performance of the proposed renewable estimator and incremental inference in the settings of linear

and logistic models. We compare our method with AI-SGD and additional existing methods, including (i) the oracle MLE obtained by processing the entire data once, (ii) sequential estimation method of online LSE in the linear model, and (iii) sequential estimation method of CEE/CUEE for the logistic model.

These methods are compared thoroughly in the aspects of parameter estimation, computation efficiency, and hypothesis testing. The evaluation criteria for parameter estimation include (a) absolute bias (A.bias), (b) asymptotic standard error (ASE), (c) empirical standard error (ESE) and (d) coverage probability (CP). We use the MLE yielded from the R package `glm` as the gold standard in all comparisons. Computation efficiency is also assessed by (e) computation time (C.Time) and (f) running time (R.Time). R.time accounts only for the data processing time, while C.time includes time spent on both loading data streams and processing data. Note that in the case of AI-SGD, one data point is run at one iteration, thus the data loading time cannot be properly captured. In this case, we consider only R.time for AI-SGD.

In all the simulation experiments considered in Tables 2.1-2.3, we set a terminal point B, and we generate the full dataset $D_B^\star$ with N_B observations independently from the respective GLMs with the mean model $\mathbb{E}(y_i|\boldsymbol{x}_i) = g(\boldsymbol{x}_i^T\boldsymbol{\beta}_0)$, $i = 1, ..., N_B$. We set $\boldsymbol{\beta}_0 = (0.2, -0.2, 0.2, -0.2, 0.2)^T$, and $\boldsymbol{x}_{i[2:5]} \sim \mathcal{N}_4(\mathbf{0}, \boldsymbol{V}_4)$ independently where $\boldsymbol{V}_4$ being a 4×4 compound symmetry covariance matrix with correlation $\rho = 0.5$, and intercept $\boldsymbol{x}_{i[1]} = 1$.

2.6.2 Evaluation of Parameter Estimation

Scenario 1: fixed B and 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 computation efficiency. These methods include (A) MLE, (B) AI-SGD, (C) online LSE for the linear model, or CEE/CUEE for the logistic model, and (D) Renewable estimation (Renew). There are B data

Table 2.1: Simulation results under the linear model with fixed $N_B = 100,000$ and $p = 5$ with varying batch sizes n_b.

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

Table 2.2: Simulation results under the setting of $N_B = 100,000$ and $p = 5$ for logistic model with varying batch size n_b.

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

streams, each with data batch size n_b, and the total of $N_B = |D_B^\star| = 100,000$ independent observations, which are simulated from a GLM with the mean model specified in Section 4.6.1. Tables 2.1 and 2.2 report the evaluation criteria for the linear and logistic models, respectively, over 500 rounds of simulations. Additional simulation results in the linear and logistic models with other values of n_b may be found in Tables A.3 and A.4 in the Supplementary Material. Also, comparison results in the case of the Poisson log-linear model are listed in Table A.5 in the Supplementary Material.

Bias and coverage probability. In the linear regression, due to the fact that the LSE is a linear function of data, it can be perfectly decomposed across data batches. Thus, MLE, online LSE and Renew are identical, leading to exactly the same bias and coverage probability, shown in Table 2.1. It is easy to see that both bias and coverage probability in the linear model are not affected by data batch size n_b. From Table 2.2 with the logistic regression, our renewable estimation always exhibits similar performances to the oracle MLE, and appears quite robust to different n_b. In contrast, CEE appears numerically unstable; as sample size of single batch n_b decreases to 200, its coverage probability drops down below 90%. Even though the

CUEE is proposed to improve the CEE (*Schifano et al.*, 2016), bias of CUEE appears much larger than that of the MLE as n_b decreases to 50. In addition, CUEE has much larger empirical standard error than that of CEE as n_b gets smaller. AI-SGD processes a single observation each time (*i.e.* $n_b = 1$ for $b = 1, \ldots, B$), so its bias, estimated and empirical standard errors are not related to batch size n_b. but all of them are constantly larger than those of the MLE or our renewable estimation. Even though the coverage probability of AI-SGD is 0.92, it does not preserve the nice property of MLE; see also the supplementary Tables A.3 to A.5.

Computation time. Two metrics are used to evaluate computation efficiency: "C.Time" in Tables 2.1 (as well as in the supplementary Tables A.4 and A.5) refers to the total amount of time required by data loading and algorithm execution, while "R.Time" is the amount of time required only for algorithm execution. With an increased B, our renewable estimation method shows clearly advantageous for much lower computation time over the three existing competitors, MLE, CEE and CUEE. AI-SGD is very competitive with noticeable computation efficiency, due to the fact that it avoids matrix inversion calculation in the algorithm, which, however, sacrifices to larger estimation bias, possibly leading to problematic inference if it were available. As pointed out above, we are not able to evaluate data loading time for AI-SGD, since it passes single data point one at a time.

Scenario 2: fixed batch size n_b but varying B

Now we turn to an interesting scenario where streaming datasets arrive at a high speed. For convenience, we fix batch size $n_b = 100$, but let N_B increase from 10^3 to 10^6. Table 2.3 lists the summaries of simulation results under the logistic model specified in Section 4.6.1.

Bias and coverage probability. When the batch size is as small as $n_b = 100$, increasing N_B does not seem to help reduce the estimation bias of CEE or CUEE estimates, and such bias exacerbates as more data streams are processed, resulting

in clearly problematic performances of statistical inference. When the number of data batches B increases to 1000, the coverage probability by CEE or CUEE remains steadily below 90%, with no sign of improvement in response to increased amount of data. It is striking to notice that when B is further increased to 10^4, the coverage probability of CUEE falls significantly down to 67%, while CEE gives the worst 0% coverage probability. This confirms that when the condition $B = \mathcal{O}(n_j^k)$, $k < 1/3$, is violated, CEE/CUEE will not have valid asymptotic distribution for inference. In contrast, our proposed method confirms the large sample properties similar to those of the oracle MLE: the average absolute bias decreases rapidly as the total sample size accumulates, and the coverage probability stays robustly around 95%. For competitor AI-SGD, the estimated standard error is much smaller than the empirical one and coverage probability is only 83% when $N_B = 10^3$. Similar to the results in Scenario 1, when N_B reaches 10^5 so that the coverage probability is around 95%, both bias and estimated standard error are much larger than those of MLE or our renewable method. Even worse, its bias stops decreasing after a certain level; for example, it 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); that is, once AI-SGD reaches a convergence phase, the subsequent estimates will jitter around the true parameter within a ball of slowly decreasing radius.

Computation time. Our renewable estimation method shows more and more advantageous as N_B increases: the combined amount of time for data loading and algorithm execution only takes less than 10 seconds, whereas the oracle MLE, when processing a total of 10^6 samples once, requires more than 5 minutes. This 35-fold faster computation by the proposed renewable estimation method does not cost any price of estimation precision and inference power. In addition, the running time for our renewable method and AI-SGD are comparable even under large sample size set-

tings such as $N_B = 10^5$ and 10^6. Once again, AI-SGD produces much larger bias and standard errors than our method. The extra small amount of time used by our renewable method on updating info.mats at the inference layer is computationally worthwhile for the proposed incremental inference.

Scenario 3: fixed N_B and B but varying large p

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

Bias and coverage probability. Table 2.4 summarizes the simulation results over 200 replications. Our renewable method has the same level of bias as the oracle MLE in this high-dimensional logistic regression. In this setting with $n_j \leq 10p$, CEE and CUEE both do not provide reliable coverage probabilities due to largely severe biases. AI-SGD has the largest bias that is more than 10 times that of MLE but very small standard errors, which may be caused by local trapping points that AI-SGD gets stuck. Consequently, estimated standard error based on perturbation resampling method will also be too small, resulting in 0% coverage probability. As also pointed out in *Fang* (2019), their method may not be able to deal with high-dimensional large-scale data.

Computation time. For large $p = 1000$ or 2500, our renewable estimation method is at least 4-fold faster than the oracle MLE, and this computation efficiency gain repeats in the low dimension case ($p = 5$) shown in Table 2.3. Although the AI-SGD runs faster than our renewable method, it is not applicable to the setting with very large p. The resulting severely large bias and small estimated standard error cannot

Table 2.3: Compare different estimators in logistic model with fixed batch size $n_b = 100$ and $p = 5$, N_B increases from 10^3 to 10^6.

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

Table 2.4: Compare different estimators in logistic regression models with a fixed total sample size $N_B = 2 \times 10^5$, each data batch size $n_b = 10^4$ and $B = 20$ batches. The number of covariates, p, increases from 1000 to 2500.

			$p = 1000$		
	AI-SGD	MLE	CEE	CUEE	Renew
A.bias	25.799	2.176	3.880	2.242	2.152
ASE	1.70×10^{-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
C.Time(min)	-	17.959	17.288	20.470	4.207
R.Time(min)	1.609	16.686	17.093	20.258	4.014
			$p = 2500$		
	AI-SGD	MLE	CEE	CUEE	Renew
A.bias	16.386	2.212	6.994	2.581	2.192
ASE	1.67×10^{-3}	2.728	3.475	2.523	2.789
ESE	1.63×10^{-3}	2.745	1.804	2.442	2.715
CP	**0**	0.946	0.561	0.874	0.954
C.Time(min)	-	126.407	122.528	149.411	31.451
R.Time(min)	4.737	123.904	122.037	148.924	30.917

provide any reliable estimation or valid inference.

The above simulation results clearly suggest that our proposed renewable estimation method can produce real-time robust and reliable estimation and inference that are similar to the oracle MLE that processes the entire data once, regardless of low or high dimension p. The high computation efficiency of the proposed method is clearly preferred to any existing methods when data streams arrive at a high speed. Instead of invoking a meta-type estimation procedure as done by CEE and CUEE, our renewable estimation method directly solves a cumulative estimating equation that is of second-order equivalency to the cumulative log-likelihood function from which the MLE is yielded. Thus, a global optimality is guaranteed for the proposed method. Consequently, unlike CEE or CUEE, our renewable estimation does not require control the relative scale of B and n_b, and thus is more desirable.

Note that the running time complexity of our method is $O\left(N_B p^2 + B p^3/3\right)$. When $p < n_b$, it reduces to $O(N_B p^2)$, a typical order of second-order online methods. When N_B is fixed and p is large, increasing data batch size n_b will make B small, and

thus greatly improves computation efficiency. This evidence has been seen in both Tables 2.1 and 2.4.

2.6.3 Evaluation of Hypothesis Testing

Now we evaluate the performance of the proposed incremental inference based on the Wald test available at the inference layer in the Rho architecture. We run a simulation study on the Wald test for $H_0 : \beta_{01} = 0.2$ vs. $H_A : \beta_{01} \neq 0.2$, where β_{01} is the regression coefficient of the intercept in the logistic model used in Tables 2.2 and 2.3. With the $\boldsymbol{\beta}^{\text{null}} = (0.2, -0.2, 0.2, -0.2, 0.2)^T$, set $\boldsymbol{\beta}_a = (\beta_{a1}, -0.2, 0.2, -0.2, 0.2)^T$ with 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 slowromancapi@ error) and power (1−type slowromancapii@ error) of the Wald test in equation (2.20) proposed in Section 2.4.2. Based on simulated data streams, with $N_B = 100,000$, each data batch with size $n_b = 200$, we calculated empirical type slowromancapi@ error and power from 500 replications.

Under H_0, as shown in Figure A.2 (the (1,1)-th panel) in the Supplementary Material, the Q-Q plot of the Wald test statistic from a total of 500 replications is distributed closely along the 45° diagonal, indicating the validity of asymptotic χ^2_1 distribution. In addition, we increase the number of coefficients being tested, and found that under H_0 the resulting Wald statistics are all shown to approximately chi-square distributed with degrees of freedom equal to the number of parameters under the alternative H_A (see the remaining panels of Figure A.2). The supplementary Table A.6 shows the empirical type I error and power based on 500 replications, where the type I errors of the Wald test for $H_0 : \beta_{01} = 0.2$ by the MLE, AI-SGD and our proposed Wald test are very close to the nominal level of 0.05, while the Wald tests based on CEE and CUEE have poor type I error control. Figure 2.5 shows that the power of AI-SGD is consistently much lower than that of the Wald test based on the renewable estimation or the MLE, while CEE or CUEE has lower power when

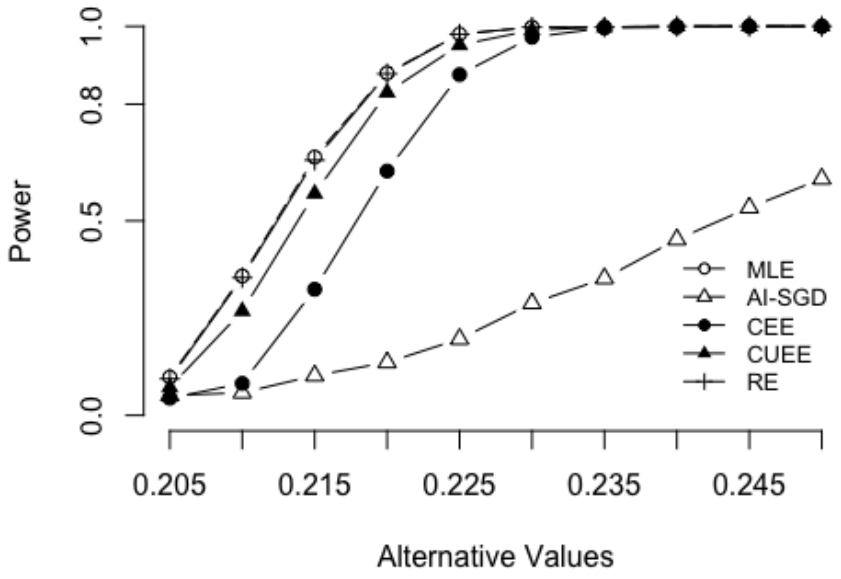

Figure 2.5: Power curves of the Wald tests.

β_{a1} has a smaller distance to $\beta_{01} = 0.2$.

2.7 Data Example

To show the usefulness of our proposed renewable estimation and incremental inference in practice, we analyzed streaming data from the National Automotive Sampling System-Crashworthiness Data System (NASS CDS). Our primary interest was to evaluate the effectiveness of graduated driver licensing (GDL), which is a nationwide legislature on novice drivers of age 21 or younger with various conditions of vehicle operation. On the other hand, there are no restrictions on vehicle operation for older drivers in the current law. To address the effect of driver's age on driving safety, we compared age groups with respect to the risk of fatal crash when an accident occurred. Three age groups were considered: "Age$<$ 21" represented the young group with a restricted GDL, while "Age$\geq$ 65" was the old group with a regular driver's

license, and those falling in between was treated as the reference group. Three age groups (Age$<$ 21, 21 $\leq$ Age $<$ 65, Age$\geq$ 65) were coded as dummy variables in our analysis. Since the number of drivers involved in accidents in the young group or the old group was much smaller than those in the reference group, it was of great interest to renew analysis results with more data being collected sequentially over time. Event of "Fatality" in a crash is a binary outcome of interest, which was analyzed using a logistic model. This outcome variable was created from the variable of Maximum Treatment in Accident (ATREAT) in the database, which indicated the most intensive treatment given to driver in an accident.

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

As shown in Figure 2.6, the 95% pointwise confidence bands over the 84 batches became narrower for all regression coefficients as more monthly streaming data batches arrived for the analysis. The top two panels display the traces plots for coefficient estimates obtained by the renewable estimation method for young and old groups, respectively. The coefficient estimates for the young group stay below 0 over the 84-month period, meaning that the young group (Age$<$ 21) has lower adjusted odds of fatal crash than the reference group. This finding is consistent with the reported results in the literature that GDL is an effective policy to protect novice drivers from severe injuring (e.g. *Chen et al.* (2014)). In contrast, the trace plot for the old age group (Age$\geq$ 65) shows an upward trend and get stabilized when the sample size

increases. This suggests that the adjusted odds of fatality in a vehicle crash for the old group becomes significant higher than the reference group when the number of streaming data batches is large enough. This may suggest a need on policy modification on restrictive vehicle operation for old drivers.

Figure 2.7 shows the trends of $-\log_{10}(p)$ values, p-values of the incremental Wald test in the 10-base logarithm, for each regression coefficient over 84 months. Clearly, all their levels of evidence against the null $H_0 : \beta_j = 0$ are increasing over time. "Seat Belt" turns out to have the strongest association to the odds of fatality in a crash among all covariates included in the model. This is an overwhelming confirmation to the enforcement of policy "buckle up" when sitting in a moving vehicle. In addition, to characterize the overall significance level for each covariate over the 84-month period, we proposed to calculate a summary statistic as of area under the p-value curve. Most of these curves have well separated patterns, so that the ranking of the overall significance by the calculated areas is well aligned with the ranking of p-values obtained at the end time of streaming data availability, namely December, 2015. It is interesting to note that "Traffic Control Function", "Light Condition" and "Sex" are among the weakest predictors.

Applying the proposed renewable estimation and incremental inference to the above CDS data analysis enabled us to visualize time-course patterns of data evidence accrual as well as stability and reproducibility of inference. As shown clearly in Figure 2.6, at the early stage of data streams, due to limited sample sizes and possibly sampling bias, both parameter estimates and test power may be unstable and even misleading. These potential shortcomings can be convincingly overcome when estimates and inferential quantities were continuously updated along with data streams, which eventually reached stability and reliable conclusions. Table 2.5 reports the results of the renewable estimation and incremental inference at the terminal time of these streaming data. Our proposed Rho architecture has made the above incremen-

Table 2.5: Results from the MLE method and the proposed renewable estimation method in logistic model with $N = 23,184, p = 9, B = 84$.

	MLE			Renew		
	Estimate	ASE	p-value	Estimate	ASE	p-value
Intercept	-4.284	0.174	3.91×10^{-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}

tal analysis straightforward. As a matter of fact, this expanded architecture with an addition of inference layer has given rise to tremendous convenience in data storage and data analytics for processing high-throughput streaming data.

2.8 Concluding Remarks

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

The renewable methodology for estimation and inference is of second-order approximation to the oracle MLE. It can sequentially renew both point estimation and asymptotic normality along data streams. We proposed a new Rho architecture for implementation, which is an extended Apache Spark Lambda architecture, with an added inference layer that carries out the storage and updating of information matrices. Both proposed statistical methodology and computational algorithms have been

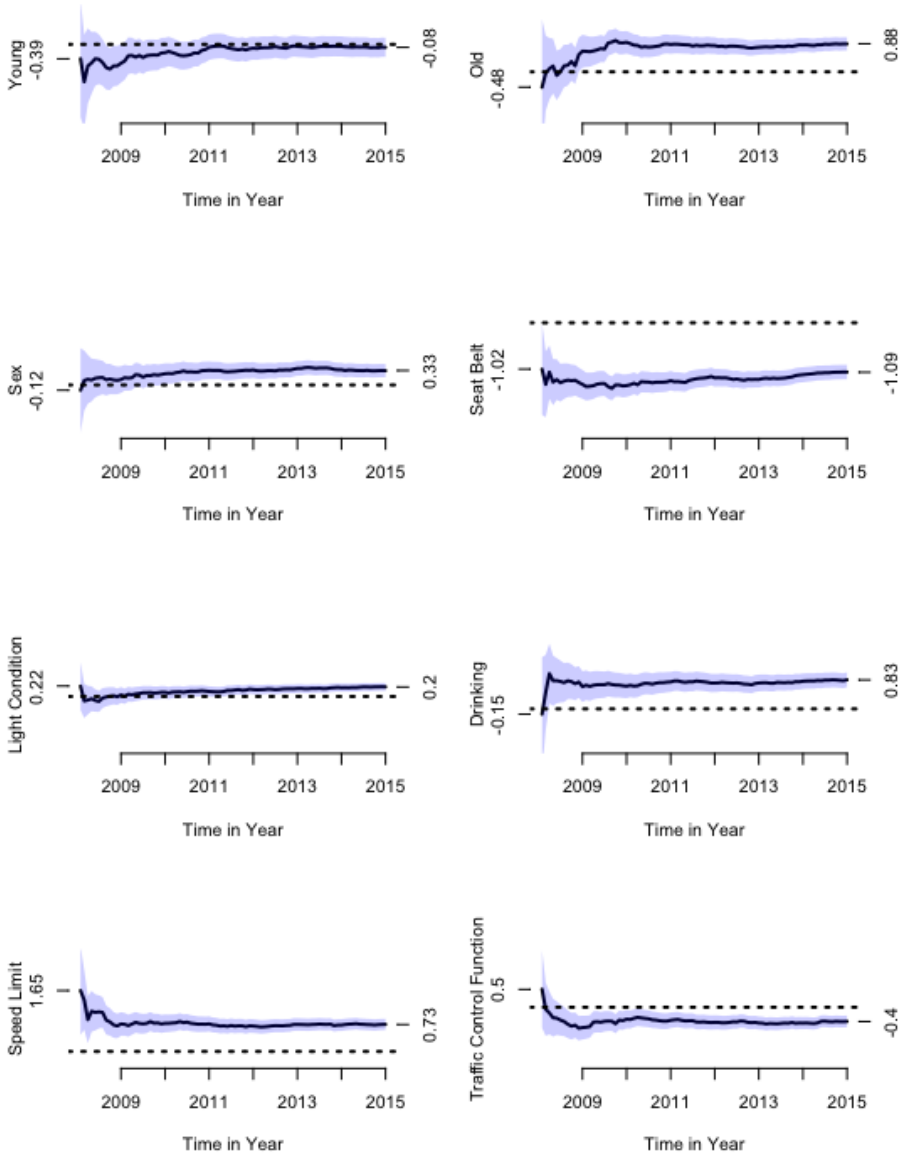

Figure 2.6: Trace plot for the coefficients estimates and 95% confidence bands.

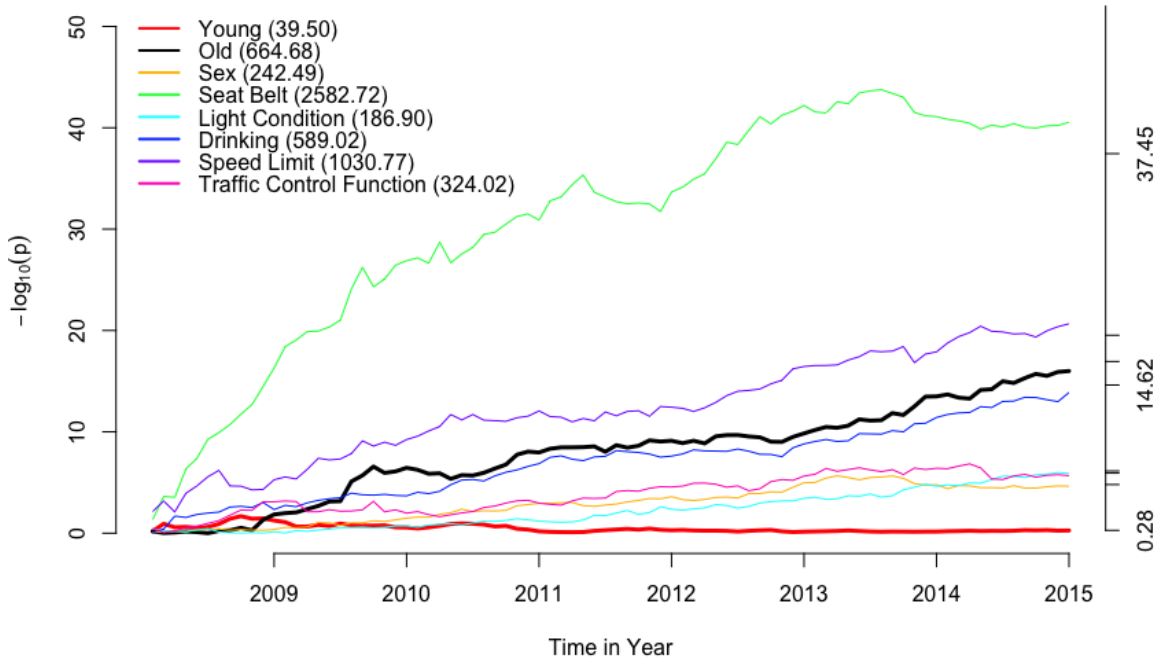

Figure 2.7: Trace plot of $-log_{10}(p)$ during January, 2009 to December, 2015.

justified theoretically and examined numerically in the setting of generalized linear models. Being a key methodology contribution, the incremental inference has been shown to be statistically valid and efficient; it has no loss of efficiency comparing to the oracle method of the maximum likelihood that processes the entire data once, while enjoys high computational efficiency. We demonstrate the connection of renewable estimators to approximate sufficient statistics, which builds a bridge between the classical statistical theory and modern online learning analytics.

Through various simulation studies, we have shown that our proposed renewable estimation method runs computationally faster than two existing methods CEE and CUEE, as well as the oracle MLE. The reason that CUEE is slower than CEE is that it requires one extra step calculation involving an intermediary estimator. Our proposed incremental updating algorithm keeps using the same inversed Hessian matrix over all iterations, where the matrix inverting operation is only carried out once per data batch, leading to tremendous saving of computation time. It is worth pointing out that the estimation consistency of CEE or CUEE is established under strong regularity conditions concerning the constraint on the ratio of batch size n_b to the number of data batches B. Such conditions may not hold in some real applications when data streams arrive perpetually. Our method has overcome this restriction, as confirmed in the second simulation study. In addition, the renewable estimator is found to be appealing especially when the number of data batches accumulates quickly, with both advantages on the asymptotic properties being virtually identical to the oracle MLE and highly efficient information storage and processing via the Rho architecture. Reliability of statistical inference is of great importance in practice to handle data streams, such as Phase IV clinical trials where drug safety, side-effect and efficacy have to be assessed at the general population mobile health data analysis, as well as traditional sensor networks, web logs and computer network traffic (*Gaber et al.*, 2005).

The proposed renewable estimation method to sequentially handle data sets may be treated as an appealing alternative approach 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 needed sufficient statistics, X^TWX and XWZ, in sequential increments and then summing them up in the Iterative Weighted Least Square (IWLS) algorithm. However, these two methods must use historical subject-level data in calculations. Thus, they are more expensive in data storage and computational inefficient in comparison to our proposed renewable estimation method. From this perspective, our renewable estimation method could also serve as a powerful alternative method to `biglm` and `speedglm`, and as well as to the parallel computing paradigm when analyzing very large static data.

The formulation of renewable estimation method is under the context of generalized linear model where the log-likelihood functions have nice properties such as the twice continuous differentiability. The analytic procedures related to the development of renewable estimator in the GLMs pave path to further generalization of such method to other important settings such as generalized estimating equations (GEEs), Cox regression model, and quantile regression model. Methods for developing renewable estimation for non-differentiable score functions such as quantile regression would be an interesting direction of research. In addition, this method is based on the assumption that data batches are all sampled from a common study population, which may be violated in some of practical studies. In this case of het-

erogeneous data streams, sequential updating procedures will be a challenging but useful methodology research topic, which is worth further exploration.

CHAPTER III

Real-time Regression Analysis of Streaming Clustered Data With or Without Data Contamination

3.1 Introduction

When a car accident happens, driver and passengers in the same car would be all likely to get injured, and their degrees of injury severity are correlated within the cluster (or car). The National Automotive Sampling System-Crashworthiness Data System (NASS CDS) is a publicly accessible source of streaming datasets containing car accident information in USA. In this paper, we consider a problem where a series of independent clustered-data batches becomes available sequentially. Similar to data of car accidents, each data batch consists of longitudinally correlated or cluster-correlated outcomes from subjects. Other examples of such streaming correlated data include cohorts of patients sequentially assembled from different clinical centers to periodically update national disease registry databases. The primary goal of processing streaming data is to be able to analyze and update statistics of interest upon the arrival of a new data batch, which enables to not only free up space for the storage of massive historical individual-level data, but also to provide real-time inference and decision making. Unfortunately, most of current available online learning methods

such as stochastic gradient descent algorithm (*Robbins and Monro*, 1951; *Toulis and Airold*, 2017) focus only on point estimation or prediction; statistical inference is lacking in the existing arsenal, especially in correlated data analysis. In effect, interim statistical inference and decision making are essential in many applied fields where sequentially adaptive intervention to observed outcomes are of critical importance to increase treatment effectiveness and to enforce quality control of production lines.

This chapter begins with an extension of renewable estimation and incremental inference in generalized linear models (GLMs) from the case of cross-sectional data in Chapter II to that of repeatedly measured responses. This extension is undertaken not only to relax the availability of likelihood function in the renewable estimation to estimating functions, but also to relax the assumption of homogeneous marginal models with no occurrences of abnormal data batches. In real world applications, practitioners often encounter outlying data batches that are not generated from the same underlying model of interest over the course of data streams. In this case, continually updating results without noticing abnormal data batches would lead to invalid statistical inference and misleading conclusions. Consequently, our second objective of this paper is to develop a quality control type of monitoring scheme by the means of change-point detection.

The method of change-point detection has been extensively studied in recent literature. There are two major classes of problem formulations according to online and offline platforms (*Poor and Hadjiliadis*, 2008; *Basseville and Nikiforov*, 1993). The class of offline methods includes procedures of estimating abrupt changes in a single static dataset, most of which are based on certain regularization algorithms such as fused LASSO and group LASSO, among others. For instance, *Harchaoui and Leduc* (2010) propose an adaption of the LASSO algorithm to detect changes in a sequence of mean values of one-dimensional Gaussian random variables; *Qian and Su* (2013) and *Angelosante and Giannakis* (2012) adopt group fused LASSO to examine

structural changes in the linear regression; *Rojas and Wahlberg* (2014) investigate properties of the fused LASSO method in the context of change-point detection for piece-wise signals; *Zhang et al.* (2015) determine change-points that would segment data into different subgroups as well as enforce sparsity within each subgroup via the sparse group LASSO.

In contrast, for the class of online methods, due to the fact that data batches arrive sequentially, the goal of monitoring is slightly different. The primary interest is to run real-time algorithms to detect abnormal changes detrimental to an online learning platform in a timely fashion with no delay, while the false-alarm rate or type I error is being properly controlled (*Johari et al.*, 2016). Such task is technically challenging, and it is the setting that is considered in this paper. In a rather simple case where both pre-change and post-change distributions are fully specified, various detection rules have been studied based on classical procedures, including Shewhart's control charts (*Amin et al.*, 1995), moving average control charts (*Amin and Search*, 1991), the CUSUM procedure (*Page*, 1954), and the Shiryayev-Roberts (SR) procedure (*Shiryayev*, 1963; *Roberts*, 1966), among others.

In the literature of statistical quality control, without loss of generality, it is often the case that pre-change distribution is typically assumed to be well defined. If not, one may first use a set of training samples to obtain point estimates of unknown pre-change parameters (*Pollak and Siegmund*, 1991). Then, Generalized Likelihood Ratio (GLR) test or as such (*Goel and Wu*, 1971) are widely used to derive online detection procedures, which require to specify pre- and post-change distributions at a point under examination. This assumption of known pre- and post-change distributions is too restrictive and unrealistic. The online version of GLR statistic needs to find maximum likelihood estimate (MLE) of unknown post-change parameters whenever a new data batch arrives in order to obtain the detection statistic.

To comply with the computation efficiency in the online paradigm, a certain re-

cursive form is certainly desirable to produce an update quickly with a new data batch. In general, MLE does not take a recursive form and cannot be updated using simple summary statistics (*Lai*, 2004). This technical gap has been filled by the content in Chapter II. An extension of this to streaming correlated data is one of the focuses in this paper. In addition, we relax the quality of data streams by allowing occurrences of abnormal data batches over streaming data collection. By an "abnormal" data batch we mean a dataset that is generated with a different set of regression coefficients from those of the model of primary interest. Our goal is to identify any abnormal data batch and exclude it from updating results of estimation and inference.

We focus on the generalized estimating equation (GEE) approach proposed by *Liang and Zeger* (1986), one of the most widely used methods for the analysis of data with correlated outcomes. This quasi-likelihood approach is based only on the first two moments of the correlated data with no need of specifying a parametric joint distribution. Consequently, likelihood is no longer available, and thus no GLR test statistic would be possibly formed for change-points detection as done extensively in the current literature. To address this technical challenge, we utilize quadratic inference function (QIF), another quasi-likelihood inference known in the analysis of longitudinal or clustered data (*Qu et al.*, 2000). QIF has several advantages in comparison to GEE: (i) QIF does not require more model assumptions than GEE; (ii) it provides a goodness-of-fit test for the first moment assumption, i.e. the mean-model specification; (iii) QIF estimator is more efficient than the GEE estimator when the working correlation is misspecified; and (iv) it is more robust with a bounded influence function against large outliers (*Qu and Song*, 2004).

For the implementation of online QIF, in the presence of potential abnormal data batches, we expand the Rho architecture developed in GLM in Chapter II with a new addition of monitoring layer in the Spark's Lambda architecture, where a QIF-based

test procedure is housed to check the compatibility of each upcoming data batch with the previous ones. Specifically, we aim to develop a new methodology with the following tasks. (i) To put forward renewable QIF estimation and incremental inference in marginal GLMs for correlated outcomes; and (ii) to investigate a QIF-based goodness-of-fit test statistic in the monitoring layer that enables to effectively detect abnormal data batches over data streams with no fixed ending monitoring time.

Our new methodology contributions include: (i) we propose a new quadratic inference function method that allows to perform real-time regression analysis with correlated outcomes in quadratic inference functions; (ii) the proposed renewable QIF estimator is asymptotically equivalent to the oracle QIF estimator obtained from the full cumulative data in the sense that their ℓ_2-norm difference decreases as the total sample size N_B increases; (iii) this renewable QIF method can be implemented in the existing Spark's Lambda architecture; and (iv) by adding a monitoring layer to the Lambda architecture, our method allows to detect abnormal data batches in real-time while conducting online correlated data analysis.

This chapter is organized as follows. Section 3.2 provides both algorithms and theoretical guarantees for our renewable QIF method. Section 3.3 presents a QIF-based test statistic for detection of abnormal data batches with relevant power analysis. Section 3.4 discusses an extended Lambda architecture with an addition of quality control layer and pseudo code for numerical implementation. Section 3.5 includes simulation results with comparisons of the proposed renewable QIF to the oracle GEE, QIF and renewable GEE with or without abnormal data batches. Section 3.6 illustrates the proposed method by a real data analysis application. All technique details are included in the appendix, including the derivation of Renewable GEE method, the proofs of estimation consistency, asymptotic normality, and the asymptotic equivalence between the RenewQIF and the oracle QIF.

3.2 Renewable QIF methodology

3.2.1 Formulation

Consider independent streaming datasets of cluster-correlated outcomes, sequentially generated from a common underlying marginal generalized linear model with an unknown regression parameter $\boldsymbol{\beta}_0 \in \mathbb{R}^p$. For the ease of exposition, we assume an equal cluster size $m_i = m$. Our goal is to evaluate population-average effects of p covariates, denoted by $\boldsymbol{\beta}_0 = (\beta_{01}, \ldots, \beta_{0p})^T$ in the marginal mean model, $\mathbb{E}(\boldsymbol{y} \mid \boldsymbol{X}) = \boldsymbol{\mu} = \left[h(\boldsymbol{x}_1^T\boldsymbol{\beta}_0), \ldots, h(\boldsymbol{x}_m^T\boldsymbol{\beta}_0)\right]^T$, where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)^T$ with $\mu_k = h(\boldsymbol{x}_k^T\boldsymbol{\beta}_0)$, $\boldsymbol{x}_k = (x_{k1}, \ldots, x_{kp})^T$, $k = 1, \ldots, m$, and the within-cluster dependence is specified by $\mathrm{Cov}(\boldsymbol{y} \mid \boldsymbol{X}) = \phi\boldsymbol{\Sigma}(\boldsymbol{\beta}_0, \boldsymbol{\alpha}) = \phi\boldsymbol{A}^{1/2}\boldsymbol{R}(\boldsymbol{\alpha})\boldsymbol{A}^{1/2}$. Here $h(\cdot)$ is a known link function, ϕ is a dispersion parameter, $\boldsymbol{A} = \mathrm{diag}\,\{v(\mu_1), \ldots, v(\mu_m)\}$ is a diagonal matrix with $v(\cdot)$ being a known variance function, and $\boldsymbol{R}(\boldsymbol{\alpha})$ is a working correlation matrix that is fully characterized by a correlation parameter vector $\boldsymbol{\alpha}$.

In the context of streaming data, consider a time point $b \geq 2$ with a total of N_b clusters arriving sequentially in b batches, $\mathcal{D}_1, \ldots, \mathcal{D}_b$, each containing $n_j = |\mathcal{D}_j|$, $j = 1, \ldots, b$, clusters. Let $\mathcal{D}_b^\star = \{\mathcal{D}_1, \ldots, \mathcal{D}_b\}$ denote the accumulated collection of clustered datasets with clustered outcomes up to data batch b, and $N_b = |\mathcal{D}_b^\star|$. For simplicity $\mathcal{D}_b$ (a single data batch b) or $\mathcal{D}_b^\star$ (an aggregation of b data batches) is used as respective sets of indices for clusters involved. $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{im})^T$ and $\boldsymbol{X}_i = (\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{im})^T$ are the correlated response vectors and associated covariates, $i = 1, \ldots, n_j$, $j = 1, \ldots, b$. According to *Liang and Zeger* (1986), a GEE estimator of $\boldsymbol{\beta}_0$ is a solution to the following generalized estimating equation for data $\mathcal{D}_b^\star$ up to time point b:

$$\boldsymbol{\psi}_b^\star(\mathcal{D}_b^\star; \boldsymbol{\beta}, \alpha) = \sum_{i \in \mathcal{D}_b^\star} \boldsymbol{D}_i^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}_i) = \boldsymbol{0}, \tag{3.1}$$

where $\boldsymbol{\mu}_i = (\mu_{i1}, \ldots, \mu_{im})^T$, $\boldsymbol{D}_i = \partial\boldsymbol{\mu}_i/\partial\boldsymbol{\beta}^T$ is an $m \times p$ matrix and $\boldsymbol{\Sigma}_i = \boldsymbol{A}_i^{1/2}\boldsymbol{R}(\boldsymbol{\alpha})\boldsymbol{A}_i^{1/2}$ with $\boldsymbol{A}_i = \mathrm{diag}\,\{v(\mu_{i1}), \ldots, v(\mu_{im})\}$. According to *Qu et al.* (2000), the formula-

tion of QIF is based on an approximation to the inverse working correlation matrix by $\boldsymbol{R}^{-1}(\boldsymbol{\alpha}) \approx \sum_{s=1}^{S} \gamma_s \boldsymbol{M}_s$, where $\gamma_1, \ldots, \gamma_S$ are constants possibly depend on $\boldsymbol{\alpha}$, and $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_S$ are known basis matrices with elements 0 and 1, which are determined by a given correlation matrix $\boldsymbol{R}(\boldsymbol{\alpha})$. In some cases, the above expansion can be exact. For example, as discussed in *Qu et al.* (2000) and *Song* (2007, Chapter 5), basis matrices for the compound symmetry working correlation matrix are $\boldsymbol{M}_1 = \boldsymbol{I}_m$ and $\boldsymbol{M}_2^{cs}$, a matrix with 0 on the diagonal and 1 off the diagonal. For AR-1 working correlation, three basis matrices include $\boldsymbol{M}_1 = \boldsymbol{I}_m$, $\boldsymbol{M}_2^{ar}$ with 1 on the two main off-diagonals and 0 elsewhere, and $\boldsymbol{M}_3$ with 1 on the corners $(1,1)$ and (m,m), and 0 elsewhere. Plugging such expansion into (3.1) leads to $\boldsymbol{\psi}_b^{\star}(\mathcal{D}_b^{\star}; \boldsymbol{\beta}, \alpha) = \sum_{i \in \mathcal{D}_b^{\star}} \sum_{s=1}^{S} \gamma_s \boldsymbol{D}_i^T \boldsymbol{A}_i^{-1/2} \boldsymbol{M}_s \boldsymbol{A}_i^{-1/2} (\boldsymbol{y}_i - \boldsymbol{\mu}_i) = \boldsymbol{0}$, which may be regarded as a combination of the following extended score vector of pS dimension:

$$\boldsymbol{g}_b^{\star}(\boldsymbol{\beta}) = \sum_{i \in \mathcal{D}_b^{\star}} \boldsymbol{g}(\boldsymbol{y}_i; \boldsymbol{X}_i, \boldsymbol{\beta}) = \sum_{i \in \mathcal{D}_b^{\star}} \begin{pmatrix} \boldsymbol{D}_i^T \boldsymbol{A}_i^{-1/2} \boldsymbol{M}_1 \boldsymbol{A}_i^{-1/2} (\boldsymbol{y}_i - \boldsymbol{\mu}_i) \\ \vdots \\ \boldsymbol{D}_i^T \boldsymbol{A}_i^{-1/2} \boldsymbol{M}_S \boldsymbol{A}_i^{-1/2} (\boldsymbol{y}_i - \boldsymbol{\mu}_i) \end{pmatrix}.$$

This is an over-identified estimating function, namely $dim(\boldsymbol{g}_b^{\star}(\boldsymbol{\beta})) > dim(\boldsymbol{\beta})$. To obtain an estimator of $\boldsymbol{\beta}_0$, following *Hansen* (1982)'s generalized method of moments (GMM), we take $\hat{\boldsymbol{\beta}}_b^{\star} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min}\, Q_b^{\star}(\boldsymbol{\beta})$ with

$$Q_b^{\star}(\boldsymbol{\beta}) = \boldsymbol{g}_b^{\star T}(\boldsymbol{\beta}) \left\{ \boldsymbol{C}_b^{\star}(\boldsymbol{\beta}) \right\}^{-1} \boldsymbol{g}_b^{\star}(\boldsymbol{\beta}), \tag{3.2}$$

where $\boldsymbol{C}_b^{\star}(\boldsymbol{\beta}) = \sum_{i \in \mathcal{D}_b^{\star}} \boldsymbol{g}(\boldsymbol{y}_i; \boldsymbol{X}_i, \boldsymbol{\beta}) \boldsymbol{g}(\boldsymbol{y}_i; \boldsymbol{X}_i, \boldsymbol{\beta})^T$ is the sample variance matrix of $\boldsymbol{g}_b^{\star}(\boldsymbol{\beta})$. Note that the nuisance parameter $\boldsymbol{\alpha}$ is not involved in (3.2) for the estimation of $\hat{\boldsymbol{\beta}}_b^{\star}$.

Instead of processing the cumulative data $\mathcal{D}_b^{\star}$ once as shown above, we may conduct the online estimation and inference via a sequentially recursive updating scheme.

In the proposed renewable estimation framework, let $\tilde{\boldsymbol{\beta}}_b$ be a renewable estimator, which is initialized by $\tilde{\boldsymbol{\beta}}_1 = \hat{\boldsymbol{\beta}}_1 = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, Q_1(\boldsymbol{\beta})$, namely the QIF estimate obtained with the first data batch. When data batch $\mathcal{D}_b$ arrives, a previous estimator $\tilde{\boldsymbol{\beta}}_{b-1}$ is updated to $\tilde{\boldsymbol{\beta}}_b$ using historical summary statistics of previous data batches $\mathcal{D}_{b-1}^{\star}$ and full observations of current data batch $\mathcal{D}_b$. After the completion of this updating, individual-level data of $\mathcal{D}_b$ is no longer accessible for the sake of data storage, but only updated estimate $\tilde{\boldsymbol{\beta}}_b$ and summary statistics are carried forward in future calculations. For the empirical version, we use $\boldsymbol{g}_b(\boldsymbol{\beta}) = \sum_{i\in\mathcal{D}_b} \boldsymbol{g}(\boldsymbol{y}_i; \boldsymbol{X}_i, \boldsymbol{\beta})$ to denote the extended score vector of data batch $\mathcal{D}_b$, clearly $\boldsymbol{g}_b^{\star}(\boldsymbol{\beta}) = \sum_{j=1}^{b} \boldsymbol{g}_j(\boldsymbol{\beta})$. Let its corresponding negative gradient and sample variance matrix of $\boldsymbol{g}_b(\boldsymbol{\beta})$ be $\boldsymbol{G}_b(\boldsymbol{\beta}) = -\sum_{i\in\mathcal{D}_b} \partial \boldsymbol{g}(\boldsymbol{y}_i; \boldsymbol{X}_i, \boldsymbol{\beta})/\partial \boldsymbol{\beta}^T$ and $\boldsymbol{C}_b(\boldsymbol{\beta}) = \sum_{i\in\mathcal{D}_b} \boldsymbol{g}(\boldsymbol{y}_i; \boldsymbol{X}_i, \boldsymbol{\beta})\boldsymbol{g}(\boldsymbol{y}_i; \boldsymbol{X}_i, \boldsymbol{\beta})^T$, respectively. In the theoretical framework, let the variability matrix and sensitivity matrix be denoted by $\mathbb{C}(\boldsymbol{\beta}) = \mathbb{E}_{\boldsymbol{\beta}}\left\{\boldsymbol{g}(\boldsymbol{y}; \boldsymbol{X}, \boldsymbol{\beta})\boldsymbol{g}^T(\boldsymbol{y}; \boldsymbol{X}, \boldsymbol{\beta})\right\}$ and $\mathbb{G}(\boldsymbol{\beta}) = \mathbb{E}_{\boldsymbol{\beta}}\left\{-\partial \boldsymbol{g}(\boldsymbol{y}; \boldsymbol{X}, \boldsymbol{\beta})/\partial \boldsymbol{\beta}^T\right\}$. In the process of renewable QIF, the same basis matrices are used all data batches.

3.2.2 Derivation

We begin the derivation with two batches, the second one $\mathcal{D}_2$ arriving after the first $\mathcal{D}_1$. This simple scenario can be easily generalized to many batch streams with little effort. According to *Qu et al.* (2000), a QIF estimator, $\hat{\boldsymbol{\beta}}_1 = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, Q_1(\boldsymbol{\beta})$ with $Q_1(\boldsymbol{\beta}) = \boldsymbol{g}_1^T(\boldsymbol{\beta})\left\{\boldsymbol{C}_1(\boldsymbol{\beta})\right\}^{-1}\boldsymbol{g}_1(\boldsymbol{\beta})$, satisfies the estimating equation:

$$\boldsymbol{G}_1^T(\hat{\boldsymbol{\beta}}_1)\left\{\boldsymbol{C}_1(\hat{\boldsymbol{\beta}}_1)\right\}^{-1}\boldsymbol{g}_1(\hat{\boldsymbol{\beta}}_1) = \boldsymbol{0}.$$

When $\mathcal{D}_2$ arrives, we aim to obtain the oracle QIF estimator, $\hat{\boldsymbol{\beta}}_2^{\star}$, based on the accumulated data $\mathcal{D}_2^{\star}$, that satisfies the estimating equation $\boldsymbol{G}_2^{\star}(\hat{\boldsymbol{\beta}}_2^{\star})^T\boldsymbol{C}_2^{\star}(\hat{\boldsymbol{\beta}}_2^{\star})^{-1}\boldsymbol{g}_2^{\star}(\hat{\boldsymbol{\beta}}_2^{\star}) = \boldsymbol{0}$, or equivalently

$$\left\{\boldsymbol{G}_1(\hat{\boldsymbol{\beta}}_2^{\star}) + \boldsymbol{G}_2(\hat{\boldsymbol{\beta}}_2^{\star})\right\}^T\left\{\boldsymbol{C}_1(\hat{\boldsymbol{\beta}}_2^{\star}) + \boldsymbol{C}_2(\hat{\boldsymbol{\beta}}_2^{\star})\right\}^{-1}\left\{\boldsymbol{g}_1(\hat{\boldsymbol{\beta}}_2^{\star}) + \boldsymbol{g}_2(\hat{\boldsymbol{\beta}}_2^{\star})\right\} = \boldsymbol{0}. \tag{3.3}$$

Clearly, solving (3.3) for $\hat{\boldsymbol{\beta}}_2^\star$ involves subject-level data from both batches $\mathcal{D}_1$ and $\mathcal{D}_2$ where $\mathcal{D}_1$ may no longer be accessible. Our renewable version of QIF estimation is able to handle this issue. To do so, we take the first-order Taylor expansions of the terms $\boldsymbol{g}_1(\hat{\boldsymbol{\beta}}_2^\star)$, $\boldsymbol{G}_1(\hat{\boldsymbol{\beta}}_2^\star)$ and $\boldsymbol{C}_1(\hat{\boldsymbol{\beta}}_2^\star)$ element-wise in (3.3) around $\hat{\boldsymbol{\beta}}_1$, given that these inferential quantities are differentiable, and yield

$$\begin{aligned} n_1^{-1}\boldsymbol{g}_1(\hat{\boldsymbol{\beta}}_2^\star) &= n_1^{-1}\boldsymbol{g}_1(\hat{\boldsymbol{\beta}}_1) + n_1^{-1}\boldsymbol{G}_1(\hat{\boldsymbol{\beta}}_1)(\hat{\boldsymbol{\beta}}_1 - \hat{\boldsymbol{\beta}}_2^\star) + O_p(\|\hat{\boldsymbol{\beta}}_1 - \hat{\boldsymbol{\beta}}_2^\star\|^2), \\ n_1^{-1}\boldsymbol{G}_1(\hat{\boldsymbol{\beta}}_2^\star) &= n_1^{-1}\boldsymbol{G}_1(\hat{\boldsymbol{\beta}}_1) + O_p(\|\hat{\boldsymbol{\beta}}_1 - \hat{\boldsymbol{\beta}}_2^\star\|^2), \\ n_1^{-1}\boldsymbol{C}_1(\hat{\boldsymbol{\beta}}_2^\star) &= n_1^{-1}\boldsymbol{C}_1(\hat{\boldsymbol{\beta}}_1) + O_p(\|\hat{\boldsymbol{\beta}}_1 - \hat{\boldsymbol{\beta}}_2^\star\|^2). \end{aligned} \tag{3.4}$$

The error term $O_p(\|\hat{\boldsymbol{\beta}}_1 - \hat{\boldsymbol{\beta}}_2^\star\|^2)$ in (3.4) may be asymptotically ignored if n_1 is large enough. Dropping such error terms, we propose a new QIF estimator $\tilde{\boldsymbol{\beta}}_2$ as a solution to the estimating equation of the form $\tilde{\boldsymbol{G}}_2(\tilde{\boldsymbol{\beta}}_2)^T\tilde{\boldsymbol{C}}_2(\tilde{\boldsymbol{\beta}}_2)^{-1}\tilde{\boldsymbol{g}}_2(\tilde{\boldsymbol{\beta}}_2) = \boldsymbol{0}$, or equivalently,

$$\Big\{\boldsymbol{G}_1(\hat{\boldsymbol{\beta}}_1) + \boldsymbol{G}_2(\tilde{\boldsymbol{\beta}}_2)\Big\}^T \Big\{\boldsymbol{C}_1(\hat{\boldsymbol{\beta}}_1) + \boldsymbol{C}_2(\tilde{\boldsymbol{\beta}}_2)\Big\}^{-1} \Big\{\boldsymbol{g}_1(\hat{\boldsymbol{\beta}}_1) + \boldsymbol{G}_1(\hat{\boldsymbol{\beta}}_1)(\hat{\boldsymbol{\beta}}_1 - \tilde{\boldsymbol{\beta}}_2) + \boldsymbol{g}_2(\tilde{\boldsymbol{\beta}}_2)\Big\} = \boldsymbol{0}, \tag{3.5}$$

where $\tilde{\boldsymbol{g}}_2$, $\tilde{\boldsymbol{G}}_2$ and $\tilde{\boldsymbol{C}}_2$ are, respectively, the resulting adjusted extended score vector, the aggregated negative gradient, and sample variance matrix. Thus, equation (3.5) updates the initial $\hat{\boldsymbol{\beta}}_1$ to be $\tilde{\boldsymbol{\beta}}_2$, and the latter $\tilde{\boldsymbol{\beta}}_2$, as shown in Theorem III.3, approximates the oracle QIF $\hat{\boldsymbol{\beta}}_2^\star$ up to the second order asymptotic error with respect to the cumulative sample size N_2. Because of this, $\tilde{\boldsymbol{\beta}}_2$ is called *a renewable QIF estimator* of $\boldsymbol{\beta}_0$, and equation (3.5) is termed as *an incremental QIF estimating equation*. Furthermore, it is straightforward to find the renewable QIF estimator $\tilde{\boldsymbol{\beta}}_2$ numerically via the Newton-Raphson algorithm. That is, at the $(r+1)$-th iteration,

$$\tilde{\boldsymbol{\beta}}_2^{(r+1)} = \tilde{\boldsymbol{\beta}}_2^{(r)} + \Big\{\tilde{\boldsymbol{G}}_2(\tilde{\boldsymbol{\beta}}_2^{(r)})^T\tilde{\boldsymbol{C}}_2(\tilde{\boldsymbol{\beta}}_2^{(r)})^{-1}\tilde{\boldsymbol{G}}_2(\tilde{\boldsymbol{\beta}}_2^{(r)})\Big\}^{-1} \tilde{\boldsymbol{G}}_2(\tilde{\boldsymbol{\beta}}_2^{(r)})^T\tilde{\boldsymbol{C}}_2(\tilde{\boldsymbol{\beta}}_2^{(r)})^{-1}\tilde{\boldsymbol{g}}_2(\tilde{\boldsymbol{\beta}}_2^{(r)}),$$

where $\tilde{\boldsymbol{G}}_2(\tilde{\boldsymbol{\beta}}_2^{(r)}) = \boldsymbol{G}_1(\hat{\boldsymbol{\beta}}_1) + \boldsymbol{G}_2(\tilde{\boldsymbol{\beta}}_2^{(r)})$ and $\tilde{\boldsymbol{C}}_2(\tilde{\boldsymbol{\beta}}_2^{(r)}) = \boldsymbol{C}_1(\hat{\boldsymbol{\beta}}_1) + \boldsymbol{C}_2(\tilde{\boldsymbol{\beta}}_2^{(r)})$. It is worth pointing out that in the above iterations, no subject-level data of $\mathcal{D}_1$, but only the historical summary statistics, including estimate $\hat{\boldsymbol{\beta}}_1$ and its negative gradient $\boldsymbol{G}_1(\hat{\boldsymbol{\beta}}_1)$ and sample variance matrix $\boldsymbol{C}_1(\hat{\boldsymbol{\beta}}_1)$, are used. Once again, the nuisance correlation parameter $\boldsymbol{\alpha}$ is not involved in the above iterations, either.

Generalizing the above procedure to a general setting of streaming datasets, we now define a renewable estimation of $\boldsymbol{\beta}_0$ as follows. Let $\hat{\boldsymbol{\beta}}_b^\star$ be the oracle QIF estimator of $\boldsymbol{\beta}_0$ with the accumulated data $\mathcal{D}_b^\star = \cup_{j=1}^b \mathcal{D}_j$ obtained from the cumulative QIF estimating equation $\boldsymbol{G}_b^\star(\hat{\boldsymbol{\beta}}_b^\star)^T \boldsymbol{C}_b^\star(\hat{\boldsymbol{\beta}}_b^\star)^{-1} \boldsymbol{g}_b^\star(\hat{\boldsymbol{\beta}}_b^\star) = \boldsymbol{0}$. A renewable estimator $\tilde{\boldsymbol{\beta}}_b$ of $\boldsymbol{\beta}_0$ is defined as a solution to the incremental QIF estimating equation: $\tilde{\boldsymbol{G}}_b(\tilde{\boldsymbol{\beta}}_b)^T \tilde{\boldsymbol{C}}_b(\tilde{\boldsymbol{\beta}}_b)^{-1} \tilde{\boldsymbol{g}}_b(\tilde{\boldsymbol{\beta}}_b) = \boldsymbol{0}$, which is equivalent to

$$\begin{aligned} &\left\{\sum_{j=1}^{b-1} \boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{G}_b(\tilde{\boldsymbol{\beta}}_b)\right\}^T \left\{\sum_{j=1}^{b-1} \boldsymbol{C}_j(\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{C}_b(\tilde{\boldsymbol{\beta}}_b)\right\}^{-1} \\ &\times \left\{\tilde{\boldsymbol{g}}_{b-1}(\tilde{\boldsymbol{\beta}}_{b-1}) + \sum_{j=1}^{b-1} \boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j)(\tilde{\boldsymbol{\beta}}_{b-1} - \tilde{\boldsymbol{\beta}}_b) + \boldsymbol{g}_b(\tilde{\boldsymbol{\beta}}_b)\right\} = \boldsymbol{0}, \end{aligned} \tag{3.6}$$

where $\tilde{\boldsymbol{G}}_b = \sum_{j=1}^b \boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j)$ is the aggregated negative gradient matrix, and $\tilde{\boldsymbol{C}}_b = \sum_{j=1}^b \boldsymbol{C}_j(\tilde{\boldsymbol{\beta}}_j)$ is the aggregated sample variance matrix. Solving (3.6) can be easily down via the following Newton-Raphson iterations:

$$\tilde{\boldsymbol{\beta}}_b^{(r+1)} = \tilde{\boldsymbol{\beta}}_b^{(r)} + \left\{\tilde{\boldsymbol{G}}_b(\tilde{\boldsymbol{\beta}}_b^{(r)})^T \tilde{\boldsymbol{C}}_b(\tilde{\boldsymbol{\beta}}_b^{(r)})^{-1} \tilde{\boldsymbol{G}}_b(\tilde{\boldsymbol{\beta}}_b^{(r)})\right\}^{-1} \tilde{\boldsymbol{G}}_b(\tilde{\boldsymbol{\beta}}_b^{(r)})^T \tilde{\boldsymbol{C}}_b(\tilde{\boldsymbol{\beta}}_b^{(r)})^{-1} \tilde{\boldsymbol{g}}_b(\tilde{\boldsymbol{\beta}}_b^{(r)}). \tag{3.7}$$

In equation (3.7), clearly we only use the subject-level data of current data batch $\mathcal{D}_b$ and summary statistics $\{\tilde{\boldsymbol{\beta}}_{b-1}, \tilde{\boldsymbol{g}}_{b-1}, \tilde{\boldsymbol{G}}_{b-1}, \tilde{\boldsymbol{C}}_{b-1}\}$ from historical data batches up to $b-1$.

3.2.3 Large Sample Properties

Let a neighborhood around true value $\boldsymbol{\beta}_0$ be $\mathbb{N}_\delta(\boldsymbol{\beta}_0) = \{\boldsymbol{\beta} : \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_2 \leq \delta\}$. We assume the following regularity conditions:

(C1) $\mathbb{E}_{\boldsymbol{\beta}}\{\boldsymbol{g}(\boldsymbol{y}; \boldsymbol{X}, \boldsymbol{\beta})\} = \mathbf{0}$ if and only if $\boldsymbol{\beta} = \boldsymbol{\beta}_0$.

(C2) The score vector $\boldsymbol{g}(\boldsymbol{y}; \boldsymbol{X}, \boldsymbol{\beta})$ is continuously differentiable, and its negative gradient $\boldsymbol{G}(\boldsymbol{y}; \boldsymbol{X}, \boldsymbol{\beta}) = -\partial \boldsymbol{g}(\boldsymbol{y}; \boldsymbol{X}, \boldsymbol{\beta})/\partial \boldsymbol{\beta}^T$ is Lipschitz continuous for $\boldsymbol{\beta} \in \Theta$.

(C3) The variability matrix $\mathbb{C}(\boldsymbol{\beta})$ is positive-definite for $\boldsymbol{\beta} \in \mathbb{N}_\delta(\boldsymbol{\beta}_0)$.

These conditions (C1)-(C3) are mild regularity conditions. Condition (C1) assumes the unbiasedness of the extended score estimating functions. Conditions (C2) and (C3) are required to establish estimation consistency and asymptotic normality.

Theorem III.1. *Under regularity conditions (C1)-(C3), the renewable estimator* $\tilde{\boldsymbol{\beta}}_b$ *given in* (3.6) *is consistent, namely* $\tilde{\boldsymbol{\beta}}_b \xrightarrow{p} \boldsymbol{\beta}_0$, *as* $N_b = \sum_{j=1}^{b} n_j \to \infty$.

Theorem III.1 presents the estimation consistency of renewable estimator $\tilde{\boldsymbol{\beta}}_b$ with respect to the cumulative sample size N_b.

Theorem III.2. *Under regularity conditions (C1)-(C3), the renewable estimator* $\tilde{\boldsymbol{\beta}}_b$ *is asymptotically normally distributed, that is,*

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

where Godambe information $\mathbb{J}(\boldsymbol{\beta}_0) = \mathbb{G}^T(\boldsymbol{\beta}_0)\mathbb{C}^{-1}(\boldsymbol{\beta}_0)\mathbb{G}(\boldsymbol{\beta}_0)$.

Note that in Theorem III.2, the convergence rate is $O_p(N_b^{-1/2})$, based on the cumulative sample size N_b. This indicates a faster convergence rate than parallelized distributed estimation where the convergence rate is based on the sample size of the smallest single dataset (*Zhou and Song*, 2017b) $\min_j \sqrt{n_j}$. The detailed proofs of both Theorems III.1 and III.2 are provided in Appendix B.2.

It is interesting to notice that the asymptotic covariance matrix of the renewable estimator $\tilde{\boldsymbol{\beta}}_b$ given in Theorem III.2 is exactly the same as that of the oracle estimator $\hat{\boldsymbol{\beta}}_b^\star$. This implies that the proposed renewable estimator achieves the same efficiency as the oracle QIF estimator. With no access to any historical subject-level data in the computation, using only the prior aggregated matrices $\tilde{\boldsymbol{G}}_b = \sum_{j=1}^b \boldsymbol{G}_j(\mathcal{D}_j; \tilde{\boldsymbol{\beta}}_j)$ and $\tilde{\boldsymbol{C}}_b = \sum_{j=1}^b \boldsymbol{C}_j(\mathcal{D}_j; \tilde{\boldsymbol{\beta}}_j)$, we can calculate the estimated asymptotic covariance matrix $\widetilde{\boldsymbol{\Sigma}_b}(\boldsymbol{\beta}_0)$ as $\widetilde{\boldsymbol{\Sigma}_b}(\boldsymbol{\beta}_0) = \left\{N_b^{-1}\tilde{\boldsymbol{J}}_b\right\}^{-1} = N_b\left\{\tilde{\boldsymbol{G}}_b^T\tilde{\boldsymbol{C}}_b^{-1}\tilde{\boldsymbol{G}}_b\right\}^{-1}$, where

$$\tilde{\boldsymbol{J}}_b = \left\{\sum_{j=1}^b \boldsymbol{G}_j(\mathcal{D}_j; \tilde{\boldsymbol{\beta}}_j)\right\}^T \left\{\sum_{j=1}^b \boldsymbol{C}_j(\mathcal{D}_j; \tilde{\boldsymbol{\beta}}_j)\right\}^{-1} \left\{\sum_{j=1}^b \boldsymbol{G}_j(\mathcal{D}_j; \tilde{\boldsymbol{\beta}}_j)\right\}.$$

It follows that the estimated asymptotic variance matrix for the renewable QIF $\tilde{\boldsymbol{\beta}}_b$ is

$$\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b) \coloneqq \widetilde{\mathrm{Var}}(\tilde{\boldsymbol{\beta}}_b) = \frac{1}{N_b}\widetilde{\boldsymbol{\Sigma}_b}(\boldsymbol{\beta}_0) = \left\{\tilde{\boldsymbol{G}}_b^T\tilde{\boldsymbol{C}}_b^{-1}\tilde{\boldsymbol{G}}_b\right\}^{-1}. \tag{3.8}$$

Theorem III.3 below presents the theoretical guarantee that the proposed renewable QIF estimator $\tilde{\boldsymbol{\beta}}_b$ is asymptotically equivalent to the oracle QIF estimator $\hat{\boldsymbol{\beta}}_b^\star$.

Theorem III.3. *Under conditions (C1)-(C3), the ℓ_2-norm difference between the oracle estimator $\hat{\boldsymbol{\beta}}_b^\star$ and the proposed renewable estimator $\tilde{\boldsymbol{\beta}}_b$ vanishes at the rate of N_b^{-1}, namely*

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

The proof of Theorem III.3 is included in Appendix B.3. When the size of cumulative dataset grows fast, numerically there is virtually no difference between $\tilde{\boldsymbol{\beta}}_b$ and $\tilde{\boldsymbol{\beta}}_b^\star$.

3.3 Detection of Abnormal Data Batches

For the case of high throughput data streams in practice, it is very likely to encounter abnormal data batches. To address this issue, we relax the renewable QIF method to a situation where abnormal data batches may occur over the course of data streams, $\mathcal{D}_2, \ldots, \mathcal{D}_b$. In this paper, a data batch $\mathcal{D}_\tau$, $\tau \in \{2, \ldots, b\}$, is regarded as being abnormal if $\mathcal{D}_\tau$ is generated from a model whose regression parameters, say $\boldsymbol{\beta}_\tau$, are different from those of the underlying main model of interest $\boldsymbol{\beta}_0$ (or the true model), i.e. $\boldsymbol{\beta}_\tau \neq \boldsymbol{\beta}_0$. In other words, $\mathcal{D}_\tau$ is an outlying data batch, which is incompatible with the data batches generated from the true model. Let $\Gamma_q = \{\tau_1, \ldots, \tau_q\}$ denote the set of indices for q abnormal data batches. In reality, we do not know set Γ_q in advance but want to find them out from streaming data. For convenience, we assume that the first data batch is generated from a model with $\boldsymbol{\beta}_0$. At each subsequent time point b ($b \geq 2$), we propose to test a hypothesis of mean-zero assumption for the pair of extended scores $H_0 : \mathbb{E}_{\boldsymbol{\beta}}(\boldsymbol{g}_{L_b}) = \mathbb{E}_{\boldsymbol{\beta}}(\boldsymbol{g}_b) = \mathbf{0}$, which essentially checks the compatibility between current data batch $\mathcal{D}_b$ under investigation and $\mathcal{D}_{L_b}$, where L_b denotes the latest data batch that is not rejected by the compatibility test. Clearly, $L_b = \max\{j : 2 \leq j < b, b \notin \Gamma_q\}$. If H_0 is rejected, we would not use current data batch $\mathcal{D}_b$ to renew $\tilde{\boldsymbol{\beta}}_{b-1}$ and set $\tilde{\boldsymbol{\beta}}_b$ equal to $\tilde{\boldsymbol{\beta}}_{b-1}$; otherwise, execute an update from $\tilde{\boldsymbol{\beta}}_{b-1}$ to $\tilde{\boldsymbol{\beta}}_b$. Then we proceed to test H_0 with next data batch $\mathcal{D}_{b+1}$, in which L_b is updated with the decision of the hypothesis test.

In the proposed test, we utilize two single data batches to monitor incidence of incompatibility, because we want to minimize the influence of potentially falsely selected or outdated historical data in the decision making (*Liu et al.*, 2017). Technically speaking, a test statistic constructed only with two data batches reduces greatly data storage demand and computational costs associated with extensive monitoring activities.

We construct a test statistic along the line of *Hansen* (1982)'s seminal goodness-

of-fit test. The quadratic inference function has useful chi-squared properties for hypothesis testing (*Lindsay and Qu*, 2003). In our setting of checking for data compatibility, we consider a quadratic inference function of the following form:

$$\Lambda_b(\boldsymbol{\beta}) = \begin{pmatrix} \boldsymbol{g}_{L_b}(\boldsymbol{\beta}) \\ \boldsymbol{g}_b(\boldsymbol{\beta}) \end{pmatrix}^T \begin{pmatrix} \boldsymbol{C}_{L_b}(\boldsymbol{\beta}) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{C}_b(\boldsymbol{\beta}) \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{g}_{L_b}(\boldsymbol{\beta}) \\ \boldsymbol{g}_b(\boldsymbol{\beta}) \end{pmatrix},$$

where $\boldsymbol{C}_{L_b}$ and $\boldsymbol{C}_b$ are the estimated sample covariances of extended scores $\boldsymbol{g}_{L_b}$ and $\boldsymbol{g}_b$, respectively, similar to the one given in (3.3). Note that the form of block covariance in Λ_b is due to the independence between D_{L_b} and D_b. Let $\check{\boldsymbol{\beta}}_b = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, \Lambda_b(\boldsymbol{\beta})$, under H_0, $\Lambda_b(\check{\boldsymbol{\beta}}_b) \xrightarrow{d} \chi^2_{df}$; under H_1, for an index $\tau \in \Gamma_q$ and a sequence of local alternatives in the form of $\boldsymbol{\beta}_\tau = \boldsymbol{\beta}_0 + (n_{L_b} + n_\tau)^{-1/2}\boldsymbol{d}$, where $\boldsymbol{d} \in \mathbb{R}^p$, $\Lambda_\tau(\boldsymbol{\beta}_\tau) \xrightarrow{d} \chi^2_{df}(\lambda)$ with $df = rank(\boldsymbol{C}_{L_b}) + rank(\boldsymbol{C}_{b(\tau)}) - p$, and non-centrality parameter $\lambda = \boldsymbol{d}^T\mathbb{J}(\boldsymbol{\beta}_0)\boldsymbol{d}$ with $\mathbb{J}$ being the Godambe information matrix in Theorem III.2. Moreover, it is easy to show that Power $= P_{H_1}\left(\Lambda_\tau(\check{\boldsymbol{\beta}}_\tau) > \chi^2_{df,\alpha}\right) \to 1$, as $(n_{L_b} + n_\tau) \to \infty$, which implies that the proposed test Λ_τ is consistent. Under a finite sample size, with fixed $\boldsymbol{d}$, the power of Λ_τ depends on both statistical significance level α and abnormal data batch size n_τ. Larger α leads to higher power and smaller type II error, but also a higher chance to produce false alarms; obviously, increasing data batch size n_b will help increase power.

Even though the above monitoring test is carried out sequentially and repeatedly, it is in fact different from the conventional sequential test (*Wald*, 1945). This is because in the proposed monitoring scheme, we do not stop the test even if we find an abnormal data batch. In other words, the previous decision does not affect a current investigation, nor future ones. Every monitoring task takes place at a given time b, which is an isolated circumstance, instead of a collective decision making scheme.

3.4 Implementation

We expand the existing Spark's Lambda architecture to reduce computing burden in the framework of renewable QIF methodology. The iterative calculation in (3.7) can be implemented in the speed and inference layers in an extended Lambda architecture shown in Figure 3.1. Here, relevant inferential statistics include the aggregated extended score vector $\tilde{\boldsymbol{g}}$ and two information matrices $\tilde{\boldsymbol{G}}$ (aggregated negative gradient) and $\tilde{\boldsymbol{C}}$ (aggregated sample variance matrix). If H_0 is not rejected, store $\mathcal{D}_b$ in the monitoring layer and update eligibility index as $L_b = b$, followed by updating $\tilde{\boldsymbol{\beta}}_{b-1}$ to $\tilde{\boldsymbol{\beta}}_b$ at the speed layer and updating $\tilde{\boldsymbol{g}}_{b-1}$, $\tilde{\boldsymbol{G}}_{b-1}$, $\tilde{\boldsymbol{C}}_{b-1}$ to $\tilde{\boldsymbol{g}}_b$, $\tilde{\boldsymbol{G}}_b$ and $\tilde{\boldsymbol{C}}_b$ at the inference layer. Otherwise, skip all updating steps and proceed to next data batch $\mathcal{D}_{b+1}$.

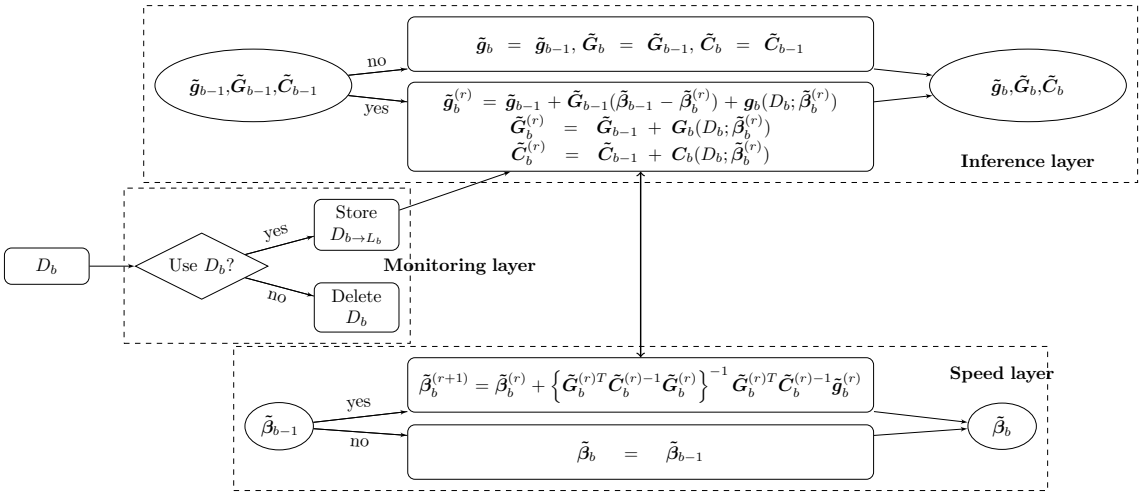

Figure 3.1: Diagram of an extended Lambda architecture.

Algorithm 3.2 lists the pseudo code for the implementation of the renewable QIF via the paradigm of the extended Lambda architecture shown in Figure 3.1. Some explanations are given below.

1. Line 1: the marginal GLM considered in this paper belongs to the family of exponential dispersion (ED) models (*Jørgensen*, 1997) and most streaming datasets

are supposed to be governed marginally by the main underlying ED model with true parameter $\boldsymbol{\beta}_0$, but also with possible abnormal data batches that are generated with $\boldsymbol{\beta} \neq \boldsymbol{\beta}_0$. The ED models automatically satisfy all regularity conditions given in Section 3.2.3.

2. Line 2: the outputs include renewable QIF estimates of the regression coefficients and the corresponding estimated asymptotic variance matrix at each time point b, and the latter is needed for statistical inference.

3. Line 3: set certain initial values for the regression coefficients, *e.g.*, set $\tilde{\boldsymbol{\beta}}_0$ sd the QIF estimates from fitting $\mathcal{D}_1$ to R function `glm()`.

4. Line 4: run through the sequential updating procedure along data streams. 5. Line 6: before renew QIF with current $\mathcal{D}_b$, first check its compatibility with $\mathcal{D}_{L_b}$, the latest data batch that passes the monitoring test. QIF estimator $\check{\boldsymbol{\beta}}_b$ is obtained by minimizing the quadratic inference function based only on the two adjacent data batches, $\mathcal{D}_{L_b} \cup \mathcal{D}_b$.

5. Line 7: if the test is rejected, we set $\tilde{\boldsymbol{\beta}}_b = \tilde{\boldsymbol{\beta}}_{b-1}$, $\tilde{\boldsymbol{g}}_b = \tilde{\boldsymbol{g}}_{b-1}$, $\tilde{\boldsymbol{G}}_b = \tilde{\boldsymbol{G}}_{b-1}$ and $\tilde{\boldsymbol{C}}_b = \tilde{\boldsymbol{C}}_{b-1}$ and jump to Line 15 directly.

6. Line 9: if the test concludes no rejection, at the inference layer, utilize the prior QIF estimate $\tilde{\boldsymbol{\beta}}_{b-1}$ and current data batch $\mathcal{D}_b$ to calculate the aggregated extended score vector $\tilde{\boldsymbol{g}}_b$, the aggregated negative gradient $\tilde{\boldsymbol{G}}_b$ and sample variance matrix $\tilde{\boldsymbol{C}}_b$. Through a communication with the speed layer, update these three summary statistics in the inference layer.

7. Line 11: run the Newton-Raphson algorithm to renew $\tilde{\boldsymbol{\beta}}_{b-1}$ to $\tilde{\boldsymbol{\beta}}_b$.

8. Line 15: at the inference layer, use inferential statistics $\tilde{\boldsymbol{G}}_b$ and $\tilde{\boldsymbol{C}}_b$ to perform statistical inference.

3.5 Simulation Experiments

3.5.1 Setup

We conduct simulation experiments to assess the performances of the proposed renewable QIF estimation and inference, as well as of the monitoring scheme for abnormal data batches, in the setting of marginal generalized linear models (MGLMs). We compare the renewable QIF method (RenewQIF) with (i) the oracle GEE estimator obtained by processing the entire cumulative data once, (ii) the oracle QIF estimator obtained by processing the entire data once, and (iii) renewable GEE estimation method (RenewGEE) that is similar to RenewQIF (see the relevant derivation in Appendix B.1).

In the first part of comparisons to be presented below, we consider the following criteria related to both parameter estimation and inference: (a) averaged absolute bias (A.bias), (b) averaged asymptotic standard error (ASE), (c) empirical standard error (ESE), and (d) coverage probability (CP). Both oracle GEE and QIF estimates are yielded from the R packages `gee` and `qif`. Computational efficiency is assessed by (e) computation time (C.Time) and (f) running time (R.Time). R.Time accounts only algorithm execution time, while C.Time includes time spent on both data loading and algorithm execution. In the second part of comparisons, we will first evaluate the type I error and power of the proposed goodness-of-fit test with different significance level α and data batch size n_b. Then the criteria for parameter estimation and inference will be compared thoroughly on methods with and without quality control.

In the simulation studies, we set a terminal point B, and generate a full dataset $\mathcal{D}_B^\star$ with N_B independent cluster-correlated observations of m dimensions from the respective MGLMs, consisting of the mean model $\mathbb{E}(\boldsymbol{y}_i \mid \boldsymbol{X}_i) = \big[h(\boldsymbol{x}_{i1}^T\boldsymbol{\beta}_0), \ldots, h(\boldsymbol{x}_{im}^T\boldsymbol{\beta}_0)\big]^T$ with $\boldsymbol{\beta}_0 = (0.2, -0.2, 0.2, -0.2, 0.2)^T$, and covariance matrix $\mathrm{Cov}(\boldsymbol{y}_i \mid \boldsymbol{X}_i) = \phi\boldsymbol{\Sigma}$, $i = 1, \ldots, N_B$, where four covariates $\boldsymbol{x}_{ij[2:5]} \overset{iid}{\sim} \mathcal{N}_4(\boldsymbol{0}, \boldsymbol{V}_x)$ and intercept $\boldsymbol{x}_{ij[1]} = 1$,

$j = 1, \ldots, m$. Here both covariance matrices $\boldsymbol{V}_x$ and $\boldsymbol{\Sigma}$ are set as compound symmetry with $\rho_x = 0.5$ and $\rho_y = 0.7$, respectively. The dispersion parameter $\phi = 1$ and the cluster size is $m = 5$. We consider both marginal linear model for continuous y_{ij} with $h(\mu_{ij}) = \mu_{ij}$ and marginal logistic model for binary y_{ij} with $h(\mu_{ij}) = \log(\mu_{ij}/(1-\mu_{ij}))$. For all four methods in comparison, the working correlation matrix is specified to be compound symmetry.

3.5.2 Evaluation of Parameter Estimation

Scenario 1: fixed N_B but varying batch size n_b

We begin with the comparison of four methods for the effect of data batch size n_b on their performance of point estimation and computational efficiency. There are B data streams, each with data batch size n_b, and the total sample size $N_B = |D_B^\star| = 10^5$ independent clusters, which are simulated, respectively, from an m-variable Gaussian linear model and an m-dimensional logistic model (using R package `SimCorMultRes`) specified in Section 3.5.1. Tables 3.1 reports the results of both linear and logistic MGLMs, over 500 rounds of simulations.

Bias and coverage probability. In linear and logistic MGLMs, both RenewGEE and RenewQIF provide similar bias and coverage probability as the two oracle methods, shown in Table 3.1, which confirms the theoretical results given in Theorems III.1 and III.2. It is easy to see that both bias and coverage probability in both the linear and logistic models are not affected by individual data batch size n_b. In other words, it depends only on N_B.

Computation time. Two metrics are used to evaluate computation efficiency: "C.Time" in Table 3.1 refers to the total amount of time required by data loading and algorithm execution. With an increased B, both RenewGEE and RenewQIF show clearly advantageous for much lower computation time over the oracle GEE and QIF, due to the fact that it is much more time consuming for the oracle methods to load in full

datasets.

Scenario 2: fixed batch size n_b but varying B

Now we turn to a streaming setting where B data batches arrive sequentially. For convenience, we fix single batch size $n_b = 100$, but let N_B increase from 10^3 to 10^6 (or B from 10 to 10^4). Tables 3.2 and 3.3 list the summaries of simulation results under the linear and logistic MGLMs specified in Section 3.5.1.

Bias and coverage probability. As number of data batches B increases from 10 to 10^4, both RenewGEE and RenewQIF confirm the large sample properties in Theorems III.1 and III.2 similar to those of the oracle GEE and QIF: the average absolute bias decreases rapidly as the total sample size accumulates, and the coverage probability stays robustly around the nominal level 95%.

Computation time. Both RenewGEE and RenewQIF methods show more and more advantageous as N_B increases: the combined amount of time for data loading and algorithm execution only takes less than 5 seconds, whereas the oracle GEE and QIF, when processing a total of 10^5 samples once, requires more than 20 seconds. This 5-fold faster computation by the proposed RenewGEE and RenewQIF methods costs little price of estimation precision, and retains the same inferential power. One thing worth mentioning for Table 3.3 is that when $N_B = 10^6$, in the logistic MGLM, the oracle GEE is computationally too intensive to produce results within 12 hours using the standard R package `gee`.

3.5.3 Evaluation of Monitoring Procedure

We also evaluate the performance of the proposed monitoring procedure using the chi-squared goodness-of-fit test Λ_b to detect abnormal data batches. First, we confirm the properties of the test statistic with respect to both type I error and power of detection abnormal data batches. Then, we compare the estimation and inference

Table 3.1: Simulation results under the linear and logistic MGLMs are summarized over 500 replications, with fixed $N_B = 10^5$ and $p = 5$ with increasing number of data batches B. "A.bias", "ASE", "ESE" and "CP" stand for the mean absolute bias, the mean asymptotic standard error of the estimates, the empirical standard error, and the coverage probability, respectively. "C.Time" and "R.Time" respectively denote computation time and running time, and the unit of both is second.

Linear MGLM												
	Oracle GEE			RenewGEE			Oracle QIF			RenewQIF		
B	100	500	2000	100	500	2000	100	500	2000	100	500	2000
A.bias$\times 10^{-3}$	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10
ASE$\times 10^{-3}$	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42
ESE$\times 10^{-3}$	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40
CP	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
C.Time(s)	9.27	12.89	20.56	2.70	4.08	8.45	2.60	5.36	13.91	1.39	2.77	6.63
R.Time(s)	8.53	9.49	8.53	2.31	2.64	3.62	1.86	1.96	1.88	1.06	1.65	2.95
Logistic MGLM												
	Oracle GEE			RenewGEE			Oracle QIF			RenewQIF		
B	100	500	2000	100	500	2000	100	500	2000	100	500	2000
A.bias$\times 10^{-3}$	2.70	2.70	2.70	2.70	2.70	2.70	2.70	2.70	2.70	2.70	2.70	2.70
ASE$\times 10^{-3}$	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31	3.31
ESE$\times 10^{-3}$	3.37	3.37	3.37	3.37	3.37	3.37	3.37	3.37	3.37	3.37	3.37	3.37
CP	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94
C.Time(s)	10.73	14.04	27.77	2.05	2.45	3.32	3.22	6.51	20.25	1.14	1.47	2.37
R.Time(s)	9.85	9.86	9.81	1.86	2.07	2.35	2.34	2.33	2.30	0.99	1.23	1.86

Table 3.2: Compare renewable estimators and oracle ones in the linear MGLM model with fixed single batch size $n_b = 100$ and $p = 5$, B increases from 10 to 10^4. Results are summarized from 500 replications.

	$B = 10$, $N_B = 10^3$				$B = 100$, $N_B = 10^4$			
	GEE		QIF		GEE		QIF	
Criterion	Oracle	Renew	Oracle	Renew	Oracle	Renew	Oracle	Renew
A.bias$\times 10^{-3}$	11.06	11.06	11.08	11.08	3.64	3.64	3.64	3.64
ASE$\times 10^{-3}$	14.19	14.16	14.15	14.13	4.49	4.49	4.49	4.49
ESE$\times 10^{-3}$	13.82	13.83	13.85	13.85	4.51	4.51	4.51	4.51
CP	0.956	0.955	0.952	0.952	0.949	0.947	0.946	0.946
C.Time(s)	0.033	0.028	0.019	0.023	0.69	0.25	0.28	0.18
R.Time(s)	0.030	0.024	0.015	0.019	0.58	0.25	0.16	0.14
	$B = 10^3$, $N_B = 10^5$				$B = 10^4$, $N_B = 10^6$			
	GEE		QIF		GEE		QIF	
Criterion	Oracle	Renew	Oracle	Renew	Oracle	Renew	Oracle	Renew
A.bias$\times 10^{-3}$	1.11	1.11	1.11	1.11	0.35	0.35	0.35	0.35
ASE$\times 10^{-3}$	1.42	1.42	1.42	1.42	0.45	0.45	0.45	0.45
ESE$\times 10^{-3}$	1.40	1.40	1.40	1.40	0.44	0.44	0.44	0.44
CP	0.952	0.954	0.952	0.952	0.955	0.955	0.955	0.955
C.Time(s)	15.38	5.70	8.57	4.26	781.46	62.30	704.11	51.38
R.Time(s)	8.72	2.96	1.90	2.18	99.21	32.12	21.85	25.21

Table 3.3: Compare renewable estimators and oracle ones in the logistic MGLM model with fixed single batch size $n_b = 100$ and $p = 5$, B increases from 10 to 10^4. Results are summarized from 500 replications. The dashed line in the column for "Oracle GEE" when $N_B = 10^6$ indicates the standard `gee` package in R does not produce output due to the excessive computational burden.

	$B = 10$, $N_B = 10^3$				$B = 100$, $N_B = 10^4$			
	GEE		QIF		GEE		QIF	
Criterion	Oracle	Renew	Oracle	Renew	Oracle	Renew	Oracle	Renew
A.bias$\times 10^{-3}$	25.92	25.82	26.07	26.01	8.17	8.16	8.17	8.16
ASE$\times 10^{-3}$	33.08	33.06	33.03	33.07	10.45	10.45	10.45	10.45
ESE$\times 10^{-3}$	32.48	32.36	32.67	32.60	10.31	10.30	10.32	10.31
CP	0.953	0.952	0.950	0.952	0.950	0.952	0.951	0.952
C.Time(s)	0.048	0.029	0.024	0.023	1.09	0.23	0.32	0.17
R.Time(s)	0.045	0.026	0.021	0.020	0.99	0.20	0.22	0.14
	$B = 10^3$, $N_B = 10^5$				$B = 10^4$, $N_B = 10^6$			
	GEE		QIF		GEE		QIF	
Criterion	Oracle	Renew	Oracle	Renew	Oracle	Renew	Oracle	Renew
A.bias$\times 10^{-3}$	2.71	2.71	2.71	2.71	-	0.82	0.82	0.82
ASE$\times 10^{-3}$	3.31	3.31	3.31	3.31	-	1.05	1.05	1.05
ESE$\times 10^{-3}$	3.39	3.39	3.39	3.39	-	1.04	1.04	1.04
CP	0.948	0.948	0.948	0.948	-	0.946	0.946	0.948
C.Time(s)	22.41	5.84	9.99	4.49	-	57.47	856.70	47.44
R.Time(s)	15.59	3.02	3.18	2.35	-	31.08	45.83	21.31

performance of the RenewQIF methods with and without monitoring procedure on the criteria including (a) A.bias, (b) ASE, (c) ESE, and (d) CP. The abnormal data batches are created by altering the true parameters via a local derivation on β_{02}, that is $\boldsymbol{\beta}_\tau = (0.2, -(0.2+d), 0.2, -0.2, 0.2)^T$, $\tau \in \Gamma_q$. We set Γ_2, containing two positions ($q = 2$) for two occurrences of abnormal data batches at $\tau_1 = 0.25B$ and $\tau_2 = 0.75B$. Table 3.4 shows the empirical type I error rates for different batch sizes n_b under various significance level $\alpha \in \{0.1, 0.05, 0.01, 0.001, 5 \times 10^{-6}\}$. They are all very close to the nominal level α. These findings confirm the theoretical insights for a fixed local alternative with departure size d. Table 3.4 also shows that the power of detection abnormal data batches drops as α becomes smaller, while the power increases with increasing n_b.

Without monitoring procedure. With a fixed $N_B = 10^4$ and $\Gamma_2 = \{0.25B, 0.75B\}$, the upper panel in Table 3.5 shows that larger data batch size n_b leads to a larger

Table 3.4: Empirical type I error rate ($\times 10^{-3}$) under a total number of $B = 100$ data batches with different data batch size n_b and various significance level α. In the calculation of empirical power, the locations of two contaminated data batches are $\tau_1 = 25$ and $\tau_2 = 75$. Results are summarized over 500 replications.

	Empirical type I error rate ($\times 10^{-3}$)				Empirical power (%)							
	$d = 0$				$d = 0.5$				$d = 1.0$			
	n_b				n_b				n_b			
$\alpha \times 10^{-3}$	50	100	200	400	50	100	200	400	50	100	200	400
100	95	91	93	89	100	100	100	100	100	100	100	100
50	41	42	44	46	100	100	100	100	100	100	100	100
10	5	8	9	8	58	98.5	100	100	100	100	100	100
1	0.2	0.5	0.4	1	21.5	90	100	100	99.5	100	100	100
0.005	0	0	0	0	0	42.5	100	100	70.5	100	100	100

bias due to the increased number of contaminated observations generated from the incompatible data model. A.bias increases almost linearly with n_b. Consequently, with similar levels of ASE and ESE, the coverage probability is departed more from the nominal level 95% as n_b increases; it drops from 78.5% to 0% as n_b rises from 50 to 200 due to more severe data contamination.

With monitoring procedure. For the purpose of quality control, larger α increases the sensitivity of rejection, so many small departures may be detected, which would be consequently ignored in the online updating. The price to pay in this case is that the resulting bias and standard error are larger than they would be if such false positives can be avoided. Nevertheless, it is worth noting that the coverage probability stays around 95% with, however, potentially wider confidence intervals. These are clearly shown in the lower panel of Table 3.5 due to the reduced proportion of used samples, defined by N_0/N_B (see also the last subplot in Figure 3.3). In contrast, choosing small α may elevate type II error and thus lose power in detecting abnormal data batches. In this case, the price to pay is not only increased bias but also decreased coverage probability. The latter is indeed a more serious problem as far as inference concerns. This phenomenon is evident when n_b is small as shown in both Figure 3.3 and Table 3.5. As an extreme case of $\alpha = 5 \times 10^{-6}$, the detection power is greatly lost

with $n_b = 100$ or 50, and the coverage probability reduces to 90%. In practice, with high throughput data streams, where cumulative sample sizes increase rapidly, using larger α, say $\alpha = 0.1$, is much safer and recommended in practice for the monitoring, resulting in a more protective process by effectively avoiding abnormal data batches.

Table 3.5: Performances with and without monitoring procedure. Fixed total number of samples $N_B = 10^4$ with varying data batch size n_b. $\tau_1 = 0.25B$ and $\tau_2 = 0.75B$. In the table "With monitoring procedure", N_0/N_B denotes the proportion of used samples in the renewable estimation and inference.

Without monitoring procedure				
n_b	50	100	200	400
A.bias$\times 10^{-3}$	7.469	14.90	29.74	59.66
ASE$\times 10^{-3}$	5.591	5.656	5.755	5.925
ESE$\times 10^{-3}$	5.721	5.225	5.540	5.389
CP	0.785	0.280	0.000	0.000

With monitoring procedure										
	$n_b = 50$					$n_b = 100$				
$\alpha \times 10^{-3}$	100	50	10	1	0.005	100	50	10	1	0.005
A.bias$\times 10^{-3}$	**4.893**	4.841	4.697	4.805	**5.568**	**4.601**	4.395	4.307	4.359	**5.044**
ASE$\times 10^{-3}$	5.850	5.679	5.573	5.559	5.568	5.871	5.720	5.613	5.596	5.596
ESE$\times 10^{-3}$	6.067	6.074	5.865	5.823	6.134	5.652	5.432	5.298	5.332	5.736
CP	0.955	0.920	0.930	0.935	**0.900**	0.970	0.980	0.955	0.960	**0.900**
N_0/N_B	0.894	0.948	0.986	0.993	0.995	0.890	0.937	0.973	0.979	0.983
	$n_b = 200$					$n_b = 400$				
$\alpha \times 10^{-3}$	100	50	10	1	0.005	100	50	10	1	0.005
A.bias$\times 10^{-3}$	**4.701**	4.479	4.377	4.417	4.468	**5.045**	4.799	4.579	4.454	4.457
ASE$\times 10^{-3}$	5.946	5.779	5.673	5.648	5.647	6.084	5.921	5.798	5.773	5.766
ESE$\times 10^{-3}$	5.813	5.623	5.488	5.556	5.599	6.221	5.888	5.666	5.525	5.516
CP	0.945	0.960	0.960	0.970	0.965	0.935	0.950	0.950	0.955	0.955
N_0/N_B	0.868	0.917	0.951	0.960	0.960	0.832	0.875	0.911	0.918	0.920

Algorithm 2: Renewable QIF in the MGLMs for streaming cluster-correlated data in the extended Lambda architecture.

Inputs: Marginal model ED($\boldsymbol{\mu}, \phi\boldsymbol{\Sigma}$); sequentially arrived datasets $\mathcal{D}_1$,...,$\mathcal{D}_b$,... ;

Outputs: $\tilde{\boldsymbol{\beta}}_b$ and $\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b)$, for $b = 1, 2, \ldots$;

Initialize: Initial values $\tilde{\boldsymbol{\beta}}_0$, $\tilde{\boldsymbol{g}}_0 = \mathbf{0}_{pS}$, $\tilde{\boldsymbol{G}}_0 = \mathbf{0}_{pS\times p}$ and $\tilde{\boldsymbol{C}}_0 = \mathbf{0}_{pS\times pS}$;

for $b = 1, 2, \ldots$ **do**

  Read in dataset $\mathcal{D}_b$;

  at the monitoring layer, if $b \geq 2$, calculate $\Lambda_b = Q_{L_b}(\check{\boldsymbol{\beta}}_b) + Q_b(\check{\boldsymbol{\beta}}_b)$ and reject H_0 if $\Lambda_b \geq \chi^2_{df,\alpha}$, say $\alpha = 0.05$.

  If H_0 is rejected, set $\tilde{\boldsymbol{\beta}}_b = \tilde{\boldsymbol{\beta}}_{b-1}$, $\tilde{\boldsymbol{g}}_b = \tilde{\boldsymbol{g}}_{b-1}$, $\tilde{\boldsymbol{G}}_b = \tilde{\boldsymbol{G}}_{b-1}$, $\tilde{\boldsymbol{C}}_b = \tilde{\boldsymbol{C}}_{b-1}$ and jump to Line 15;

  if H_0 is not rejected, store $\mathcal{D}_b$ in the monitoring layer, and start iterations with $\tilde{\boldsymbol{\beta}}_b$ initialized by $\tilde{\boldsymbol{\beta}}_{b-1}$,

  repeat

    at the inference layer, calculate $\tilde{\boldsymbol{g}}_b^{(r)} = \tilde{\boldsymbol{g}}_{b-1} + \tilde{\boldsymbol{G}}_{b-1}(\tilde{\boldsymbol{\beta}}_{b-1} - \tilde{\boldsymbol{\beta}}_b) + \boldsymbol{g}_b(\mathcal{D}_b; \tilde{\boldsymbol{\beta}}_b^{(r)})$,

    $\tilde{\boldsymbol{G}}_b^{(r)} = \tilde{\boldsymbol{G}}_{b-1} + \boldsymbol{G}_b(\mathcal{D}_b; \tilde{\boldsymbol{\beta}}_b^{(r)})$ and $\tilde{\boldsymbol{C}}_b^{(r)} = \tilde{\boldsymbol{C}}_{b-1} + \boldsymbol{C}_b(\mathcal{D}_b; \tilde{\boldsymbol{\beta}}_b^{(r)})$;

    at the speed layer, $\tilde{\boldsymbol{\beta}}_b^{(r+1)} = \tilde{\boldsymbol{\beta}}_b^{(r)} + \left\{\tilde{\boldsymbol{G}}_b^{(r)^T}\tilde{\boldsymbol{C}}_b^{(r)^{-1}}\tilde{\boldsymbol{G}}_b^{(r)}\right\}^{-1}\tilde{\boldsymbol{G}}_b^{(r)^T}\tilde{\boldsymbol{C}}_b^{(r)^{-1}}\tilde{\boldsymbol{g}}_b^{(r)}$;

  until *converegence*;

  At the inference layer, calculate $\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b) = \left\{\tilde{\boldsymbol{G}}_b^T\tilde{\boldsymbol{C}}_b^{-1}\tilde{\boldsymbol{G}}_b\right\}^{-1}$;

  Save $\tilde{\boldsymbol{\beta}}_b$, $\tilde{\boldsymbol{g}}_b$, $\tilde{\boldsymbol{G}}_b$ and $\tilde{\boldsymbol{C}}_b$ at the speed and inference layers, respectively;

  Release dataset $\mathcal{D}_b$ from the memory.

end

Return $\tilde{\boldsymbol{\beta}}_b$ and $\tilde{\boldsymbol{V}}(\tilde{\boldsymbol{\beta}}_b)$, for $b = 1, 2, \ldots$.

Figure 3.2: Pseudo code for the implementation of renewable QIF.

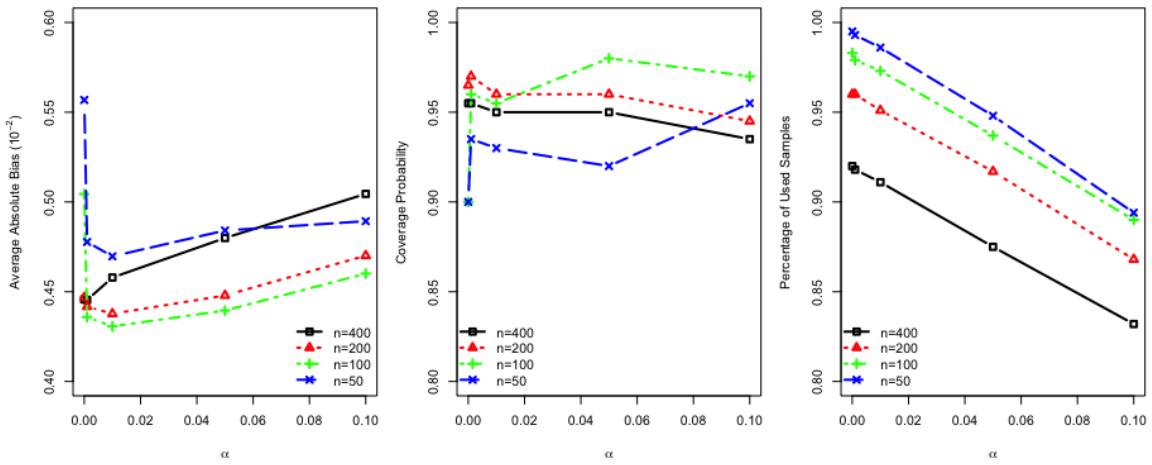

Figure 3.3: For fixed $N_B = 10^4$, the relationship between type I error, estimation bias, coverage probability and percentage of used data.

3.6 Analysis of NASS CDS Data

In regard to injuries involved in car accidents, we are interested in not only the extent of injuries for drivers but also for passengers. Apparently, injury levels of driver and passengers within the same vehicle are correlated, and such within-cluster correlation needs to be taken into account in the analysis. In this real data application, we focus on the analysis of a series of car crash datasets from the National Automotive Sampling System-Crashworthiness Data System (NASS CDS) from January, 2009 to December, 2015. Our primary interest was to evaluate the effectiveness of graduated driver licensing (GDL) on overall driving safety with respect to injury levels in both driver and passengers. GDL is a nationwide legislature on novice drivers of age 21 or younger with various conditions of vehicle operation. In contrast, under the current law, there are no restrictions on vehicle operation for older drivers with age, say older than 65. Thus, we wanted to compare drivers' age groups with respect to the extent of injury when a car accident happened. We first categorized the "Age" variable into three age groups: "Age<21" representing the young group under a restricted GDL, and "Age$\geq$65" for the old group with a regular full driver's license, while those of age in between was treated as the reference group. Extent of "Injury" in a crash is a binary variable, 1 for a moderate or severe injury, and 0 for minor or no injury. This outcome variable was created from the variable of Maximum Known Occupant Ais (MAIS), which indicates the single most severe injury level reported for each occupant. Other potential risk factors were also included in the model, including seat belt use (Seat Belt, 1 for used and 0 for no), drinking (Drinking, 1 for yes and 0 for no), speed limit (Speed Limit), vehicle weight (Vehicle Weight, 0 for $\leq$ 3000, 1 for 3000$\sim$4000, 2 for $\geq$4000), air bag system deployed (Air Bag, 1 for yes and 0 for no), number of lanes (Number of Lanes, 0 for $\leq$ 2 and 1 for else), drug involvement in this accident (Drug Use, 1 for yes and 0 for no), driver's distraction/inattention to driving (Distraction, 1 for attentive and 0 for else), roadway surface condition (Surface Condition, 1 for

dry and 0 for else), and has vehicle been in previous accidents (Previous Accidents, 0 for no and 1 for else).

Streaming data were formed by quarterly accident data from the period of 7 years from January, 2009 to December 2015, with $B = 28$ data batches and a total of $N_B = 18,832$ crashed vehicles that contain $26,330$ occupants with complete records. Each vehicle was treated as a cluster, and the cluster size varies from 1 to 10 with an average of 2 occupants. We invoked RenewQIF to fit marginal logistic regression model with the compound symmetry correlation to account for the within vehicle correlation. In the analysis, we are interested in a 7-year average risk assessment and thus assuming constant associations between extent of injuries and risk factors over time. Since this sequence of data streams arrived with low speed and large data batch size, we would expect to have high power to detect the abnormal data batch even if we chose a small α, say $\alpha = 0.01$. Additionally, samples from NASS CDS have gone through extensive data cleaning and pre-processing steps, such a stringent α was a reasonable choice to make full use of samples. At $\alpha = 0.01$, our proposed monitoring procedure identified data batch 8 as the incompatible one, corresponding to the 4th quarter in year 2010. Table 3.6 reports estimated coefficients, standard errors and p-values obtained by oracle QIF, RenewQIF with and without data batch 8.

Figure 3.4 shows the trajectories of $-\log_{10}(p)$ values of the Wald test in the 10-base logarithm, each for one regression coefficient over 28 quarters. Even though the total sample size N_B increases over time, not all of them show steep monotonic increasing trends in evidence against the null $H_0 : \beta_j = 0$. "Seat Belt" turns out to have the strongest association to the odds of injury in a crash among all covariates included in the model. This is an overwhelming confirmation to the enforcement of policy "buckle up" when sitting in a moving vehicle. For the convenience of comparison, we reported a summary statistic as of the area under the p-value curve for each covariate. "Seat Belt" (2297.45), "Drug Use" (1779.85), "Air Bag" (1249.49) and

"Previous Accidents" (1342.46) appeared well separated from the other risk factors. Their ranking is well aligned with the ranking of p-values obtained at the end time of streaming data availability, namely December, 2015.

The trajectories of both young and old age groups were evaluated in Figure 3.5 with or without data batch 8. The trace of the estimators for the young age group (Age<21) stays below 0 over the 28-quarter period, indicating that it has lower adjusted odds of moderate/severe injury than the reference group. This finding confirms the effectiveness of GDL in protecting young novice drivers. Unfortunately, in contrast, the old age group (Age≥65) turns out to suffer from significantly higher adjusted odds of moderate/severe outcomes comparing to the middle age group. This suggests a need of policy-making to protect older drivers from injuries when an accident happens. Furthermore, the abnormal data batch seems to affect marginally the estimates for age groups if we compare the plots with (right) and without (left) monitoring procedure (see the red vertical dashed line).

Applying the proposed RenewQIF to the above CDS data analysis enabled us to visualize time-course patterns of data evidence accrual as well as stability and reproducibility of inference. As shown in Figure 3.5, at the early stage of data streams, due to limited sample sizes and possibly sampling bias, both parameter estimates and test power may be unstable and even possibly misleading. These potential shortcomings can be overcome when estimates and inferential quantities were continuously updated along with data streams, which eventually reached stability and reliable conclusions.

3.7 Concluding Remarks

Due to the advantage of technologies in data storage and data collection, streaming data arise from many practical areas such as healthcare (*Peek et al.*, 2014). Healthcare data are typically measurements in forms of clusters or time series, such as patients from the same clinic, or data from personal wearable devices (*Sahoo et al.*, 2016).

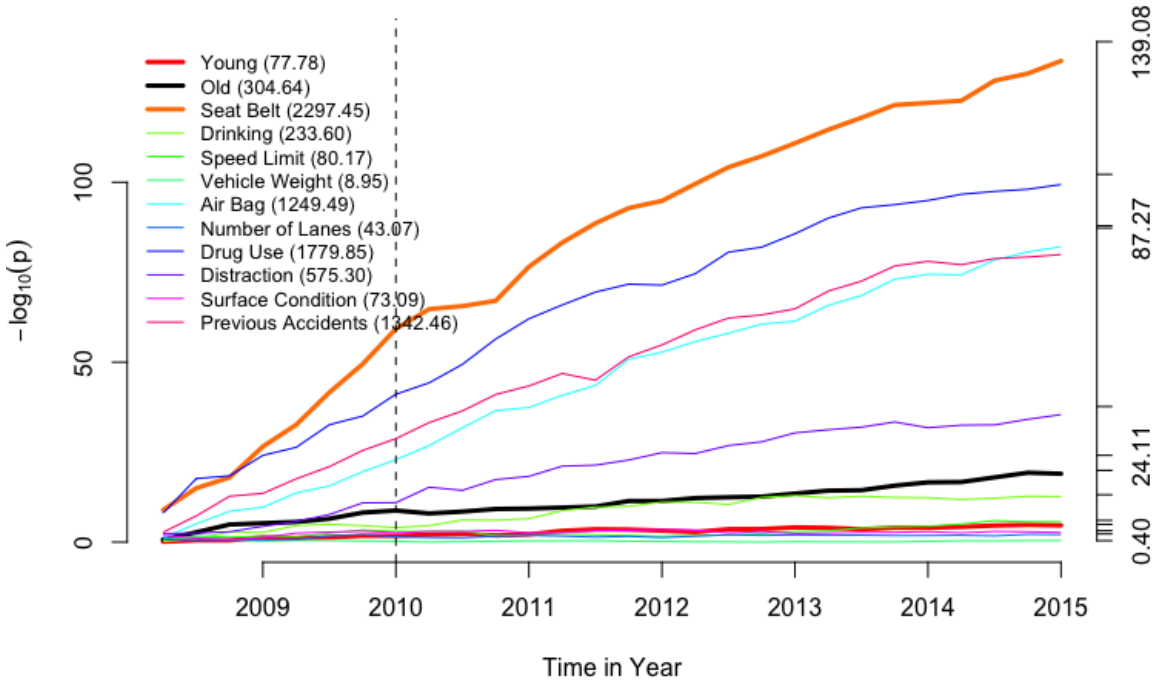

Figure 3.4: Trace plots of $-\log_{10}(p)$ over quarterly data batches from January, 2009 to December, 2015, each for one regression coefficient. Dashed vertical line indicates the location of detected abnormal data batch.

The traditional methods for clustered/longitudinal data analysis such as generalized estimating equations (GEE) and quadratic inference functions (QIF) that process the entire dataset once may be greatly challenged due to the following reasons: (i) they become computationally prohibitive as the total sample size accumulates too fast and too large, so to exceed the available computational power; see Table 3.3 where GEE failed to produce output; and (ii) historical subject-level data may no longer be accessible due to storage, time, or privacy issues. This type of problem has been extensively tackled in the framework of online updating where stochastic gradient descent algorithms are the primary methods of choice to provide fast updating with no use of historical data. However, most online learning algorithms have not considered

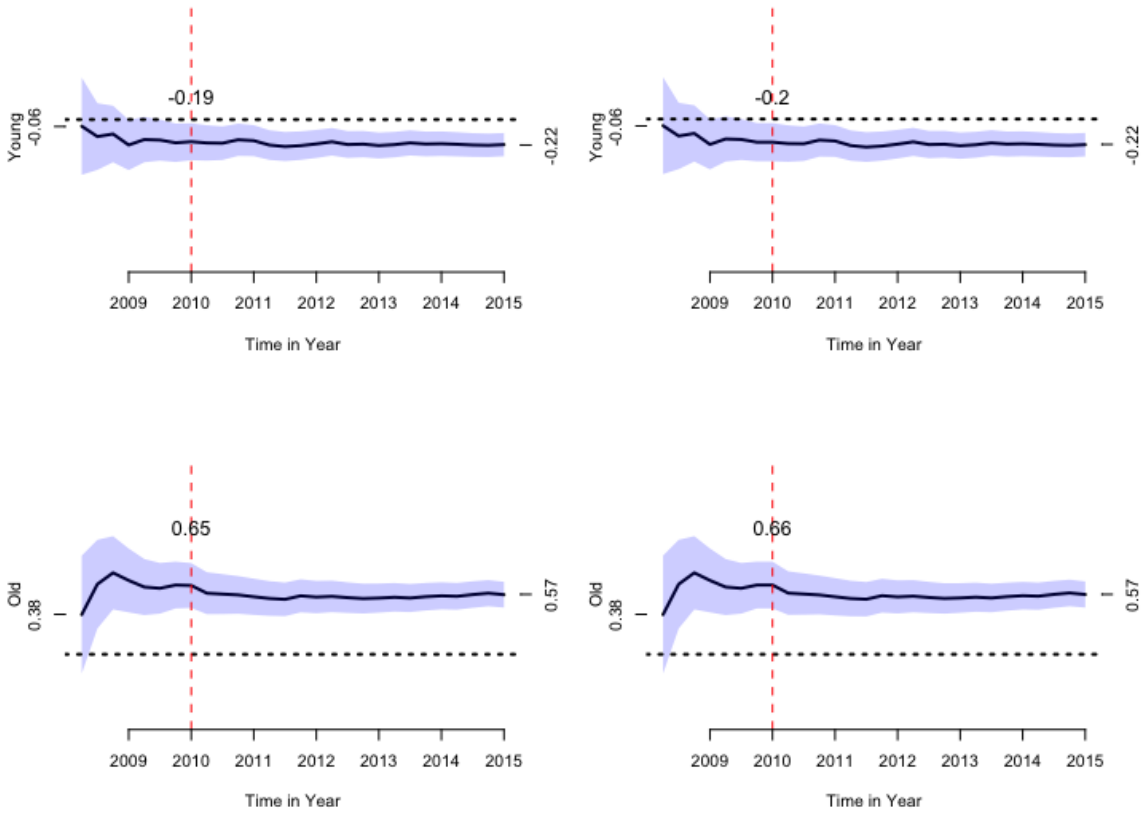

Figure 3.5: Trace plots for the coefficients estimates and 95% pointwise confidence bands of "Young" and "Old". The subplot on the left corresponds to RenewQIF without monitoring procedure, and the one on the right is obtained by excluding data batch 8. Numerical numbers on two sides denote the estimated regression coefficients after the arrival of first and last batches, while the ones above the traces denote the estimates at the 8th data batch.

Table 3.6: Results from the oracle QIF method ($N_B = 18,832$), the proposed RenewQIF in logistic model with data batch 8 ($N_B = 18,832$, $B = 28$), and RenewQIF$_{qc}$ without data batch 8 ($N_0 = 18,157$, $B = 27$).

	QIF			RenewQIF			RenewQIF$_{qc}$		
	$\hat{\beta}_B^\star$	ASE	p	$\tilde{\beta}_B$	ASE	p	$\tilde{\beta}_B$	ASE	p
Intercept	-0.90	0.088	0.000	-0.89	0.092	0.000	-0.91	0.094	0.000
Young	-0.22	0.050	0.000	-0.22	0.051	0.000	-0.22	0.052	0.000
Old	0.57	0.061	0.000	0.57	0.062	0.000	0.57	0.063	0.000
Seat Belt	-1.16	0.046	0.000	**-1.15**	0.047	0.000	**-1.14**	0.048	0.000
Drinking	0.36	0.049	0.000	**0.36**	0.049	0.000	**0.37**	0.050	0.000
Speed Limit	0.17	0.034	0.000	0.17	0.035	0.000	0.17	0.036	0.000
Vehicle Weight	-0.024	0.028	0.399	**-0.026**	0.029	0.365	**-0.025**	0.029	0.385
Air Bag	-1.00	0.050	0.000	-1.00	0.052	0.000	-1.00	0.053	0.000
Number of Lanes	-0.10	0.035	0.005	**-0.10**	0.037	0.008	**-0.11**	0.037	0.003
Drug Use	0.96	0.044	0.000	**0.95**	0.045	0.000	**0.94**	0.046	0.000
Distraction	-0.45	0.035	0.000	-0.45	0.036	0.000	-0.45	0.036	0.000
Surface Condition	0.16	0.046	0.001	0.15	0.049	0.002	0.16	0.050	0.002
Previous Accident	1.17	0.059	0.000	1.16	0.061	0.000	1.16	0.062	0.000

statistical inference.

This gap has been filled in this paper by the renewable QIF. The proposed RenewQIF method provides a new paradigm of renewable estimation and incremental inference, in which parameter estimates are recursively updated with current data and inferential statistics of historical data, but does not require the accessibility to any historical subject-level data. To achieve efficient communications between current data and historical summary statistics, we design an extended Spark's Lambda architecture to execute both data storage and analysis updates. Both proposed statistical methodology and computational algorithms have been investigated for their theoretical guarantees and examined numerically via extensive simulation studies. The proposed RenewQIF has been shown to be much more computationally efficient with no loss of inferential power in comparison to the oracle GEE or oracle QIF.

Additionally, we added a monitoring procedure to detect abnormal data batches in data streams for the proposed RenewQIF. The utilization of a goodness-of-fit test in the QIF framework enabled us to check the compatibility of two adjacent data batches effectively and efficiently. The proposed monitoring procedure has been integrated

into an extended Lambda architecture with an additional monitoring layer.

The formulation of RenewQIF is under the assumption that clusters arrive independently over data streams, and when cluster size $m = 1$, it reduces to generalized linear model as a special case. A direction of interest is to consider the case of inter-correlated batches; for example, serially dependent data streams generated by individual wearable devices. Such types of data streams are pervasive in healthcare where thousands of physiological measurements are recorded per second, such as body temperature, heart rate, respiratory rate and blood pressure (*Priyanka and Kulennavar*, 2014). Therefore, the analytic tools for the analysis of serially dependent data streams is an important future research as part of new analytics for handling massive data volumes and making decisions of medical treatment.

CHAPTER IV

Online Multivariate Regression Analysis with Heterogeneous Streaming Data

4.1 Introduction

The advent of distributed cluster-computing paradigms such as Apache Spark (*Bifet et al.*, 2015) has motivated new developments in data analytics for large-scale data processing. Such innovation enables effective analyses of streaming data assembled through, for example, national disease registries, mobile health consortia and infectious disease surveillance programs. One of the defining features for these streaming data is that observations become available sequentially over time at high velocity in some occasions. Researchers would utilize the sequence of data batches to answer scientific questions of interest including assessing disease biomarkers, monitoring product safety, validating drug efficacy and side-effects. In these scenarios, it is essential for practitioners to apply certain analytic tools to process data streams sequentially as part of real-time monitoring and decision-making.

This paper is motivated by a large-scale electronic health records (EHR) database managed by the Scientific Registry of Transplant Recipients (SRTR) since 1984. The EHR data is constantly updated at SRTR in which new patients are added to the transplant waiting list in the US every ten minutes, resulting in a yearly average over

25,000 transplants entered into SRTR since the mid-2000s. Due to the lack of suitable data analytic methods, such data collected in real-time have been analyzed in a static fashion, leading to latency in the transition of data to clinical knowledge. Also, this conventional data analysis approach is often challenged by limitations in data storage, data maintenance and computational capacity when dealing with data of such fast growing volumes. These analytic and computational challenges call for reliable and efficient real-time statistical methodology that promotes timely processing of data to improve clinical decision-making.

Our motivating data in this paper consists of a sequence of yearly updated EHR datasets of kidney transplants during the period from 1987 to 2017. Our analysis uses a total of 221,337 kidney transplant recipients in the USA with complete personal clinical information collected by SRTR. A primary analytic interest is to update estimation and inference for effects of the important risk factors on the outcome of post-transplant serum creatinine, an important biomarker of renal function to monitor the graft condition of the new organ. Thus, we aim to update these estimated effects on a regular basis shortly after the arrival of data each year. Traditional static analysis would first create a very large data file comprised of both old and new data, and then analyze a large dataset using suitable statistical methods and software. This traditional static approach is not efficient; if we plan to run the same analysis annually over the next 30 years, we need to acquire such continually growing data in each of the 30 years, and repeat 30 times the same data cleaning, pre-processing, and analysis procedures with the expanded data. This process is laborious, expensive, and time-consuming. Thus, it is appealing to develop a smarter solution to this type of data analysis task, in particular for streaming data that arrive at a fast rate with a large volume, such as mobile health data.

Most existing online streaming data analytics such as stochastic gradient descent (SGD) are built under the homogeneous assumption that all data batches are gen-

erated with the same underlying data generation mechanism. Additionally, each observation arriving over time is treated as being independently sampled. Arguably, such an *i.i.d.* assumption is only for mathematical convenience and may be violated in many real-world applications. In practice, different data batches are often heterogeneous and correlated over the sampling points. In the SRTR dataset, it is clinically more so that associations between some of the risk predictors and post-transplant serum creatinine may evolve dynamically, rather than remaining constant over the 30-year period due to many factors such as better organ-matching strategies. Improvements in medical care or facilities over years, for example, may be modeled as temporal confounding variables while the risk factors (e.g. age, sex, BMI) of main interest may be assumed as fixed effects. In the literature, continuous data streams are structured as time series data. For instance, traffic sensors (*Chen et al.*, 2005), health sensors (*Dias and Cunha*, 2018), transaction logs (*Zhang et al.*, 2009), and activity logs (*Ciuciu et al.*, 2008). Incorporating dynamic heterogeneity and correlation in the analysis of data streams leads to increased complexity in modeling and statistical inference, which is known as a difficult problem even in offline settings (*L'Heureux et al.*, 2017; *Sadik et al.*, 2018).

State space models, also termed as dynamic models, are a very flexible class of models for analyzing time series data or longitudinal data when the number of repeated observations is too large (*West and Harrison*, 1997; *Kitagawa*, 1987; *Jørgensen et al.*, 1999). It is widely used in many areas, such as economics, engineering, and biology. The classical state space models refer to a class of hierarchical models in that the observation process is driven by a latent state process that may incorporate trend, seasonal, or time-varying covariate effects.

To analyze streaming data, state space models appear very flexible in the modeling of certain stochastic behaviors where the latent state process may account for both inter-data batch correlation and time-varying heterogeneity in a sequence of observed

data batches. This latent process represents evolving batch-specific effects, either temporally or spatially. In most cases, learning the latent states, say"filtering" and "smoothing", is a primary goal of statistical analyses. However, our analytic need in streaming data analysis is based on real-time regression, where we focus primarily on parameter estimation and inference on the fixed effects of risk factors that are shared by the sequence of data batches. This type of state space model with the addition of fixed effects is termed as *state space mixed models* by *Czado and Song* (2008).

In applications with a large volume of streaming datasets, existing offline approaches to fit state space models require large amounts of computing memory on data storage, and fitting such models repeatedly over time may become computationally expensive and even infeasible. In the case where the sample space of the latent state is finite such as hidden Markov model, an efficient online Expectation-Maximization (EM) algorithm (*Dempster et al.*, 1977) based on sufficient statistics has been developed by *Cappé* (2011). But this algorithm is greatly challenged from a computational perspective in the state space models in that the sample space of the latent process is infinite, leading to the invocation of Monte Carlo computation approximation (*Cappé and Moulines*, 2009). It is worth noting that most online methods for fitting state space models are built in a Bayesian paradigm where the inference on the latent process, rather than on the fixed effects, is of primary interest. One such example is streaming variational Bayes method (*Broderick et al.*, 2013) that is developed in a Gaussian process state space model (*Frigola et al.*, 2014). There is a lack of online regression analysis (MORA) via state space models with the focus on the estimation and inference for fixed effects, adjusting for dynamic effects governed by the latent process. In a regression analysis, fixed effects are of primary interest to examine the relationship between an outcome and covariates. State space regression models that contain both deterministic and random predictors have been widely studied in many static settings, for example, in the analysis of longitudinal count

data by *Jørgensen et al.* (1999) and binomial response by *Czado and Song* (2008).

In this paper, we develop a Kalman filter on online estimation procedure for linear state space mixed models. This new method enables to update real-time estimation of both fixed effects and their standard errors. In an online regression paradigm based on the linear state space mixed models (LSSMM), renewable estimation and incremental inference methodology (*Luo and Song*, 2020) on fixed effects will gain efficiency along the utility of streaming data. In the meanwhile, the inter-data batch heterogeneity is modeled by a latent batch-specific effect that follows a stationary Gaussian AR(1) process. A crucial step in the proposed MORA is to obtain the conditional distribution of state variables given the data and other model parameters, similar to the E-step in the EM algorithm. Calculating the maximum likelihood estimation (MLE) is challenging due to the lack of closed form expressions whose likelihood function typically involves high-dimensional integrals. In the setting of MORA, these integrals become infinite where data batches arrive perpetually over time. Thus, certain approximations are inevitable.

Approximation based on Monte Carlo is less appealing as far as the computation burden concerns. An analytic solution to the approximation based on the best linear unbiased predictor (BLUP) is our choice in this paper, which is given as an extension of the classical Kalman filter recursions (*Harvey*, 1981; *Song*, 2007). The Kalman filter is a computation efficient method that utilizes the first-order Markovian properties of the latent state to calculate conditional moments recursively. The resulting recursive estimation method fits well to the need of sequentially processing MORA with the historical subject-level data being not retrievable and thus not used. The proposed inference essentially resembles the offline version of *Kalman Estimating Equation* (KEE) (*Song*, 2007). KEE is a generalization of the EM algorithm, in which the E-step is based on a recursive BLUP, and the M-step solves an augmented estimating equation. KEE avoids the use of Monte Carlo in the E-step, it instead adopts

an analytic recursive BLUP to carry out the Kalman filter. Our proposed multivariate online regression analysis (MORA) method generalized further those offline ideas to add in heterogeneity in streaming data. Our generalization consists of two new technical elements: the first is to use Kalman filter in the E-step to recursively update mean of dynamic latent states, and the second is to update the fixed effects using summary statistics of historical data rather than historical individual-level data (*Luo and Song*, 2020). In the setting of linear state space mixed models, solving KEE for the fixed effects has a closed-form solution that is linearly separable by data batches. This separability makes the generalization of the offline KEE into online KEE feasible for streaming data.

The organization of this paper is as follows. Section 4.2 begins with a brief overview of modeling assumptions, and relevant recursive formulas to the Kalman filter which are needed for recursive updating of the latent state variables and quantities concerning inference. Section 4.3 presents key analytic derivations and establishes theoretical guarantees for our proposed MORA. Section 4.5 concerns the architecture and pesudo code for the implementation of MORA via the Rho architecture in Sparks (*Luo and Song*, 2020). Simulation experiments are given in Section 4.6 to evaluate the performance of MORA. We apply MORA to analyze the EHR data example, adjusting for some time effects in Section 4.7. Finally we make some concluding remarks in Section 4.8. Detailed proofs of the large sample properties are included in the Appendix.

4.2 Model

4.2.1 Formulation

At a time point $b \geq 2$, a sequence of b data batches arriving sequentially with a cumulative sample size $N_b = \sum_{j=1}^{b} n_j$, each with sample size n_j, $j = 1, \ldots, b$. The

j-th data batch is denoted by $D_j = \{\boldsymbol{y}_j, \boldsymbol{X}_j, \boldsymbol{Z}_j\}$ where $\boldsymbol{y}_j = (y_{j1}, \ldots, y_{jn_j})^T$, $\boldsymbol{X}_j = (\boldsymbol{x}_{j1}, \ldots, \boldsymbol{x}_{jn_j})^T$ and $\boldsymbol{Z}_j = (\boldsymbol{z}_{j1}, \ldots, \boldsymbol{z}_{jn_j})^T$, $j = 1, \ldots, b$ are the vector of response and the matrices of associated covariates in the observed and latent processes, respectively. Clearly, $n_j = |D_j|$. Let $D_b^\star = \{D_1, \ldots, D_b\}$ be the cumulative data up to batch b with $N_b = |D_b^\star|$. Note that in a streaming data setting, batch size n_b is not supposed to diverge to infinity, but the cumulative sample size N_b is. For simplicity, D_b may be used as the set of indices for data points involved. In the framework of state space mixed models, we consider a first-order Markov process $\{\boldsymbol{\beta}_b, b \geq 1\}$ to account over-batch heterogeneity. Moreover, we assume that two series $\{D_b, b \geq 1\}$ and $\{\boldsymbol{\beta}_b, b \geq 1\}$ follow a dynamic hierarchical system, as shown in Figure 4.1, defined as follows:

(A1) Given $\boldsymbol{\beta}_b$, $\boldsymbol{y}_b$ is conditionally independent of the other $\boldsymbol{y}_b$'s;

(A2) $\{\boldsymbol{\beta}_b, b \geq 1\}$ is a first-order Markov process with initial state $\boldsymbol{\beta}_1$ being assumed to be a fixed unknown parameter;

(A3) $\boldsymbol{y}_b = \boldsymbol{X}_b\boldsymbol{\alpha} + \boldsymbol{Z}_b\boldsymbol{\beta}_b + \boldsymbol{\epsilon}_b$, with $\boldsymbol{\epsilon}_b \overset{iid}{\sim} \mathcal{N}(\boldsymbol{0}, \phi\boldsymbol{I}_{n_b})$ where $\boldsymbol{\alpha}$ is the common fixed effects of covariates $\boldsymbol{X}_b$, and $\boldsymbol{\beta}_b$ is the random effects of covariates $\boldsymbol{Z}_b$;

(A4) $\boldsymbol{\beta}_{b+1} = \boldsymbol{B}_b\boldsymbol{\beta}_b + \boldsymbol{\xi}_b$, where $\boldsymbol{B}_b$ is a $q \times q$ transition matrix and Gaussian white noise $\boldsymbol{\xi}_b \overset{iid}{\sim} \mathcal{N}(\boldsymbol{0}, \delta\boldsymbol{I}_q)$. The two random errors $\boldsymbol{\xi}_b$ and $\boldsymbol{\epsilon}_b$ are independent. In particular, for a stationary AR(1) process, $\boldsymbol{B}_b = \rho\boldsymbol{I}_q$ with $|\rho| < 1$; for a random walk process, $\boldsymbol{B}_b = \boldsymbol{I}_q$, so $\boldsymbol{\beta}_{b+1} = \boldsymbol{\beta}_b + \boldsymbol{\xi}_b$, where variance $\delta = 0$ leads to the homogeneity case $\boldsymbol{\beta}_{b+1} = \boldsymbol{\beta}_b$, an assumption extensively used in the current literature of online regression analysis.

Among many state space models, in this paper we focus on a class of linear state space models with a stationary latent process. That is, we further assume (A3) and (A4).

4.2.2 Conditional and Marginal Moments

We derive Kalman filter that is essential to establish a recursive BLUP for the proposed MORA. To do so, we first derive the conditional and marginal moments of the

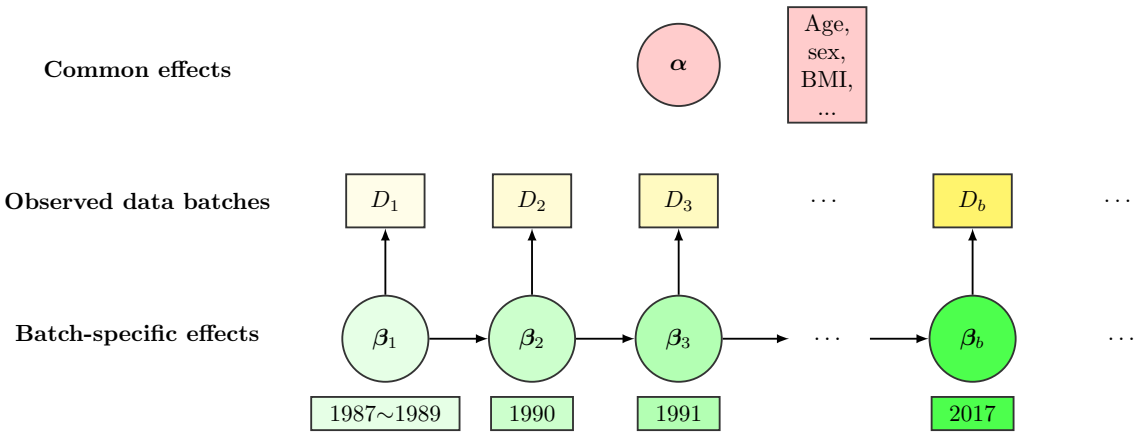

Figure 4.1: A comb structure for dynamic hierarchical system. Starting with an initial effect $\boldsymbol{\beta}_1$, subsequent $\boldsymbol{\beta}_j$'s are governed by a sequence of linear transitions specified in (A4). The observed process $\{\boldsymbol{y}_b, b \geq 1\}$ includes both common effect $\boldsymbol{\alpha}$ and varying batch-specific effects $\{\boldsymbol{\beta}_b, b \geq 1\}$.

Gaussian stationary linear model given by (A1)-(A4), when (A4) is an AR(1) process.

(a) The conditional moments of the observed process and the latent process are respectively

$$\mathbb{E}(\boldsymbol{y}_b \mid \boldsymbol{\beta}_b) = \boldsymbol{X}_b\boldsymbol{\alpha} + \boldsymbol{Z}_b\boldsymbol{\beta}_b, \ \text{var}(\boldsymbol{y}_b \mid \boldsymbol{\beta}_b) = \phi\boldsymbol{I}_{n_b};$$

$$\mathbb{E}(\boldsymbol{\beta}_{b+1} \mid \boldsymbol{\beta}_b) = \boldsymbol{B}_b\boldsymbol{\beta}_b, \ \text{var}(\boldsymbol{\beta}_{b+1} \mid \boldsymbol{\beta}_b) = \delta\boldsymbol{I}_q.$$

(b) The marginal moments of the observed process are given by

$$\mathbb{E}(\boldsymbol{y}_b) = \boldsymbol{X}_b\boldsymbol{\alpha} + \boldsymbol{Z}_b\mathbb{E}(\boldsymbol{\beta}_b), \ \text{var}(\boldsymbol{y}_b) = \phi\boldsymbol{I}_{n_b} + \boldsymbol{Z}_b\text{var}(\boldsymbol{\beta}_b)\boldsymbol{Z}_b^T,$$

where under an AR(1) process for $\{\boldsymbol{\beta}_b, b \geq 1\}$,

$$\begin{aligned}
\mathbb{E}(\boldsymbol{\beta}_b) &= \boldsymbol{B}_{b-1} \cdots \boldsymbol{B}_1 \boldsymbol{\beta}_1 = \rho^{b-1} \boldsymbol{\beta}_1, \ b \geq 2 \\
\text{var}(\boldsymbol{\beta}_b) &= \delta \boldsymbol{I}_q + \delta \boldsymbol{B}_{b-1} \boldsymbol{B}_{b-1}^T + \delta (\boldsymbol{B}_{b-1} \boldsymbol{B}_{b-2})(\boldsymbol{B}_{b-1} \boldsymbol{B}_{b-2})^T + \cdots + \\
&\quad \delta (\boldsymbol{B}_{b-1} \boldsymbol{B}_{b-2} \cdots \boldsymbol{B}_1)(\boldsymbol{B}_{b-1} \boldsymbol{B}_{b-2} \cdots \boldsymbol{B}_1)^T \\
&= \frac{\delta(1-\rho^{2b})}{1-\rho^2} \boldsymbol{I}_q.
\end{aligned}$$

(c) The covariances are

$$\begin{aligned}
&\text{cov}(\boldsymbol{y}_b, \boldsymbol{\beta}_b) = \boldsymbol{Z}_b \text{var}(\boldsymbol{\beta}_b), \\
&\text{cov}(\boldsymbol{\beta}_b, \boldsymbol{\beta}_{b+h}) = \text{var}(\boldsymbol{\beta}_b)(\boldsymbol{B}_{b+h-1} \cdots \boldsymbol{B}_b)^T = \rho^h \text{var}(\boldsymbol{\beta}_b), \\
&\text{cov}(\boldsymbol{y}_b, \boldsymbol{y}_{b+h}) = \boldsymbol{Z}_b \text{cov}(\boldsymbol{\beta}_b, \boldsymbol{\beta}_{b+h}) \boldsymbol{Z}_{b+h}^T = \rho^h \text{var}(\boldsymbol{\beta}_b) \boldsymbol{Z}_b \boldsymbol{Z}_b^T, \\
&\text{cov}(\boldsymbol{y}_b, \boldsymbol{\beta}_{b+h}) = \boldsymbol{Z}_b \text{cov}(\boldsymbol{\beta}_b, \boldsymbol{\beta}_{b+h}) = \rho^h \boldsymbol{Z}_b \text{var}(\boldsymbol{\beta}_b), \\
&\text{cov}(\boldsymbol{y}_{b+h}, \boldsymbol{\beta}_b) = \boldsymbol{Z}_{b+h} \text{cov}(\boldsymbol{\beta}_b, \boldsymbol{\beta}_{b+h}) = \rho^h \boldsymbol{Z}_b \text{var}(\boldsymbol{\beta}_b).
\end{aligned}$$

4.2.3 Kalman Filter

The Kalman filter is used to estimate the conditional mean and variance of latent state variable or batch-specific effects $\boldsymbol{\beta}_b$'s. Under the model of (A1)-(A4), given the prediction at data batch b with the conditional mean $\boldsymbol{m}_{b-1}$ and covariance $\boldsymbol{C}_{b-1}$, the Kalman filter proceeds recursively as follows:

(i) compute two predictions

$$\boldsymbol{\beta}_b \mid D_{b-1}^\star \sim [\boldsymbol{B}_{b-1} \boldsymbol{m}_{b-1}; \boldsymbol{H}_b] \text{ and } D_b \mid D_{b-1}^\star \sim [\boldsymbol{f}_b; \boldsymbol{Q}_b],$$

where

$$\begin{aligned}
\boldsymbol{H}_b &= \mathrm{var}(\boldsymbol{\beta}_b \mid D^\star_{b-1}) = \boldsymbol{B}_{b-1}\boldsymbol{C}_{b-1}\boldsymbol{B}^T_{b-1} + \delta \boldsymbol{I}_q \overset{AR(1)}{=\!=} \rho^2 \boldsymbol{C}_{b-1} + \delta \boldsymbol{I}_q, \\
\boldsymbol{f}_b &= \mathbb{E}(D_b \mid D^\star_{b-1}) = \boldsymbol{Z}_b\boldsymbol{B}_{b-1}\boldsymbol{m}_{b-1} + \boldsymbol{X}_b\boldsymbol{\alpha} \overset{AR(1)}{=\!=} \rho \boldsymbol{Z}_b\boldsymbol{m}_{b-1} + \boldsymbol{X}_b\boldsymbol{\alpha}, \\
\boldsymbol{Q}_b &= \mathrm{var}(D_b \mid D^\star_{b-1}) = \phi \boldsymbol{I}_{n_b} + \boldsymbol{Z}_b\boldsymbol{H}_b\boldsymbol{Z}^T_b.
\end{aligned}$$

(ii) Let $\boldsymbol{K}_b = \boldsymbol{H}^T_b\boldsymbol{Z}^T_b\boldsymbol{Q}^{-1}_b$, and update the prediction of $\boldsymbol{\beta}_b$ given $D^\star_b$,

$$\boldsymbol{\beta}_b \mid D^\star_b \sim [\boldsymbol{m}_b; \boldsymbol{C}_b],$$

where

$$\begin{aligned}
\boldsymbol{m}_b &= \mathbb{E}(\boldsymbol{\beta}_b \mid D^\star_b) = \boldsymbol{B}_{b-1}\boldsymbol{m}_{b-1} + \boldsymbol{H}^T_b\boldsymbol{Z}^T_b\boldsymbol{Q}^{-1}_b(\boldsymbol{Y}_b - \boldsymbol{f}_b) \overset{AR(1)}{=\!=} \rho\boldsymbol{m}_{b-1} + \boldsymbol{K}_b(\boldsymbol{Y}_b - \boldsymbol{f}_b), \\
\boldsymbol{C}_b &= \mathrm{var}(\boldsymbol{\beta}_b \mid D^\star_b) = (\boldsymbol{I}_q - \boldsymbol{K}_b\boldsymbol{Z}_b)\boldsymbol{H}_b.
\end{aligned}$$

Consequently, the two inferential quantities needed in MORA can be updated by the Kalman filter of the following form:

$$\mathbb{E}(\boldsymbol{\beta}_b \mid D^\star_b, \tilde{\boldsymbol{\alpha}}_{b-1}, \tilde{\boldsymbol{\zeta}}_{b-1}) = \boldsymbol{m}_b, \ \mathrm{var}(\boldsymbol{\beta}_b \mid D^\star_b, \tilde{\boldsymbol{\alpha}}_{b-1}, \tilde{\boldsymbol{\zeta}}_{b-1}) = \boldsymbol{C}_b, \tag{4.1}$$

where $\boldsymbol{\zeta} = (\phi, \rho, \delta)$ is the vector of nuisance parameters.

4.2.4 Mean Square Error

Let $\vec{\boldsymbol{m}}_b = (\boldsymbol{m}^T_1, \boldsymbol{m}^T_2, \ldots, \boldsymbol{m}^T_b)^T$ and $\vec{\boldsymbol{\beta}}_b = (\boldsymbol{\beta}^T_1, \boldsymbol{\beta}^T_2, \ldots, \boldsymbol{\beta}^T_b)^T$. Then

$$\vec{\boldsymbol{\beta}}_b \mid D^\star_b \sim [\vec{\boldsymbol{m}}_b; \boldsymbol{\Sigma}_b],$$

where $\boldsymbol{\Sigma}_b = \mathbb{E}\left\{(\vec{\boldsymbol{\beta}}_b - \vec{\boldsymbol{m}}_b)(\vec{\boldsymbol{\beta}}_b - \vec{\boldsymbol{m}}_b)^T\right\}$ is the mean square error matrix of $bq \times bq$ dimension. Matrix $\boldsymbol{\Sigma}_b$ has diagonal elements given by $\boldsymbol{\Sigma}_b(j,j) = \boldsymbol{C}_j$ $(j = 1, \ldots, b)$, and

the off-diagonal blocks given by $\boldsymbol{\Sigma}_b(j, j+h) = \boldsymbol{\Sigma}_b^T(j+h, j) = \mathbb{E}\left\{(\boldsymbol{\beta}_j - \boldsymbol{m}_j)(\boldsymbol{\beta}_{j+h} - \boldsymbol{m}_{j+h})^T\right\}$. Following *Jørgensen and Song* (2007), we obtain

$$\boldsymbol{\Sigma}_b(j, j+h) = \rho^h \boldsymbol{C}_{j+h} \prod_{i=0}^{h-1} \boldsymbol{W}_{j+i}^{-1} \boldsymbol{C}_{j+i},$$

where $\boldsymbol{W}_j = \mathrm{var}(\boldsymbol{\beta}_{j+1} - \rho \boldsymbol{m}_j) = \rho^2 \boldsymbol{C}_j + \delta \boldsymbol{I}_q/(1-\rho^2)$. In particular, $\boldsymbol{\Sigma}_b(j, j+1) = \rho \boldsymbol{C}_{j+1} \boldsymbol{W}_j^{-1} \boldsymbol{C}_j$.

4.3 Online Regression Analysis

4.3.1 Estimation of Fixed Effects

In this paper, we focus on online estimation and inference on the common fixed effect $\boldsymbol{\alpha}$, which will benefit from the accumulation of data batches. For batch-specific effects $\boldsymbol{\beta}_b$'s, we just repeat the results from a single batch based analysis.

To proceed with the maximum likelihood estimation, we first write out the marginal likelihood function for the parameters of interest $(\boldsymbol{\alpha}, \boldsymbol{\zeta})$:

$$L(\boldsymbol{\alpha}, \boldsymbol{\zeta} \mid D_b^\star) = \int_{\mathbb{R}^{q(b-1)}} P(D_j \mid \boldsymbol{\beta}_j; \boldsymbol{\alpha}, \boldsymbol{\zeta}) P(\boldsymbol{\beta}_j \mid \boldsymbol{\beta}_{j-1}; \boldsymbol{\zeta}) d\boldsymbol{\beta}_2 d\boldsymbol{\beta}_3 \cdots d\boldsymbol{\beta}_b,$$

where the integral is $q(b-1)$-dimensional, and both $P(D_j \mid \boldsymbol{\beta}_j; \boldsymbol{\alpha}, \boldsymbol{\zeta})$ and $P(\boldsymbol{\beta}_j \mid \boldsymbol{\beta}_{j-1}; \boldsymbol{\zeta})$ are multivariate conditional normal distributions.

Treating $\boldsymbol{\beta}_b$'s as "missing data", we obtain the augmented log-likelihood:

$$\ell(\boldsymbol{\alpha}, \boldsymbol{\zeta} \mid D_b^\star, \vec{\boldsymbol{\beta}}_b) = \sum_{j=1}^{b} \log P(D_j \mid \boldsymbol{\beta}_j, \boldsymbol{\alpha}, \boldsymbol{\zeta}) + \sum_{j=1}^{b-1} \log P(\boldsymbol{\beta}_{j+1} \mid \boldsymbol{\beta}_j, \boldsymbol{\zeta}).$$

In order to use the EM algorithm to obtain MLE, we maximize the following Q-function $Q(\boldsymbol{\alpha}, \boldsymbol{\zeta} \mid \boldsymbol{\alpha}', \boldsymbol{\zeta}') = \mathbb{E}\{\ell(\boldsymbol{\alpha}, \boldsymbol{\zeta} \mid D_b^\star, \vec{\boldsymbol{\beta}}_b)\}$, where the expectation is taken under the conditional distribution $P(\vec{\boldsymbol{\beta}}_b \mid D_b^\star, \boldsymbol{\alpha}', \boldsymbol{\zeta}')$. Here $\boldsymbol{\alpha}'$ and $\boldsymbol{\zeta}'$ being updated

parameter values from the previous iteration. This maximization can be carried out by solving the augmented score equations:

$$\begin{aligned} \boldsymbol{s}_1(\boldsymbol{\alpha}, \boldsymbol{\zeta}) &= \sum_{j=1}^{b} \boldsymbol{X}_j^T \left\{ \boldsymbol{y}_j - \boldsymbol{X}_j \boldsymbol{\alpha} - \boldsymbol{Z}_j \mathbb{E}(\boldsymbol{\beta}_j \mid D_b^\star, \boldsymbol{\alpha}', \boldsymbol{\zeta}') \right\} = \mathbf{0}, \\ \boldsymbol{s}_2(\boldsymbol{\alpha}, \boldsymbol{\zeta}) &= \sum_{j=1}^{b-1} \left\{ \boldsymbol{\beta}_{j+1} - \boldsymbol{B}_j \mathbb{E}(\boldsymbol{\beta}_j \mid D_b^\star, \boldsymbol{\alpha}', \boldsymbol{\zeta}') \right\} = \mathbf{0}. \end{aligned} \tag{4.2}$$

Instead of using the Monte Carlo technique to compute the conditional mean $\mathbb{E}(\boldsymbol{\beta}_j \mid D_b^\star, \boldsymbol{\alpha}', \boldsymbol{\zeta}')$, BLUP (*Robinson*, 1991) is used to speed up computation. An obvious advantage for the utility of BLUP is that it can be fast carried out via the Kalman recursive formula. In MORA, since historical subject-level data are not available, we adopt Kalman filter $\mathbb{E}(\boldsymbol{\beta}_b \mid D_b, \tilde{\boldsymbol{\alpha}}_{b-1}, \tilde{\boldsymbol{\zeta}}_{b-1})$, which is recursively updated with the use of only individual-level data in current data batch D_b, rather than historical cumulative data $D_{b-1}^\star$. Upon the arrival of one data batch, following *Titterington* (1984) and *Cappé and Moulines* (2009), we perform one-step update recursive formula via the EM algorithm rather than iteratively till convergence.

To further speed up the algorithm, instead of solving $\boldsymbol{s}_2 = \mathbf{0}$, we propose to use the method of moments estimators to estimate $\boldsymbol{\zeta}$, as the cumulative sample size N_b would quickly increase, maybe the choice of the estimator for $\boldsymbol{\zeta}$ becomes less critical.

In summary, the estimation procedure proceeds as follows:

- Step 1: Choose initial values for parameters, denoted by $\tilde{\boldsymbol{\alpha}}_0$ and $\tilde{\boldsymbol{\zeta}}_0$.
- Step 2: For $b \geq 1$, given $\sqrt{N_{b-1}}$-consistent estimators $(\tilde{\phi}_{b-1}, \tilde{\rho}_{b-1}, \tilde{\delta}_{b-1})$ from the prior iteration, we update fixed effects $\tilde{\boldsymbol{\alpha}}_{b-1}$ to $\tilde{\boldsymbol{\alpha}}_b$ by the solution to the following unbiased aggregated Kalman Estimating Equation (KEE):

$$\tilde{\boldsymbol{U}}_b(\boldsymbol{\alpha}) = \sum_{i=1}^{N_b} \boldsymbol{U}_i(\boldsymbol{\alpha}) = \sum_{j=1}^{b} \boldsymbol{X}_j^T \left(\boldsymbol{y}_j - \boldsymbol{X}_j \boldsymbol{\alpha} - \boldsymbol{Z}_j \boldsymbol{m}_j \right) = \mathbf{0}, \tag{4.3}$$

where $\boldsymbol{m}_b = \mathbb{E}(\boldsymbol{\beta}_b \mid D_b, \tilde{\boldsymbol{\alpha}}_{b-1}, \tilde{\boldsymbol{\zeta}}_{b-1})$ is the Kalman filter obtained upon the arrival of D_b, the previous updates $\tilde{\boldsymbol{\alpha}}_{b-1}$ and $\tilde{\boldsymbol{\zeta}}_{b-1}$.

- Step 3: Given $\tilde{\boldsymbol{\alpha}}_b$, update the parameter vector $\tilde{\boldsymbol{\zeta}}_{b-1}$ to $\tilde{\boldsymbol{\zeta}}_b$ by the method of moments given in Section 4.3.2.

In the Gaussian linear model, equation (4.3) has a closed-form solution of the form:

$$\tilde{\boldsymbol{\alpha}}_b = \left(\sum_{j=1}^{b} \boldsymbol{X}_j^T \boldsymbol{X}_j \right)^{-1} \left(\sum_{j=1}^{b} \boldsymbol{X}_j^T (\boldsymbol{y}_j - \boldsymbol{Z}_j \boldsymbol{m}_j) \right).$$

4.3.2 Estimation of Dispersion and Correlation Parameters

We use the method of moments to estimate both dispersion and correlation parameters $\boldsymbol{\zeta} = (\phi, \rho, \delta)^T$. First, note that the equation $\text{var}(\boldsymbol{y}_j - \boldsymbol{X}_j\boldsymbol{\alpha} - \boldsymbol{Z}_j\boldsymbol{m}_j) = \phi \boldsymbol{I}_{n_j} + \boldsymbol{Z}_j \boldsymbol{C}_j \boldsymbol{Z}_j^T$ leads to the following moment estimator for the dispersion parameter ϕ,

$$\hat{\phi}_b^\star = \frac{1}{N_b} \sum_{j=1}^{b} (\boldsymbol{y}_j - \boldsymbol{X}_j \hat{\boldsymbol{\alpha}}_b^\star - \boldsymbol{Z}_j \boldsymbol{m}_j)^T (\boldsymbol{y}_j - \boldsymbol{X}_j \hat{\boldsymbol{\alpha}}_b^\star - \boldsymbol{Z}_j \boldsymbol{m}_j) - \frac{1}{N_b} \sum_{j=1}^{b} \sum_{i \in D_j} \boldsymbol{P}_j(i,i),$$

where $\boldsymbol{P}_j = \boldsymbol{Z}_j \boldsymbol{C}_j \boldsymbol{Z}_j^T$ and $\boldsymbol{P}_j(i,i)$ denote its i-th block diagonal. Additionally, note that

$$\begin{aligned}
\delta \boldsymbol{I}_q &= \text{var}(\boldsymbol{\beta}_{j+1} - \boldsymbol{B}_j \boldsymbol{\beta}_j) \\
&= \text{var}\left\{ \boldsymbol{\beta}_{j+1} - \boldsymbol{m}_{j+1} - \boldsymbol{B}_j (\boldsymbol{\beta}_j - \boldsymbol{m}_j) \right\} + \text{var}(\boldsymbol{m}_{j+1} - \boldsymbol{B}_j \boldsymbol{m}_j) \\
&= \boldsymbol{C}_{j+1} + \boldsymbol{B}_j \boldsymbol{C}_j \boldsymbol{B}_j^T - 2\boldsymbol{\Sigma}_b(j+1, j) \boldsymbol{B}_j^T + \text{var}(\boldsymbol{m}_{j+1} - \boldsymbol{B}_j \boldsymbol{m}_j).
\end{aligned}$$

Similarly, let $\boldsymbol{E}_j = \boldsymbol{C}_{j+1} + \boldsymbol{B}_j \boldsymbol{C}_j \boldsymbol{B}_j^T - 2\boldsymbol{\Sigma}_b(j+1, j)\boldsymbol{B}_j^T$ and $\boldsymbol{E}_j(i,i)$ denote its i-th block diagonal. A moment estimator of δ is given by

$$\hat{\delta}_b^\star = \frac{1}{bq}\sum_{j=1}^{b}(\boldsymbol{m}_{j+1} - \boldsymbol{B}_j\boldsymbol{m}_j)^T(\boldsymbol{m}_{j+1} - \boldsymbol{B}_j\boldsymbol{m}_j) + \frac{1}{bq}\sum_{j=1}^{b}\sum_{i=1}^{q}\boldsymbol{E}_j(i,i).$$

$\sqrt{N_b}$-consistent estimators of ϕ and δ are updated by

$$\begin{aligned}\tilde{\phi}_b &= \frac{N_{b-1}}{N_b}\tilde{\phi}_{b-1} + \frac{n_b}{N_b}\hat{\phi}_b,\\ \tilde{\delta}_b &= \frac{b-2}{b-1}\tilde{\delta}_{b-1} + \frac{1}{b-1}\hat{\delta}_b,\ b \geq 1\end{aligned}$$

where $\hat{\phi}_b = \frac{1}{n_b}(\boldsymbol{y}_b - \boldsymbol{X}_b\tilde{\boldsymbol{\alpha}}_b - \boldsymbol{Z}_b\boldsymbol{m}_b)^T(\boldsymbol{y}_b - \boldsymbol{X}_b\tilde{\boldsymbol{\alpha}}_b - \boldsymbol{Z}_b\boldsymbol{m}_b) - \frac{1}{n_b}\sum_{i\in D_b}\boldsymbol{P}_b(i,i)$, $\hat{\delta}_b = \frac{1}{q}\|\boldsymbol{m}_b - \boldsymbol{B}_{b-1}\boldsymbol{m}_{b-1}\|^2 + \frac{1}{q}\sum_{i=1}^{q}\boldsymbol{E}_b(i,i)$.

The estimation of ρ is

$$\mathrm{cov}(\boldsymbol{m}_b, \boldsymbol{m}_{b-1}) = \rho\mathrm{var}(\boldsymbol{m}_{b-1}) + \boldsymbol{C}_{b-1}^T\mathrm{cov}(\boldsymbol{Y}_b - \boldsymbol{f}_b, \boldsymbol{m}_{b-1}) = \rho\boldsymbol{C}_{b-1}.$$

Therefore, the lag-1 autocorrelation of the standardized filtering may serve as an estimator of ρ. We carry out the updating in the following way:

$$\tilde{\rho}_b = \frac{\sum_{j=1}^{b}\boldsymbol{m}_j^T\boldsymbol{m}_{j+1}}{\sum_{j=2}^{b}\boldsymbol{m}_j^T\boldsymbol{m}_j},\ b \geq 2 \text{ and } \tilde{\rho}_1 = 0.$$

4.4 Theoretical Guarantees

In this section, we establish large sample properties under the assumptions (A1)-(A4). Let a neighborhood around true value $\boldsymbol{\alpha}_0$ be $\mathbb{N}_\epsilon(\boldsymbol{\alpha}_0) = \{\boldsymbol{\alpha} : \|\boldsymbol{\alpha} - \boldsymbol{\alpha}_0\|_2 \leq \epsilon\}$. The sensitivity matrix and variability matrix are given respectively by $\tilde{\boldsymbol{S}}_b(\boldsymbol{\alpha}) = \mathbb{E}_{\boldsymbol{\alpha}}\left\{-\frac{\partial\tilde{\boldsymbol{U}}_b(\boldsymbol{\alpha})}{\partial\boldsymbol{\alpha}^T}\right\}$ and $\tilde{\boldsymbol{V}}_b(\boldsymbol{\alpha}) = \mathbb{E}_{\boldsymbol{\alpha}}\left\{\sum_{i=1}^{N_b}\boldsymbol{U}_i(\boldsymbol{\alpha})\boldsymbol{U}_i^T(\boldsymbol{\alpha})\right\}$. We assume the following regularity conditions:

(C1) The unbiasedness of KEE: $\mathbb{E}_{\boldsymbol{\alpha}}\{\tilde{\boldsymbol{U}}_b(\boldsymbol{\alpha})\} = \mathbf{0}$ if and only if $\boldsymbol{\alpha} = \boldsymbol{\alpha}_0$.

(C2) The sensitivity matrix $\tilde{\boldsymbol{S}}_b(\boldsymbol{\alpha})$ is Lipschitz continuous for $\boldsymbol{\alpha} \in \Theta$.

(C3) The variability matrix $\tilde{\boldsymbol{V}}_b(\boldsymbol{\alpha})$ is positive-definite for $\boldsymbol{\alpha} \in \mathbb{N}_\epsilon(\boldsymbol{\alpha}_0)$.

The unbiasedness condition (C1) is not only required for consistency but also implies the $\boldsymbol{\zeta}$-insensitivity of the estimating equation, namely $\mathbb{E}\left\{\frac{\partial \tilde{\boldsymbol{U}}_b(\boldsymbol{\alpha})}{\partial \boldsymbol{\zeta}^T}\right\} = \mathbf{0}$. This property ensures that the efficiency of the nuisance parameter estimators would have a marginal effect on the estimate of $\boldsymbol{\alpha}$. Conditions (C2) and (C3) are required to establish both estimation consistency and asymptotic normality.

Theorem IV.1. *Under regularity conditions (C1)-(C3) that hold automatically in linear models. For fixed ρ, ϕ and δ, $\tilde{\boldsymbol{\alpha}}_b$ is consistent and asymptotically normal*

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

with $\boldsymbol{\Sigma}(\boldsymbol{\alpha}_0) = \lim_{N_b} N_b \tilde{\boldsymbol{J}}_b^{-1}(\boldsymbol{\alpha}_0)$, where $\tilde{\boldsymbol{J}}_b(\boldsymbol{\alpha}_0) = \tilde{\boldsymbol{S}}_b^T(\boldsymbol{\alpha}_0)\tilde{\boldsymbol{V}}_b^{-1}(\boldsymbol{\alpha}_0)\tilde{\boldsymbol{S}}_b(\boldsymbol{\alpha}_0)$ is the Godambe information matrix of the inference function given in estimating equation (4.3).

where $\tilde{\boldsymbol{U}}_b(\boldsymbol{\alpha})$ is the aggregated score function specified in equation (4.3). The asymptotic covariance matrix is estimated by $\widetilde{\boldsymbol{\Sigma}}_b = N_b \tilde{\boldsymbol{J}}_b^{-1}(\tilde{\boldsymbol{\alpha}}_b)$.

The Godambe information matrix is calculated as follows. It is easy to see that the $p \times p$ sensitivity matrix $\tilde{\boldsymbol{S}}_b(\boldsymbol{\alpha})$:

$$\tilde{\boldsymbol{S}}_b(\boldsymbol{\alpha}) = \sum_{j=1}^{b} \boldsymbol{X}_j^T \left\{\boldsymbol{X}_j + \boldsymbol{Z}_j \boldsymbol{L}_j(\boldsymbol{\alpha})\right\},$$

where $\boldsymbol{L}_b(\boldsymbol{\alpha}) = \mathbb{E}\left\{\partial \boldsymbol{m}_b / \partial \boldsymbol{\alpha}^T\right\} = (\boldsymbol{I}_q - \boldsymbol{K}_b \boldsymbol{Z}_b)\boldsymbol{B}_{b-1}\boldsymbol{L}_{b-1}(\boldsymbol{\alpha}) - \boldsymbol{K}_b \boldsymbol{X}_b$ and $\boldsymbol{L}_0(\boldsymbol{\alpha}) = \mathbf{0}$.

Now we derive the variability matrix. Let $\vec{\boldsymbol{X}}_b = (\boldsymbol{X}_1^T, \ldots, \boldsymbol{X}_b^T)^T$, $\vec{\boldsymbol{Y}}_b = (\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_b^T)^T$, and $\vec{\boldsymbol{Z}}_b = (\boldsymbol{Z}_1^T, \ldots, \boldsymbol{Z}_b^T)^T$. Since $\mathbb{E}(\boldsymbol{y}_j - \boldsymbol{X}_j\boldsymbol{\alpha} - \boldsymbol{Z}_j\boldsymbol{m}_j) = \mathbf{0}$, the $p \times p$ variability matrix

is given by

$$\tilde{\boldsymbol{V}}_b(\boldsymbol{\alpha}) = \vec{\boldsymbol{X}}_b^T \mathrm{var}(\vec{\boldsymbol{Y}}_b - \vec{\boldsymbol{X}}_b\boldsymbol{\alpha} - \vec{\boldsymbol{Z}}_b\vec{\boldsymbol{m}}_{\boldsymbol{b}})\vec{\boldsymbol{X}}_b, \tag{4.4}$$

where $\mathrm{var}(\vec{\boldsymbol{Y}}_b - \vec{\boldsymbol{X}}_b\boldsymbol{\alpha} - \vec{\boldsymbol{Z}}_b\vec{\boldsymbol{m}}_{\boldsymbol{b}})$ is an $N_b \times N_b$ symmetric matrix (consisting of $b \times b$ blocks) whose $(j, j+h)$-th block is

$$\begin{aligned}
&\mathrm{cov}(\boldsymbol{y}_j - \boldsymbol{X}_j\boldsymbol{\alpha} - \boldsymbol{Z}_j\boldsymbol{m}_j, \boldsymbol{y}_{j+h} - \boldsymbol{X}_{j+h}\boldsymbol{\alpha} - \boldsymbol{Z}_{j+h}\boldsymbol{m}_{j+h}) \\
=&\begin{cases} \phi\boldsymbol{I}_{n_j} + \boldsymbol{Z}_j\boldsymbol{C}_j\boldsymbol{Z}_j^T, & h = 0, \\ -\boldsymbol{Z}_j\boldsymbol{\Sigma}_b(j, j+h)\boldsymbol{Z}_{j+h}^T, & h \neq 0. \end{cases}
\end{aligned}$$

Thus,

$$\tilde{\boldsymbol{V}}_b(\boldsymbol{\alpha}) = \sum_{j=1}^{b} \boldsymbol{X}_j^T(\phi\boldsymbol{I}_{n_j} + \boldsymbol{Z}_j\boldsymbol{C}_j\boldsymbol{Z}_j^T)\boldsymbol{X}_j - 2\sum_{j=1}^{b-1}\sum_{h=1}^{b-j} \boldsymbol{X}_j^T\boldsymbol{Z}_j\boldsymbol{\Sigma}_b(j, j+h)\boldsymbol{Z}_{j+h}^T\boldsymbol{X}_{j+h},$$

where the off-diagonal blocks take the following form:

$$\begin{aligned}
\boldsymbol{\Sigma}_b(1,2) &= \rho\boldsymbol{C}_2\boldsymbol{W}_1^{-1}\boldsymbol{C}_1 = \rho\boldsymbol{C}_2\left(\rho^2\boldsymbol{C}_1 + \frac{\delta}{1-\rho^2}\boldsymbol{I}_q\right)^{-1}\boldsymbol{C}_1, \\
\boldsymbol{\Sigma}_b(1,3) &= \rho^2\boldsymbol{C}_3\boldsymbol{W}_1^{-1}\boldsymbol{C}_1\boldsymbol{W}_2^{-1}\boldsymbol{C}_2, \\
&\vdots \\
\boldsymbol{\Sigma}_b(1,b) &= \rho^{b-1}\boldsymbol{C}_b\boldsymbol{W}_1^{-1}\boldsymbol{C}_1\cdots\boldsymbol{W}_{b-1}^{-1}\boldsymbol{C}_{b-1}.
\end{aligned}$$

Therefore, we need to store all $\boldsymbol{W}_j^{-1}\boldsymbol{C}_j$ and $\boldsymbol{X}_j^T\boldsymbol{Z}_j$'s in order to calculate all the off-diagonal blocks. To reduce the data storage burden and speed up computing, we propose an approximation by considering only the correlation between adjacent data batches:

$$\tilde{\boldsymbol{V}}_b(\boldsymbol{\alpha}) \approx \sum_{j=1}^{b} \boldsymbol{X}_j^T(\phi\boldsymbol{I}_{n_j} + \boldsymbol{Z}_j\boldsymbol{C}_j\boldsymbol{Z}_j^T)\boldsymbol{X}_j - 2\sum_{j=1}^{b-1} \boldsymbol{X}_j^T\boldsymbol{Z}_j\boldsymbol{\Sigma}_b(j, j+1)\boldsymbol{Z}_{j+1}^T\boldsymbol{X}_{j+1}.$$

This approximation is legitimate because $\boldsymbol{C}_b \to \delta/(1-\rho^2)$ and $\boldsymbol{W}_b^{-1}\boldsymbol{C}_b \to 1/(1+\rho^2)$

as $b \to \infty$. Additionally, since the latent process starts with a fixed $\boldsymbol{\beta}_1$ where $\boldsymbol{C}_1$ is arbitrarily small, and for such a stationary process, all subsequent $\boldsymbol{C}_j$'s are bounded, $\boldsymbol{\Sigma}_b(1, j)$ decays with an approximate factor $\rho/(1 + \rho^2)$ which is less than 0.5.

4.5 Implementation

Apache Spark is a unified data analytics platform for large-scale data processing. Built on a distributed computing paradigm, it offers high performance for both batch and streaming data. Its Lambda architecture is designed to achieve efficient communication and coordination between batch layer and speed layer to handle streaming data. To implement our proposed MORA, we expand the speed layer in the existing Spark's Lambda architecture to accommodate inferential statistics such as sensitivity and variability matrices together with other needed quantities in the Kalman filter recursive calculation. Consequently, the resulting architecture consists of a speed layer and an inference layer responsible for the iterative calculation detailed in Section 4.3. As shown in Figure 4.2, when a new data batch D_b arrives, the inference layer calculates the matrices involved in the Kalman filter and inferential statistics. Then these quantities are sent to the speed layer to update the point estimates of $\boldsymbol{\alpha}$ and $\boldsymbol{\zeta}$. Finally, the outputs from both layers are combined to generate online regression analysis results.

Algorithm 4.3 lists the pseudo code for the implementation of online regression analysis with dynamic heterogeneity in the expanded Lambda architecture.

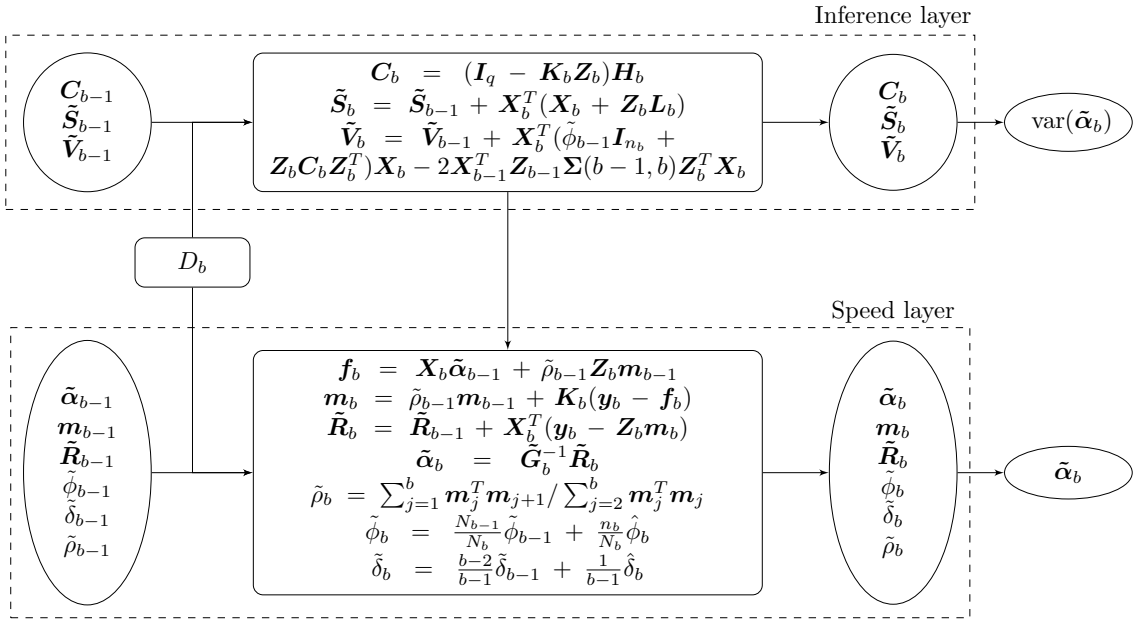

Figure 4.2:
Diagram of the expanded Lambda architecture in which $\tilde{\boldsymbol{\alpha}}_{b-1}$ and $\tilde{\boldsymbol{\zeta}}_{b-1}$ are updated to $\tilde{\boldsymbol{\alpha}}_b$ and $\tilde{\boldsymbol{\zeta}}_b$ at the speed layer, and $\tilde{\boldsymbol{S}}_{b-1}$ and $\tilde{\boldsymbol{V}}_{b-1}$ are updated to $\tilde{\boldsymbol{S}}_b$ and $\tilde{\boldsymbol{V}}_b$ at the inference layer.

Algorithm 1: Online regression analysis for heterogeneous streaming correlated data via the implementation of an expanded Lambda architecture.

Inputs: sequentially arrived datasets D_1,...,D_b,... ;

Outputs: $\tilde{\boldsymbol{\alpha}}_b$, var$(\tilde{\boldsymbol{\alpha}}_b)$, $\tilde{\phi}_b$, $\bar{\delta}_b$, $\bar{\rho}_b$, $\boldsymbol{m}_b$ and $\boldsymbol{C}_b$ for $b = 1, 2, \ldots$;

Initialize: set initials $\tilde{\boldsymbol{\alpha}}_0 = \mathbf{0}_{p\times 1}$, $\tilde{\boldsymbol{R}}_0 = \mathbf{0}_{p\times 1}$, $\tilde{\boldsymbol{G}}_0 = \tilde{\boldsymbol{S}}_0 = \tilde{\boldsymbol{V}}_0 = \mathbf{0}_{p\times p}$, $\boldsymbol{L}_0 = \mathbf{0}_{q\times p}$, $\tilde{\phi}_0 = \bar{\delta}_0 = \bar{\rho}_0 = 10^{-3}$;

for $b = 1, \ldots$ **do**

Read in dataset D_b;

At the inference layer, calculate

$\boldsymbol{H}_b = \bar{\rho}_{b-1}^2 \boldsymbol{C}_{b-1} + \bar{\delta}_{b-1}\boldsymbol{I}_q$, $\boldsymbol{Q}_b = \tilde{\phi}_{b-1}\boldsymbol{I}_{n_b} + \boldsymbol{Z}_b\boldsymbol{H}_b\boldsymbol{Z}_b^T$, $\boldsymbol{K}_b = \boldsymbol{H}_b^T\boldsymbol{Z}_b^T\boldsymbol{Q}_b^{-1}$,

$\boldsymbol{C}_b = (\boldsymbol{I}_q - \boldsymbol{K}_b\boldsymbol{Z}_b)\boldsymbol{H}_b$, $\boldsymbol{L}_b = (\boldsymbol{I}_q - \boldsymbol{K}_b\boldsymbol{Z}_b)\boldsymbol{B}_{b-1}\boldsymbol{L}_{b-1} - \boldsymbol{K}_b\boldsymbol{X}_b$,

$\tilde{\boldsymbol{S}}_b = \tilde{\boldsymbol{S}}_{b-1} + \boldsymbol{X}_b^T(\boldsymbol{X}_b + \boldsymbol{Z}_b\boldsymbol{L}_b)$,

$\boldsymbol{W}_{b-1} = \bar{\rho}_{b-1}^2\boldsymbol{C}_{b-1} + \bar{\delta}_{b-1}/(1 - \bar{\rho}_{b-1}^2)\boldsymbol{I}_q$,

$\boldsymbol{\Sigma}(b-1, b) = \bar{\rho}_{b-1}\boldsymbol{C}_b\boldsymbol{W}_{b-1}^{-1}\boldsymbol{C}_{b-1}$,

$\tilde{\boldsymbol{V}}_b = \tilde{\boldsymbol{V}}_{b-1} + \boldsymbol{X}_b^T(\tilde{\phi}_{b-1}\boldsymbol{I}_{n_b} + \boldsymbol{Z}_b\boldsymbol{C}_b\boldsymbol{Z}_b^T)\boldsymbol{X}_b - 2\boldsymbol{X}_{b-1}^T\boldsymbol{Z}_{b-1}\boldsymbol{\Sigma}(b-1, b)\boldsymbol{Z}_b^T\boldsymbol{X}_b$,

$\boldsymbol{P}_b = \boldsymbol{Z}_b\boldsymbol{C}_b\boldsymbol{Z}_b^T$ and $\boldsymbol{E}_b = \boldsymbol{C}_b + \bar{\rho}_{b-1}^2\boldsymbol{C}_{b-1} - 2\bar{\rho}_{b-1}\boldsymbol{\Sigma}(b-1, b)$;

At the speed layer, calculate

$\boldsymbol{f}_b = \boldsymbol{X}_b\tilde{\boldsymbol{\alpha}}_{b-1} + \bar{\rho}_{b-1}\boldsymbol{Z}_b\boldsymbol{m}_{b-1}$,

$\boldsymbol{m}_b = \bar{\rho}_{b-1}\boldsymbol{m}_{b-1} + \boldsymbol{K}_b(\boldsymbol{y}_b - \boldsymbol{f}_b)$,

$\tilde{\boldsymbol{R}}_b = \tilde{\boldsymbol{R}}_{b-1} + \boldsymbol{X}_b^T(\boldsymbol{y}_b - \boldsymbol{Z}_b\boldsymbol{m}_b)$ and $\tilde{\boldsymbol{G}}_b = \tilde{\boldsymbol{G}}_{b-1} + \boldsymbol{X}_b^T\boldsymbol{X}_b$,

$\tilde{\boldsymbol{\alpha}}_b = \tilde{\boldsymbol{G}}_b^{-1}\tilde{\boldsymbol{R}}_b$,

$\bar{\rho}_b = \sum_{j=1}^b \boldsymbol{m}_j^T\boldsymbol{m}_{j+1} / \sum_{j=2}^b \boldsymbol{m}_j^T\boldsymbol{m}_j$,

$\hat{\phi}_b = \frac{1}{n_b}(\boldsymbol{y}_b - \boldsymbol{X}_b\tilde{\boldsymbol{\alpha}}_b - \boldsymbol{Z}_b\boldsymbol{m}_b)^T(\boldsymbol{y}_b - \boldsymbol{X}_b\tilde{\boldsymbol{\alpha}}_b - \boldsymbol{Z}_b\boldsymbol{m}_b) - \frac{1}{n_b}\sum_{i\in D_b}\boldsymbol{P}_b(i,i)$,

$\hat{\delta}_b = \frac{1}{q}\|\boldsymbol{m}_b - \bar{\rho}_b\boldsymbol{m}_{b-1}\|^2 + \frac{1}{q}\sum_{i=1}^q \boldsymbol{E}_b(i,i)$, then update $\tilde{\phi}_b$ and $\bar{\delta}_b$.

Save $\tilde{\boldsymbol{\alpha}}_b$, $\boldsymbol{m}_b$, $\tilde{\boldsymbol{R}}_b$, $\tilde{\boldsymbol{G}}_b$, $\tilde{\phi}_b$, $\bar{\delta}_b$, $\bar{\rho}_b$ and $\boldsymbol{C}_b$, $\tilde{\boldsymbol{S}}_b$, $\tilde{\boldsymbol{V}}_b$ at the speed and inference layers, respectively;

Release dataset D_b from the memory.

end

Return $\tilde{\boldsymbol{\alpha}}_b$, var$(\tilde{\boldsymbol{\alpha}}_b) = \tilde{\boldsymbol{S}}_b^T\tilde{\boldsymbol{V}}_b^{-1}\tilde{\boldsymbol{S}}_b$, $\tilde{\phi}_b$, $\bar{\delta}_b$, $\bar{\rho}_b$, $\boldsymbol{m}_b$ and $\boldsymbol{C}_b$ for $b = 1, 2, \ldots$.

Figure 4.3: Pseudo code for the implementation of MORA.

4.6 Simulation Studies

4.6.1 Setup

We conduct simulation studies to assess the performance of the proposed online multivariate regression methods with streaming datasets. We compare our method with the naive linear regression model (LM) yielded from the R package `glm` without considering either inter-data batch correlation or heterogeneity, and oracle Kalman Estimating Equation estimator (KEE) obtained by processing the entire data once. The evaluation criteria in parameter estimation and inference in $\boldsymbol{\alpha}$ include (a) absolute bias ($\boldsymbol{\alpha}$.bias), (b) average estimated standard error ($\boldsymbol{\alpha}$.ASE), (c) empirical standard error ($\boldsymbol{\alpha}$.ESE) and (d) coverage probability ($\boldsymbol{\alpha}$.CP). We only report the absolute bias in the nuisance parameter $\boldsymbol{\zeta} = (\phi, \rho, \delta)^T$, including to (e) ϕ.bias, (f) ρ.bias and (g) δ.bias, respectively, where the latter two are included in KEE and MORA only. Computation efficiency is assessed by (h) computation time (C.Time) and (i) running time. C.Time includes time spent on both data loading time and running the algorithm while R.Time accounts only algorithm execution time.

In simulation experiments, we set a terminal point B. Consider data batch $D_b = \{\boldsymbol{y}_b, \boldsymbol{X}_b\}$ with outcome $\boldsymbol{y}_b = (y_{b1}, \ldots, y_{bn_b})^T$, covariates for fixed effects $\boldsymbol{X}_b = (\boldsymbol{x}_{b1}, \ldots, \boldsymbol{x}_{bn_b})^T$ and batch-specific covariates $\boldsymbol{Z}_b = (\boldsymbol{z}_{b1}, \ldots, \boldsymbol{z}_{bn_b})^T$. Outcomes $\boldsymbol{y}_b \mid \boldsymbol{X}_b, \boldsymbol{Z}_b$ are independently sampled from a Gaussian distribution with mean $\boldsymbol{\mu}_b = (\mu_{b1}, \ldots, \mu_{bn_b})^T$ and variance $\phi \boldsymbol{I}$ such that $\mu_{bi} = \mathbb{E}(y_{bi} \mid \boldsymbol{x}_{bi}, \boldsymbol{z}_{bi}) = \boldsymbol{x}_{bi}^T \boldsymbol{\alpha} + \boldsymbol{z}_{bi}^T \boldsymbol{\beta}_b$ and variance $v(y_{bi} \mid \boldsymbol{x}_{bi}, \boldsymbol{z}_{bi}) = \phi$. We consider a 2-dimensional vector stationary AR(1) process to characterize batch-specific heterogeneity with regression coefficients satisfying $\boldsymbol{\beta}_{b+1} = \boldsymbol{B}_b \boldsymbol{\beta}_b + \boldsymbol{\xi}_b$ where $\boldsymbol{B}_b = \text{diag}(\rho_1, \rho_2)$ is the transition matrix with the respective autocorrelation coefficients ρ_1 and ρ_2, and the noise $\boldsymbol{\xi}_b \overset{iid}{\sim} \mathcal{N}_2(\boldsymbol{0}, \boldsymbol{\delta})$, $b = 1, \ldots, B$.

We choose the true regression coefficient parameter by generating $\boldsymbol{\alpha}_0 \sim \mathcal{N}_5(\boldsymbol{0}, \boldsymbol{I}_5)$

where $\boldsymbol{I}_5$ is the 5×5 identity matrix. Set initial value for the dynamic coefficients $\boldsymbol{\beta}_1 = \mathbf{0}$. Covariates are independently sampled from $\boldsymbol{x}_i \overset{iid}{\sim} \mathcal{N}_5(\mathbf{0}, \boldsymbol{V}_5)$ $(i = 1, \ldots, N_b)$, where $\boldsymbol{V}_5$ is a 5×5 compound symmetry covariance matrix with correlation $\rho_x = 0.5$. The variance parameters of the two covariance matrices are set as $\phi = 1$ and $\delta = 1$. As far as the online procedure concerns, we only consider the correlation between adjacent data batches. Thus, we examine the performances under different correlation coefficients $\rho_1 = 0.1$, 0.5 and 0.9 while ρ_2 is fixed at 0.5.

4.6.2 Evaluation of Parameter Estimation

Scenario 1: fixed N_B but varying batch size n_b

We begin with evaluating effect of data batch size n_b on the performance of our proposed MORA in parameter estimation and computation efficiency. There are B data batches, each with data batch size n_b, and the total sample size is $N_B = |D_b^\star| = 10,000$. They are generated by data batches from a linear state space mixed model specified in Section 4.6.1. Table 4.1 reports the evaluation criteria over 500 replications.

Bias and coverage probability in $\boldsymbol{\alpha}$. In both offline KEE and our proposed MORA as shown in Table 4.1, it is easy to see that their estimation bias and coverage probability are very close to each other, and neither of them changes with varying batch sample size n_b. This confirms the theoretical results given in Theorem IV.1. In other words, the statistical inference by MORA depends only on the cumulative sample size N_B. However, in the LM where outcomes are treated as independent ones, $\boldsymbol{\alpha}$.bias is larger than that in KEE or MORA due to the loss of statistical efficiency. The coverage probability in LM is still around 95% because `glm` in `R` uses the iteratively weighted least squares where the extra variability is accounted by the empirical weighting matrix. Additionally, considering correlation between only

adjacent data batches has marginal effects on the inference performance on $\boldsymbol{\alpha}$: the coverage probabilities are close to the nominal 95% level under different ρ_1.

Bias in $\boldsymbol{\zeta}$. Similar as estimation performance in $\boldsymbol{\alpha}$, MORA provides similar level of estimation bias and empirical standard errors in estimating $\boldsymbol{\zeta}$ as the offline KEE. However, comparing to MORA, ϕ.bias is much larger with LM, because the dynamic heterogeneity is accounted by this variance parameter. It also explains why ϕ.bias increases with B, namely an increased level of heterogeneity. We also find that the latent stationary AR(1) process get stabilized with larger B, and therefore gradually reduces both bias and variability in both ρ and δ. Additionally, if we compare across the three panels in Table 4.1, we can also see that in LM, as ρ increases, ϕ.bias becomes larger while this is not the case with either KEE or MORA. This is because the increased correlation is ignored by LM. In MORA, only δ.bias increases with ρ_1 due to the increased variability of the AR(1) process.

Computation time. Computation efficiency is assessed by "C.Time" and "R.Time" in Table 4.1 which refer to the total amount of time and algorithm execution respectively. As expected, MORA is more efficient than KEE while providing similar statistical performance. Besides, while maintaining the similar bias and coverage probability, MORA is more efficient to process data with a relatively smaller data batch size n_b.

Table 4.1: Simulation results under the linear state space mixed model are summarized over 500 replications, with fixed $N_B = 10,000$ and $p = 5$ with varying batch sizes n_b.

	$\rho_1 = 0.1$, $\rho_2 = 0.5$								
$B \times n_b$	5×2000			50×200			500×20		
	LM	KEE	MORA	LM	KEE	MORA	LM	KEE	MORA

$\boldsymbol{\alpha}$.bias$\times 10^{-3}$	16.49	10.56	10.96	18.84	10.45	10.55	18.66	10.73	10.76
$\boldsymbol{\alpha}$.ASE$\times 10^{-3}$	20.68	12.92	13.66	23.31	12.97	13.19	23.58	13.58	13.64
$\boldsymbol{\alpha}$.ESE$\times 10^{-3}$	21.14	13.25	13.81	23.51	12.99	13.19	23.38	13.36	13.38
$\boldsymbol{\alpha}$.CP	0.946	0.945	0.947	0.953	0.953	0.953	0.950	0.952	0.951
ϕ.bias	1.645	0.012	0.106	2.271	0.021	0.022	2.333	0.176	0.175
ϕ.ESE	0.999	0.014	0.230	0.405	0.014	0.035	0.130	0.012	0.014
ρ_1.bias		0.310	0.337		0.109	0.108		0.037	0.037
ρ_1.ESE		0.319	0.325		0.135	0.134		0.046	0.046
ρ_2.bias		0.226	0.226		0.101	0.101		0.031	0.031
ρ_2.ESE		0.273	0.274		0.127	0.128		0.040	0.039
δ.bias		0.497	0.512		0.117	0.145		0.041	0.047
δ.ESE		0.803	0.672		0.148	0.186		0.047	0.051
C.Time(s)	0.044	44.91	16.47	0.087	1.970	0.573	0.447	1.865	0.488
R.Time(s)	0.028	44.90	16.46	0.041	1.925	0.539	0.038	1.456	0.311

$\rho_1 = 0.5,\ \rho_2 = 0.5$

$B \times n_b$	5×2000			50×200			500×20		
	LM	KEE	MORA	LM	KEE	MORA	LM	KEE	MORA
$\boldsymbol{\alpha}$.bias$\times 10^{-3}$	16.29	10.55	10.95	19.47	10.45	10.54	19.80	10.73	10.76
$\boldsymbol{\alpha}$.ASE$\times 10^{-3}$	20.50	12.92	13.67	24.24	12.98	13.20	24.66	13.58	13.64
$\boldsymbol{\alpha}$.ESE$\times 10^{-3}$	20.98	13.25	13.81	24.39	12.99	13.19	24.76	13.36	13.38
$\boldsymbol{\alpha}$.CP	0.946	0.946	0.947	0.952	0.953	0.953	0.948	0.952	0.951
ϕ.bias	1.599	0.011	0.106	2.537	0.020	0.022	2.648	0.176	0.175
ϕ.ESE	0.976	0.013	0.231	0.470	0.013	0.035	0.157	0.012	0.014
ρ_1.bias		0.232	0.225		0.092	0.096		0.031	0.031
ρ_1.ESE		0.259	0.280		0.113	0.120		0.039	0.039
ρ_2.bias		0.234	0.234		0.092	0.102		0.031	0.031

ρ_2.ESE		0.285	0.286		0.131	0.130		0.040	0.039
δ.bias		0.600	0.612		0.119	0.176		0.047	0.060
δ.ESE		0.753	0.750		0.155	0.216		0.047	0.053
C.Time(s)	0.059	43.13	17.12	0.103	2.162	0.567	0.513	2.113	0.547
R.Time(s)	0.042	43.12	17.10	0.046	2.104	2.104	0.042	1.642	0.356

$\rho_1 = 0.9$, $\rho_2 = 0.5$

$B \times n_b$	5×2000			50×200			500×20		
	LM	KEE	MORA	LM	KEE	MORA	LM	KEE	MORA
$\boldsymbol{\alpha}$.bias$\times 10^{-3}$	16.55	10.55	10.94	24.03	10.45	10.55	28.20	10.74	10.77
$\boldsymbol{\alpha}$.ASE$\times 10^{-3}$	20.86	12.92	13.67	30.41	12.97	13.20	34.86	13.58	13.66
$\boldsymbol{\alpha}$.ESE$\times 10^{-3}$	21.38	13.20	13.81	30.61	12.99	13.19	35.31	13.36	13.39
$\boldsymbol{\alpha}$.CP	0.948	0.946	0.946	0.951	0.953	0.953	0.944	0.952	0.952
ϕ.bias	1.696	0.011	0.108	4.675	0.020	0.023	6.316	0.176	0.176
ϕ.ESE	1.029	0.013	0.239	1.845	0.013	0.038	0.963	0.012	0.015
ρ_1.bias		0.394	0.366		0.070	0.072		0.017	0.017
ρ_1.ESE		0.259	0.254		0.075	0.081		0.020	0.020
ρ_2.bias		0.267	0.257		0.106	0.105		0.031	0.032
ρ_2.ESE		0.322	0.315		0.135	0.132		0.040	0.040
δ.bias		0.992	0.793		0.150	0.564		0.071	0.151
δ.ESE		1.636	1.279		0.198	0.675		0.050	0.084
C.Time(s)	0.060	43.85	16.39	0.096	2.491	0.637	0.447	1.874	0.497
R.Time(s)	0.043	43.83	16.10	0.042	2.437	0.604	0.037	1.464	0.316

Scenario 2: fixed data batch size n_b but increasing number of data batches B

Now we consider a scenario where a sequence of data batches arriving with high speed. For convenience, we fix data batch size $n_b = 100$ but let B increase from 10 to 1000. Table 4.2 summarizes the simulation results under the same model specified in Section 4.6.1.

Bias and coverage probability in $\boldsymbol{\alpha}$. Similar to what we observed before, MORA preserves similar level of bias and coverage probability as KEE: as number of data batches B increases from 10 to 1000, $\boldsymbol{\alpha}$.bias decreases at an empirical rate of $O(\sqrt{N_B})$ which further confirms the large sample property in Theorem IV.1. The coverage probability stays robustly around 95%. Similar to scenario 1, estimation bias and coverage probability in $\boldsymbol{\alpha}$ in MORA remain robust across different ρ_1, but larger ρ_1 leads to slightly larger bias in LM due to the ignorance of dependence.

Bias in $\boldsymbol{\zeta}$. First, by looking at Table 4.2 horizontally, we can easily find that in both KEE and MORA, ρ.bias and δ.bias decrease consistently with increasing B, while ϕ.bias first decreases and then gets stabilized. But in LM, ϕ.bias increases with B which further confirms what we find in scenario 1. Then if we compare the results vertically, ϕ.bias and ρ.bias in both KEE and MORA stay robust against different ρ_1, while LM again shows an increased ϕ.bias.

Computation time. With a fixed data batch size n_b, both C.Time and R.Time in MORA increase linearly with B. When B is small, LM takes less C.Time than MORA, but it becomes reversed once B reaches 1000 due to the large data loading time. It is worth noting that both C.Time and R.Time in offline KEE are almost 10 times the MORA. This further demonstrates the strong computation advantage in MORA especially when a large sample size has been accumulated over time.

Table 4.2: Simulation results under the linear state space mixed model are summarized over 500 replications, with fixed $n_b = 100$ and $p = 5$ with B increased from 10 to $1,000$.

	$\rho_1 = 0.1$, $\rho_2 = 0.5$, $n_b = 100$								
B	10			100			1,000		
	LM	KEE	MORA	LM	KEE	MORA	LM	KEE	MORA
$\boldsymbol{\alpha}$.bias$\times 10^{-3}$	56.22	34.00	34.59	18.48	10.50	10.55	5.91	3.32	3.32
$\boldsymbol{\alpha}$.ASE$\times 10^{-3}$	70.13	41.26	43.23	23.46	13.05	13.18	7.46	4.12	4.13
$\boldsymbol{\alpha}$.ESE$\times 10^{-3}$	70.76	42.70	43.83	23.20	13.06	13.12	7.41	4.16	4.17
$\boldsymbol{\alpha}$.CP	0.947	0.944	0.949	0.955	0.951	0.952	0.949	0.948	0.948
ϕ.bias	1.985	0.050	0.071	2.305	0.038	0.035	2.343	0.039	0.039
ϕ.ESE	0.813	0.045	0.122	0.288	0.015	0.018	0.090	0.005	0.005
ρ_1.bias		0.210	0.218		0.084	0.084		0.025	0.025
ρ_1.ESE		0.260	0.266		0.103	0.103		0.031	0.031
ρ_2.bias		0.206	0.207		0.072	0.072		0.025	0.021
ρ_2.ESE		0.251	0.256		0.090	0.090		0.027	0.026
δ.bias		0.357	0.412		0.099	0.099		0.026	0.027
δ.ESE		0.550	0.588		0.103	0.124		0.033	0.034
C.Time(s)	0.007	0.076	0.027	0.071	0.621	0.182	5.895	17.63	2.68
R.Time(s)	0.004	0.073	0.022	0.023	0.573	0.153	0.383	12.12	2.26
	$\rho_1 = 0.5$, $\rho_2 = 0.5$, $n_b = 100$								
B	10			100			1,000		
	LM	KEE	MORA	LM	KEE	MORA	LM	KEE	MORA
$\boldsymbol{\alpha}$.bias$\times 10^{-3}$	56.80	34.00	34.55	19.26	10.50	10.56	6.16	3.32	3.32
$\boldsymbol{\alpha}$.ASE$\times 10^{-3}$	70.80	41.26	43.26	24.47	13.05	13.19	7.81	4.12	4.13

$\boldsymbol{\alpha}$.ESE$\times 10^{-3}$	71.77	42.69	43.76	24.16	13.06	13.12	7.74	4.16	4.17
$\boldsymbol{\alpha}$.CP	0.947	0.944	0.950	0.960	0.951	0.952	0.952	0.948	0.948
ϕ.bias	2.047	0.050	0.072	2.598	0.038	0.035	2.664	0.039	0.039
ϕ.ESE	0.881	0.045	0.127	0.344	0.015	0.018	0.113	0.005	0.005
ρ_1.bias		0.204	0.202		0.073	0.073		0.022	0.022
ρ_1.ESE		0.242	0.243		0.092	0.092		0.028	0.028
ρ_2.bias		0.212	0.215		0.073	0.074		0.022	0.021
ρ_2.ESE		0.255	0.263		0.091	0.092		0.027	0.026
δ.bias		0.369	0.496		0.082	0.115		0.026	0.029
δ.ESE		0.547	0.740		0.104	0.136		0.033	0.034
C.Time(s)	0.011	0.113	0.038	0.120	1.065	0.289	4.118	16.51	2.663
R.Time(s)	0.006	0.108	0.032	0.038	0.983	0.242	0.393	11.48	2.245

$\rho_1 = 0.9$, $\rho_2 = 0.5$, $n_b = 100$

B		10			100			1,000	
	LM	KEE	MORA	LM	KEE	MORA	LM	KEE	MORA
$\boldsymbol{\alpha}$.bias$\times 10^{-3}$	60.08	34.00	34.55	25.31	10.50	10.56	8.73	3.32	3.32
$\boldsymbol{\alpha}$.ASE$\times 10^{-3}$	74.68	41.26	43.34	32.46	13.05	13.19	11.18	4.12	4.13
$\boldsymbol{\alpha}$.ESE$\times 10^{-3}$	76.37	42.70	43.73	31.95	13.06	13.12	11.01	4.16	4.17
$\boldsymbol{\alpha}$.CP	0.949	0.944	0.951	0.960	0.951	0.952	0.949	0.948	0.948
ϕ.bias	2.425	0.050	0.076	5.417	0.038	0.035	6.516	0.039	0.039
ϕ.ESE	1.286	0.045	0.150	1.684	0.015	0.020	0.743	0.005	0.005
ρ_1.bias		0.278	0.273		0.044	0.044		0.012	0.012
ρ_1.ESE		0.216	0.217		0.054	0.054		0.015	0.014
ρ_2.bias		0.225	0.226		0.074	0.075		0.021	0.021
ρ_2.ESE		0.257	0.265		0.091	0.091		0.026	0.026
δ.bias		0.770	1.226		0.094	0.340		0.028	0.063

δ.ESE		1.631	2.229		0.113	0.351		0.033	0.049
C.Time(s)	0.013	0.131	0.041	0.111	1.019	0.283	5.799	17.78	3.069
R.Time(s)	0.007	0.125	0.034	0.035	0.943	0.239	0.441	12.42	2.595

4.7 SRTR Data Example

In the analysis of the kidney transplant data collected by Scientific Registry of Transplant Recipients (SRTR), we aim to evaluate the effects of some key risk factors on the outcome of serum creatinine level one-year post-transplantation. Many studies have unveiled that post-transplant renal function in the first year is highly related to long-term kidney transplant survival (*Sundaram et al.*, 2002). We consider the scenario where the transplant data batches arrive yearly during the period of 31 years from 1987 to 2017, with $B = 29$ and $N_B = 221,337$ recipient creatinine level at the first post-transplant year with no missing data. Note that these are the years of transplantation and serum creatinine levels are obtained one year later, namely from 1988 to 2018. There are few records prior to 1989, so we combine the data from 1987 to 1989 to form the first data batch. We are interested in assessing population average effects of the risk factors on the serum creatinine one year after transplantation, adjusting for dynamic batch-specific heterogeneous effects.

We take a logarithm transformation of the serum creatinine level so the resulting log-transformed variable is approximately normal, which is analyzed using the proposed linear mixed state space model with the following set of fixed risk factors: "donor's and recipient's age" (standardized), "donor-recipient sex" (1 for homosexual pair and 0 otherwise), "donor's and recipient's BMI" (1 for obese and 0 for no obese), "donor-recipient height ratio" (1 for greater than 1 and 0 otherwise), "donor-recipient weight ratio" (1 for greater than 0.9 and 0 otherwise), "donor-recipient race" (1 for

homoracial pair and 0 otherwise), and "duration of dialysis" (0 for less than 3 years and 1 otherwise). A preliminary analysis where we fit a cross-sectional linear regression model to yearly single data batch separately is shown in Figure 4.4. Based on the autocorrelation (ACF) and partial correlation (PACF) plots in Figure 4.5, we choose "time (in year)" and "donor age" as dynamic batch-specific effects that follow an AR(1) stationary process to account for the underlying heterogeneity over the sequence of data batches. Such an analysis work can hardly be done via the offline KEE method due to the intensive computation burden incurred by both large data batch size n_b and cumulative sample size N_B. Therefore, we apply our proposed MORA method to sequentially update parameter estimates and standard errors.

Table 4.3 reports the results from fitting a linear state space mixed model using our proposed MORA, at the terminal year 2017. Due to the large cumulative sample size in this streaming data setting, all p-values are become too small to be useful for making conclusions. Thus, we focus on point estimates, standard errors and z-values in this table. The major findings are: (i) "donor-recipient height ratio" and "donor-recipient weight ratio" are the top two risk factors among others. Such an association between donor-recipient weight mismatch (donor<recipient) and graft failure has also been found by *Miller et al.* (2017); *Tillmann et al.* (2019); (ii) additionally, recipients with younger age and matched race transplant are associated with better graft function; (iii) deceased donor, higher recipient's or donor's BMI, homosexual transplantation and longer duration of dialysis may have negative effects on post-transplant renal function. This may provide health practitioners some insights on how to correctly analyze these type of cumulative EHR data while accounting for dynamics and dependence. Additionally, the dynamic changes in "time effect" and "donor age" are also shown in Figure 4.6. It is clear that baseline serum creatinine decreases from year 1987 to 2003 before get stabilized, and donor age also shows a slow decreasing trend. These might be related to the FDA's approval of immunosup-

pressive drugs such as CellCept in 1995 and Tacrolimus in 1997 to be used in kidney transplantation.

Table 4.3: Results from fitting a linear state space mixed model with our proposed MORA method, at the end of year 2017. The total sample size is $N_B = 221,337$, $p = 9$, $q = 2$, $B = 29$.

	Estimate	Std.Err$\times 10^{-3}$	z-value
Recipient's age	-0.015	0.858	-17.55
Donor-recipient sex	0.070	5.010	14.03
Recipient's BMI	0.050	3.622	13.79
Donor's BMI	0.020	1.941	10.56
Donor-recipient height ratio	-0.118	4.012	-29.48
Donor-recipient weight ratio	-0.083	4.619	-17.99
Donor-recipient race	-0.041	4.826	-8.501
Donor type	0.058	6.228	9.294
Duration of dialysis	0.025	2.426	10.41
ϕ	0.111		
ρ	0.923, 0.988		
δ	0.008		

Figure 4.7 shows the trajectories of $-\log_{10}(p)$ values over 31 years, the 10-base log p-values of the z-test for each of the regression coefficient equals to 0. Among all these risk factors, "donor-recipient height ratio" turns out to have the largest effect. To characterize the overall significance level for each covariate over the 31-year period, we calculate a summary statistic as of the area under the p-value curve. For most of these curves, the ranking of overall significance by these areas is well aligned with the ranking of p-values obtained at the terminal year 2017, except for "recipient age" and "donor-recipient weight ratio" that crossover at around year 2014. This also happens to "donor-recipient race" and "donor type". By looking into these trajectories rather than only the end-point p-values, we can see that recipient's age has an overall higher significant association with post-transplant renal function than weight ratio. This summary statistic provides useful evidence besides the terminal p-values.

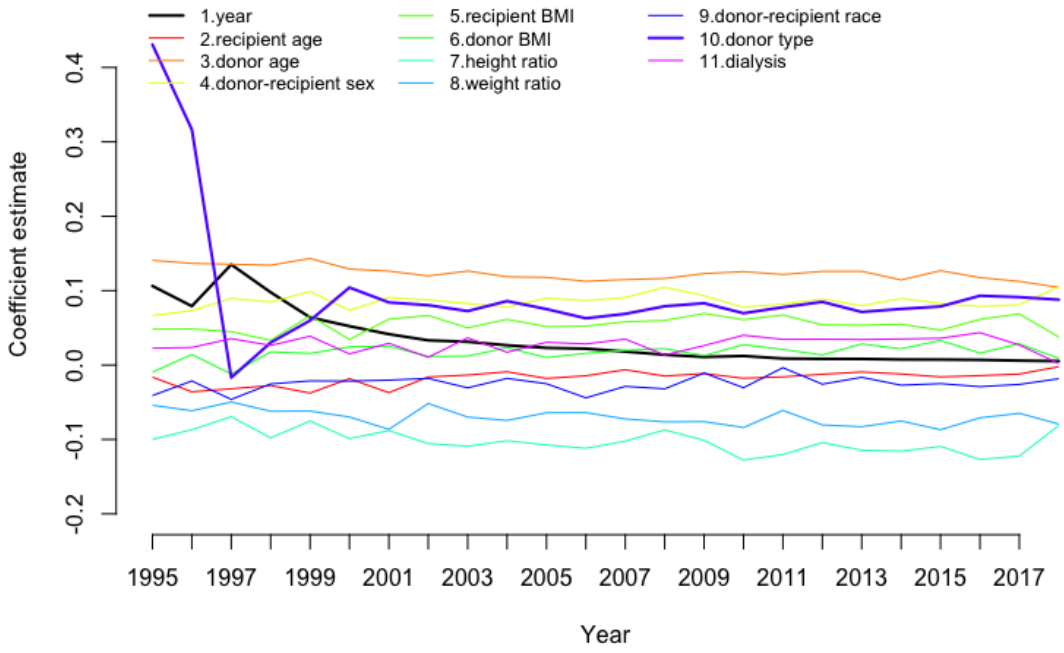

Figure 4.4: Preliminary cross-sectional analysis results showing the trends of individual regression coefficient estimates by fitting the linear regression model to each single yearly data batch, respectively.

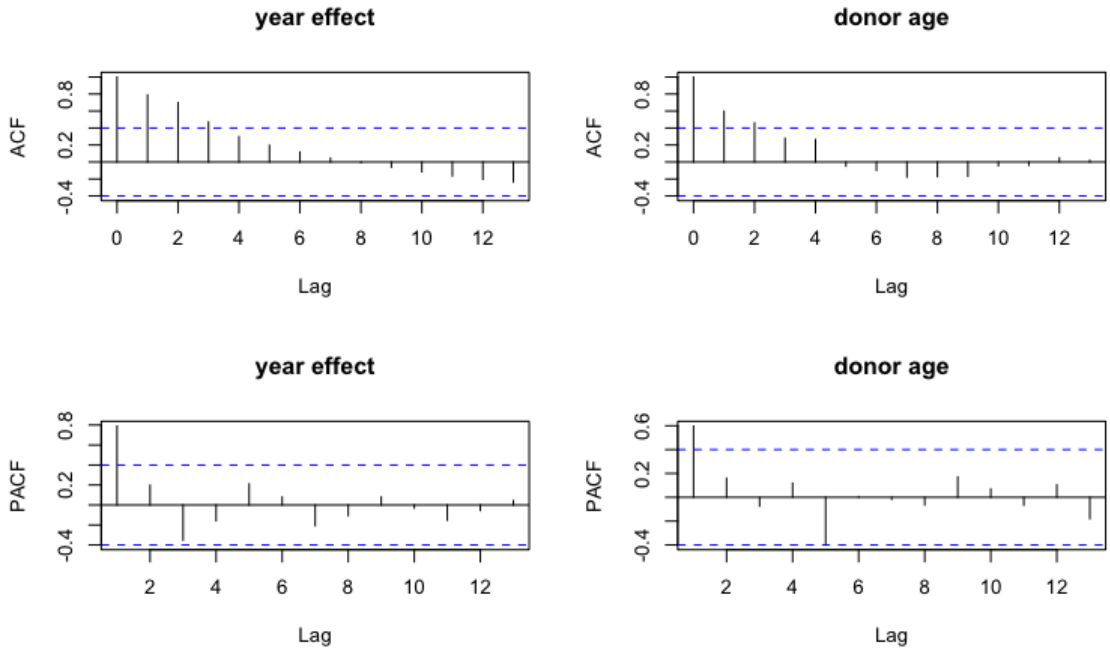

Figure 4.5: Empirical ACF and PACF plots of regression coefficient estimates from preliminary analysis. It is clear that risk factors "year effect" and "donor age" follow a stationary AR(1) process.

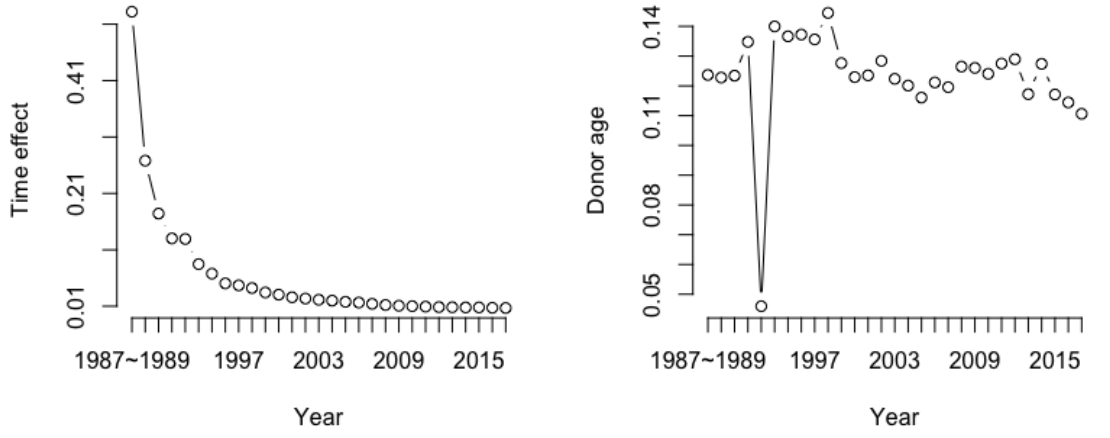

Figure 4.6: Trace plots of the dynamic effects of the "time effect" and "donor age" over 31 years period.

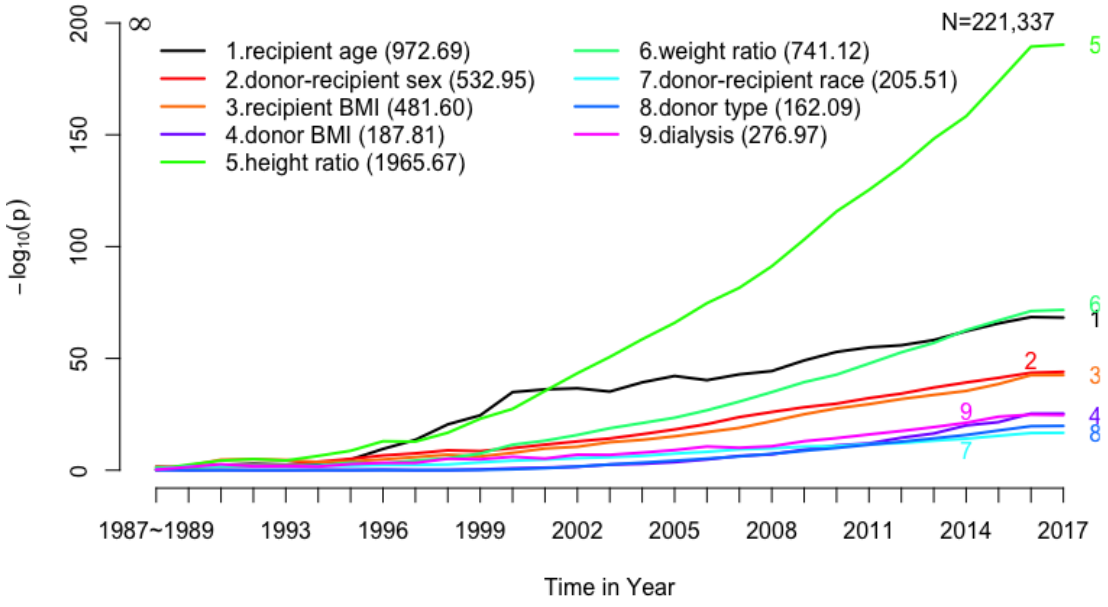

Figure 4.7: Trajectories of $-\log_{10}(p)$ over yearly data batches from 1987 to 2017, each for one risk factor. Numbers on the left y-axis are the negative logarithm p-values obtained by z-test and labels on the x-axis correspond to the end of each year. The values in the brackets next to covariate names denote respective areas under the p-value curves.

4.8 Concluding Remarks

As streaming data becomes one of the most pervasive data collection scheme in the field of Data Science, there is a surge increase in the number of applications that require real-time processing of massive data with high velocity. Conventional offline techniques suffer from many limitations when being applied to streaming data analytics tasks. Online learning technique is a promising area under exploration to tackle the emerging challenges of mining data streams. The history of sequential processing may date back to 1950s when *Robbins and Monro* (1951) proposed theory of stochastic approximation, and a variety of online learning methods such as stochastic gradient descent algorithm are developed thereafter (*Sakrison*, 1965; *Duchi et al.*, 2011; *Toulis and Airoldi*, 2015). However, there are two major issues that are not fully addressed by the methods along this line of research: (i) most of them are motivated by applications in the field of engineering where point estimation or prediction rather than statistical inference is of the main focus. As opposed to needs in the biomedical research; and (ii) there is only a fixed common parameter in model specification so that the dynamic heterogeneity over the data streams cannot be dealt with. As shown in the marginal LM analysis, ignoring the sequential heterogeneity may lead to large estimation bias and low statistical efficiency.

The first gap has been filled by (*Luo and Song*, 2020) and the second one is the main focus of this chapter. To account for dynamic heterogeneity, we propose a new framework of linear state space models, in which the dynamically changed regression coefficients are governed by an AR(1) process. The main idea of the estimation is rooted in the EM algorithm where in the E-step, the conditional mean of the batch-specific effect is calculated using the Kalman recursive technique, and in the M-step, summary statistics rather than historical subject-level data are used to facilitate the efficiency of online regression analysis. Both proposed statistical methodology and computational algorithms have been investigated for their theoretical guarantees and

examined numerically via extensive simulation studies. The proposed online multivariate regression analysis (MORA) method with heterogeneity is computationally more efficient with smaller data batch size with no loss of statistical efficiency.

A direction worth a further exploration is to consider the case of a non-stationary latent process such as random walks. One of the challenges pertains to the inter-data batch correlation that does not decay over the sequence of data batches which is beyond the ϕ-mixing process used in this paper to establish large sample properties. A related problem of interest is to test the stationarity of the underlying latent process, i.e. $H_0 : \rho = 1$ versus $H_1 : 0 < \rho < 1$ where ρ is the autocorrelation parameter. Besides, in this paper, we start with the linear state space model with Gaussian outcomes. This framework may be relaxed to non-Gaussian responses to analyze other real-world time series data. For example, in biomedical field such as data streams captured by wearable devices, data may be discrete such activity counts or binary variables, or physiological measurements that are highly skewed such as body temperature. Therefore, some extensions to handle non-normal time series is an important future research area as part of new analytic tools for high frequency mobile health data.

CHAPTER V

Summary and Future Work

Motivated by the streaming data collection scheme where data arrives sequentially and perpetually overtime, this dissertation has focused on transforming important offline statistical methods of parameter estimation and inference to improve online Big Data analytics. Under the streaming data setting, several of the main challenges include data storage, re-computation and dynamic heterogeneity in data streams. Traditional offline methods may no longer be suitable in these scenarios due to the computation burden and latency in providing real-time analytic results. The main idea in my proposed methods is rooted in the use of current data and summary statistics of historical data, instead of any historical subject-level data to achieve real-time estimation and inference. It forms the technical core of Chapter II. Besides, biomedical data may involve correlated clusters, and this has motivated me to carry out an extension concerning the development of an incremental inference method via quadratic inference functions in Chapter III. Furthermore, data streams collected over time may have inter-data batch correlation and heterogeneity. To address the dynamics in regression coefficients, I utilized the state space model framework and developed an online multivariate regression analysis method in Chapter IV. The three methods in this dissertation respectively addressed different statistical challenges, which includes: data storage and re-computation, correlation and outlier detection,

and dynamic heterogeneity. Each of the methods can be further extended and improved along in their own framework and settings as have been discussed in each of the chapters. It is of great interest to further transform offline new methodologies developed by others to further generalize the streaming data analytical toolbox, and construct a standard framework that is applicable to a broader range of problems pertaining to modern streaming data features. Therefore, we conclude this chapter by pointing out potential directions for future research of modeling for streaming data.

One promising direction is to work on the quantile regression (QR) model where estimating functions are non-differentiable. Smoothing is one of the techniques adopted by a large body of literature on estimation and inference for QR (*Koenker*, 2005; *Chen et al.*, 2018). The method of smoothing the non-smooth QR objective function dates back to *Horowitz* (1998) where he used the bootstrap method to obtain asymptotic refinements. Since the smoothing technique overcomes the difficulty in higher-order expansion of the estimating equation, it will help to build a connection to the methods developed for the smooth objective functions in Chapters II and III. An immediate area of application is obesity research, such as my collaborative project Black Women Wellness Project (BWWP), where BMI quantiles are used in clinical studies to define overweight and obesity.

Driven by the recent initiatives of analyzing data collected by wearable devices, another promising direction and natural extension of my dissertation work is to develop novel statistical methods with emphasis on mobile health analytics. I am greatly interested in building dynamic statistical models for real-time regression analysis of mobile health data streams. From my third dissertation project, I have seen a few promising extensions: (1) using Generalized State Space Models for high-frequency non-normal time series data; (2) a non-stationary latent process such as random walk.

Another direction concerns the problem of online change-point detection, which

is critical in tracking outliers and systematic shifts. This topic is not only of great importance in health data analysis, but also in high-tech companies such as Google. According to my internship experience, tracking data such as App clicks or downloads are pervasive in experiment monitoring platforms, and algorithms that can detect abnormal change-points in a timely fashion are of great need in industry. I plan to extend the content in Chapter II with the mixture sequential probability ratio test (mSPRT) (*Johari et al.*, 2016) to develop a real-time estimation and inference method with change-point detection.

I view collaboration as an imperative channel to apply statistics to solve real-world problems, and more importantly to generate new ideas on methodology development. Motivated by my background in biology and experiences in applications, I plan to collaborate with the Department of Biostatistics at the University of Iowa on applied projects, especially those related to human diseases, genetics, randomized controlled trials, and mobile health data. Besides seeking collaborations in academia, I am also passionate about building connections with industry to work on cutting-edge projects in data science.

APPENDICES

APPENDIX A

Appendices for Chapter II

Table A.1: In the column **Method**, "SGD" includes both first-order procedures and second-order procedures that are based only on the diagonal elements of an approximated Hessian matrix, not on the full estimated Hessian. In the column **Hessian matrix**, "Full" indicates whether the full $p \times p$ (approximated) Hessian matrix is used in an algorithm; "Exact" indicates whether the Hessian matrix is approximated or obtained by the second-order derivative of the log-likelihood function (i.e. no approximation). In the column **Inference**, "Yes" means the availability of statistical inference. See more details in the Appendix below.

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

Table A.2: Summary of notations. "SubH." corresponds to the negative Hessian matrix for a single data batch D_b, and "AggH." denotes the aggregated negative Hessian. $\hat{\boldsymbol{\beta}}_b$ (appears only in CEE) denotes the estimator for a single data batch D_b while $\check{\boldsymbol{\beta}}_b$ (used only in CUEE) is an intermediary estimator similar to the CEE estimator.

Method	Estimator	SubH.	AggH.	Variance
Oracle MLE	$\hat{\boldsymbol{\beta}}_b^\star$	-	-	$\hat{\boldsymbol{V}}_b^\star$
AI-SGD	$\boldsymbol{\beta}_{N_b}^{\text{aim}}$	-	-	-
OLSE	$\tilde{\boldsymbol{\beta}}_b^{\text{olse}}$	$\boldsymbol{X}_b^T\boldsymbol{X}_b$	$\sum_{j=1}^b \boldsymbol{X}_j^T\boldsymbol{X}_j$	$CMSE_b \times \left(\sum_{j=1}^b \boldsymbol{X}_j^T\boldsymbol{X}_j\right)$
CEE	$\tilde{\boldsymbol{\beta}}_b^{\text{cee}}$	$\boldsymbol{A}_b^{\text{cee}}$	$\tilde{\boldsymbol{A}}_b^{\text{cee}}$	$\tilde{\boldsymbol{V}}_b^{\text{cee}}$
CUEE	$\tilde{\boldsymbol{\beta}}_b^{\text{cuee}}$	$\boldsymbol{A}_b^{\text{cuee}}$	$\tilde{\boldsymbol{A}}_b^{\text{cuee}}$	$\tilde{\boldsymbol{V}}_b^{\text{cuee}}$
Renewable	$\tilde{\boldsymbol{\beta}}_b$	$\boldsymbol{J}_b$	$\tilde{\boldsymbol{J}}_b$	$\tilde{\phi}_b\tilde{\boldsymbol{J}}_b^{-1}$

A.1 Chapter II: Proof of Consistency

Proof. Assume the conditions (C1)-(C3) given in Section II.3 hold. The MLE of the cumulative dataset to time point b is $\hat{\boldsymbol{\beta}}_b^\star = \arg\max_{\boldsymbol{\beta}\in\mathbb{R}^p} \ell_{N_b}(\boldsymbol{\beta}, \phi; D_b^\star)$. Under the condition $(C2)$, *i.e.*, $\boldsymbol{\mathcal{I}}_{N_b}(\boldsymbol{\beta})$ is positive-definite, there exists a unique solution to the unit score equation $\sum_{j=1}^b \boldsymbol{U}_j(D_j;\boldsymbol{\beta}) = 0$, which is the MLE $\hat{\boldsymbol{\beta}}_b^\star$ for this cumulative dataset.

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

Define a function $f_b(\boldsymbol{\beta}) = -\frac{1}{N_b}\sum_{j=1}^{b-1} \boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}_{b-1}) + \frac{1}{N_b}\boldsymbol{U}_b(D_b;\boldsymbol{\beta})$. According to equation (2.15), the renewable estimator $\tilde{\boldsymbol{\beta}}_b$ satisfies

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

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

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

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

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

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

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

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

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

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

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

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

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

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

definite. It follows that

$$\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0 \xrightarrow{p} \mathbf{0}, \quad N_b \to \infty.$$

□

A.2 Chapter II: Proof of Asymptotic Normality

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

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

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

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

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

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

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

It follows that

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

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

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

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

$$\frac{1}{N_b}\sum_{j=1}^{b}\boldsymbol{U}_j(D_j;\boldsymbol{\beta}_0)-\frac{1}{N_b}\left\{\sum_{j=1}^{b-1}\boldsymbol{J}_j(D_j;\tilde{\boldsymbol{\beta}}_j)+\boldsymbol{J}_b(D_b;\boldsymbol{\beta}_0)\right\}(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)+O_p\left(\sum_{j=1}^{b}\frac{n_j}{N_b}\|\tilde{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_0\|^2\right)=\mathbf{0}. \tag{A.12}$$

Since according to Theorem II.3, all $\tilde{\boldsymbol{\beta}}_j$ are consistent for $j=1,...,b-1$, and by condition (C3), the Continuous Mapping Theorem implies that

$$\frac{1}{N_b}\sum_{j=1}^{b}\boldsymbol{U}_j(D_j;\boldsymbol{\beta}_0)-\frac{1}{N_b}\sum_{j=1}^{b}\boldsymbol{J}_j(D_j;\boldsymbol{\beta}_0)(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)+O_p\left(\sum_{j=1}^{b}\frac{n_j}{N_b}\|\tilde{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_0\|^2\right)=\mathbf{0}. \tag{A.13}$$

Furthermore, since $\tilde{\phi}_b$ is a consistent estimator of ϕ_0 due to the weak law of large numbers (WLLN), we have

$$\frac{1}{N_b}\tilde{\phi}_b^{-1}\sum_{j=1}^{b}\boldsymbol{J}_j(D_j;\boldsymbol{\beta}_0)\xrightarrow{p}\boldsymbol{\Sigma}_0^{-1},\ N_b\to\infty.$$

By condition (C2), $\boldsymbol{\mathcal{I}}_{N_b}^{-1}(\boldsymbol{\beta}_0)$ exists, and thus the Central Limit Theorem implies

$$\sqrt{N_b}(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0) = \left\{\sum_{j=1}^{b} \boldsymbol{J}_j(D_j;\boldsymbol{\beta}_0)\right\}^{-1} \frac{1}{\sqrt{N_b}}\sum_{j=1}^{b} \boldsymbol{U}_j(D_j;\boldsymbol{\beta}_0)+o_p(1) \xrightarrow{d} \mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_0),\ N_b\to\infty. \tag{A.14}$$

□

A.3 Chapter II: Asymptotic Equivalency between the renewable estimator and oracle MLE

Proof. Now we prove Theorem II.6. The difference of two equations (A.9) and (A.13) suggests that

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

Theorem II.4 or equation (A.14) implies that $\|\tilde{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_0\|^2 = O_p(1/N_j),\ j=1,\ldots,b$. By Condition (C2), it is easy to see that

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

□

Table A.3: Simulation results summarized from 500 replications, under the setting of $N_B = 100,000$ and $p = 5$ for the linear model. Batch size n_b varies from 50 to 2000.

		$B = 50, n_b = 2000$			$B = 100, n_b = 1000$		
	AI-SGD	MLE	online LSE	Renew	MLE	online LSE	Renew
A.bias$\times 10^{-3}$	13.48	3.17	3.17	3.17	3.17	3.17	3.17
ASE$\times 10^{-3}$	15.08	3.83	3.82	3.83	3.83	3.83	3.83
ESE$\times 10^{-3}$	17.24	3.94	3.94	3.94	3.94	3.94	3.94
CP	0.92	0.94	0.94	0.94	0.94	0.94	0.94
C.Time(s)	-	0.44	0.06	0.08	0.56	0.08	0.12
R.Time(s)	0.14	0.31	0.02	0.04	0.32	0.02	0.07
		$B = 200, n_b = 500$			$B = 500, n_b = 200$		
	AI-SGD	MLE	online LSE	Renew	MLE	online LSE	Renew
Abs. bias$\times 10^{-3}$	13.48	3.17	3.17	3.32	3.17	3.17	3.32
ASE$\times 10^{-3}$	15.08	3.83	3.83	4.08	3.83	3.83	4.08
ESE$\times 10^{-3}$	17.24	3.94	3.94	3.94	3.94	3.94	3.94
CP	0.92	0.94	0.94	0.95	0.94	0.94	0.95
C.Time(s)	-	0.86	0.11	0.17	1.68	0.19	0.34
R.Time(s)	0.14	0.32	0.04	0.11	0.30	0.07	0.24
		$B = 1000, n_b = 100$			$B = 2000, n_b = 50$		
	AI-SGD	MLE	online LSE	Renew	MLE	online LSE	Renew
A.bias$\times 10^{-3}$	13.48	3.17	3.17	3.17	3.17	3.17	3.17
ASE$\times 10^{-3}$	15.08	3.83	3.83	3.83	3.83	3.83	3.83
ESE$\times 10^{-3}$	17.24	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
C.Time(s)	-	3.012	0.348	0.648	5.906	0.660	1.273
R.Time(s)	0.14	0.29	0.14	0.47	0.29	0.28	0.95

Table A.4: Simulation results summarized from 500 replications, under the setting of $N_B = 100,000$ and $p = 5$ for the Binomial logistic model. Batch size n_b varies from 50 to 2000.

			$B = 50, n_b = 2000$			$B = 100, n_b = 1000$		
	AI-SGD	MLE	CEE	CUEE	Renew	CEE	CUEE	Renew
A.bias$\times 10^{-3}$	24.98	6.31	6.33	6.32	6.32	6.40	6.34	6.32
ASE$\times 10^{-3}$	27.10	7.82	7.83	7.82	7.82	7.84	7.83	7.82
ESE$\times 10^{-3}$	31.14	7.93	7.90	7.92	7.92	7.88	7.93	7.92
CP	0.92	0.95	0.95	0.95	0.95	0.94	0.95	0.95
			$B = 50, n_b = 2000$			$B = 100, n_b = 1000$		
	AI-SGD	MLE	CEE	CUEE	Renew	CEE	CUEE	Renew
A.bias$\times 10^{-3}$	24.98	6.31	6.66	6.42	6.32	8.31	6.89	6.32
ASE$\times 10^{-3}$	27.10	7.82	7.87	7.84	7.82	7.94	7.86	7.82
ESE$\times 10^{-3}$	31.14	7.93	7.82	7.99	7.92	7.67	8.43	7.93
CP	0.92	0.95	0.93	0.94	0.95	0.88	0.92	0.95
			$B = 50, n_b = 2000$			$B = 100, n_b = 1000$		
	AI-SGD	MLE	CEE	CUEE	Renew	CEE	CUEE	Renew
A.bias$\times 10^{-3}$	24.98	6.31	**13.01**	**8.26**	6.32	**24.50**	**11.98**	6.32
ASE$\times 10^{-3}$	27.10	7.82	8.07	7.89	7.82	8.34	7.94	7.82
ESE$\times 10^{-3}$	31.14	7.93	7.45	10.01	7.92	7.02	15.64	7.92
CP	0.92	0.95	**0.66**	**0.87**	0.95	**0.12**	**0.74**	0.95

Table A.5: Simulation results summarized from 500 replications, under the setting of $N_B = 100,000$ and $p = 5$ for the Poisson log-linear model. Batch size n_b varies from 50 to 2000.

			$B = 50, n_b = 2000$			$B = 100, n_b = 1000$		
	AI-SGD	MLE	CEE	CUEE	Renew	CEE	CUEE	Renew
A.bias$\times 10^{-3}$	13.30	2.76	2.80	2.77	2.76	2.87	2.78	2.76
ASE$\times 10^{-3}$	15.15	3.42	3.42	3.42	3.42	3.42	3.42	3.42
ESE$\times 10^{-3}$	15.99	3.42	3.42	3.42	3.42	3.42	3.43	3.42
CP	0.91	0.95	0.95	0.95	0.95	0.95	0.95	0.95
			$B = 50, n_b = 2000$			$B = 100, n_b = 1000$		
	AI-SGD	MLE	CEE	CUEE	Renew	CEE	CUEE	Renew
Abs. bias$\times 10^{-3}$	13.30	2.76	3.13	2.86	2.76	**4.27**	3.20	2.76
ASE$\times 10^{-3}$	15.15	3.42	3.42	3.42	3.42	3.42	3.42	3.42
ESE$\times 10^{-3}$	15.99	3.42	3.42	3.50	3.42	3.42	3.82	3.42
CP	0.91	0.95	0.91	0.94	0.95	**0.78**	0.90	0.95
			$B = 50, n_b = 2000$			$B = 100, n_b = 1000$		
	AI-SGD	MLE	CEE	CUEE	Renew	CEE	CUEE	Renew
A.bias$\times 10^{-3}$	13.30	2.76	**6.26**	**4.01**	2.76	**10.48**	**5.98**	2.76
ASE$\times 10^{-3}$	15.15	3.42	3.41	3.41	3.42	3.41	3.39	3.42
ESE$\times 10^{-3}$	15.99	3.42	3.42	4.59	3.42	3.43	6.82	3.42
CP	0.91	0.95	**0.76**	**0.81**	0.95	**0.67**	**0.74**	0.95

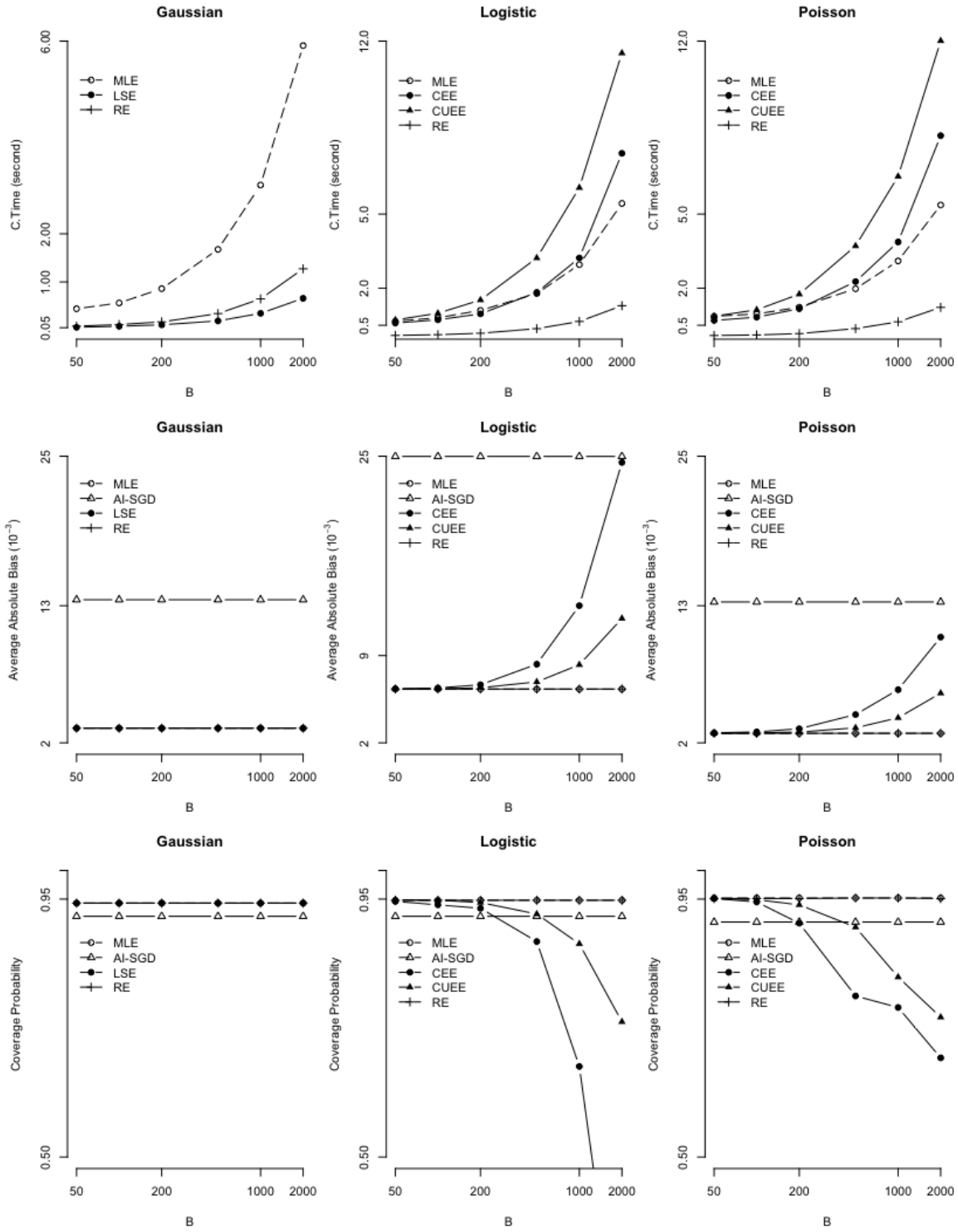

Figure A.1: Average computation time, average bias and coverage probabilities for MLE, AI-SGD, online LSE, sequential CEE and CUEE, and Renewable estimation. AI-SGD is not included in C.Time comparison.

Table A.6: Empirical size and power of a simple hypothesis test over 500 replications in the logistic regression model with $p = 5, n_b = 200, B = 500$.

	Size	Power									
β_{03}	0.2	0.205	0.210	0.215	0.220	0.225	0.230	0.235	0.240	0.245	0.250
MLE	0.050	0.098	0.358	0.664	0.880	0.980	0.998	0.998	1.000	1.000	1.000
AI-SGD	0.050	0.051	0.058	0.102	0.136	0.196	0.288	0.352	0.452	0.534	0.608
CEE	0.110	0.044	0.082	0.324	0.628	0.876	0.972	0.996	0.998	1.000	1.000
CUEE	0.062	0.072	0.268	0.570	0.830	0.952	0.990	0.998	1.000	1.000	1.000
Renew	0.048	0.094	0.354	0.656	0.878	0.980	0.998	0.998	1.000	1.000	1.000

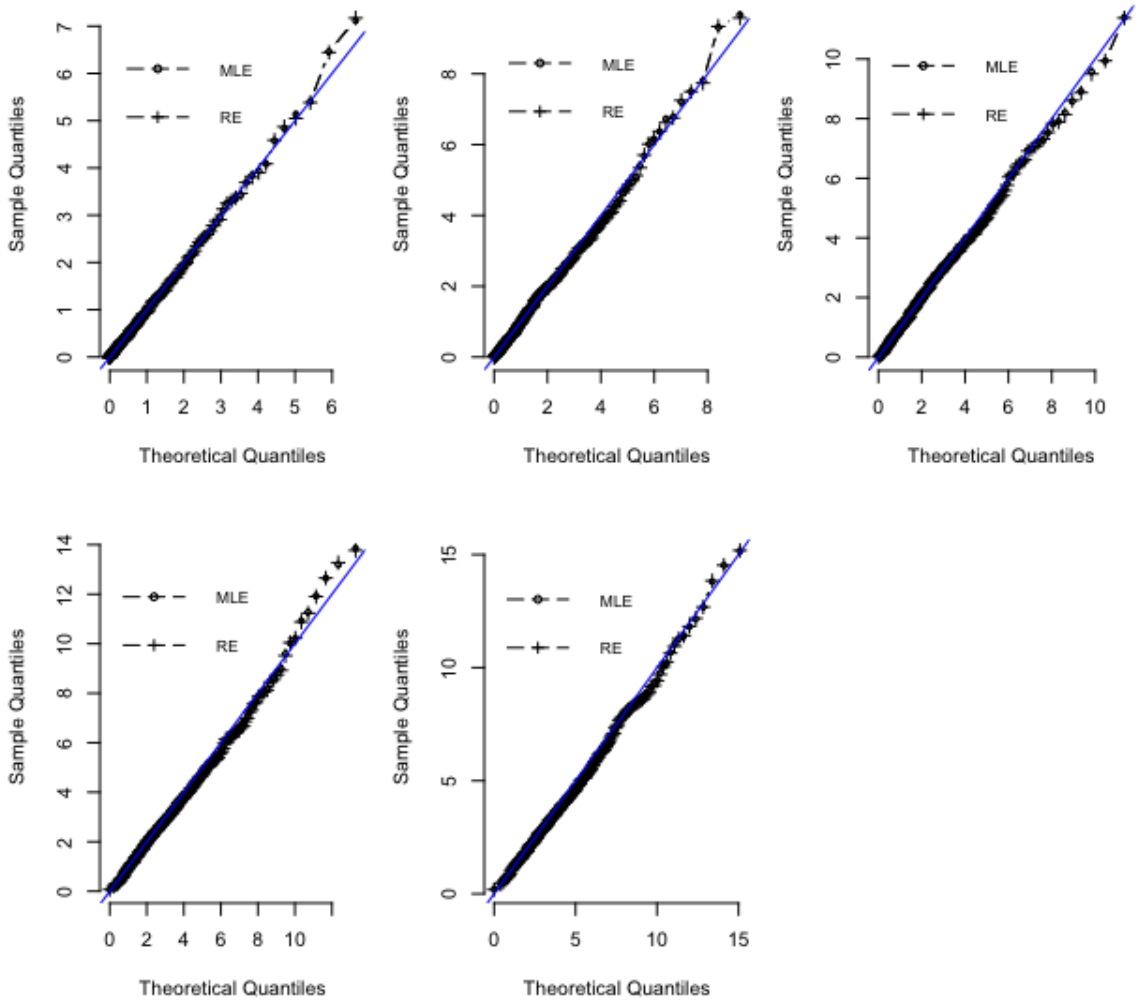

Figure A.2: Quantiles of the Wald test statistics under H_0 with degrees of freedom equal to $1, 2, 3, 4, 5$.

APPENDIX B

Appendices for Chapter III

B.1 Chapter III: Derivation of Renewable GEE

Proof. The initial estimate $\hat{\boldsymbol{\beta}}_1$ satisfies estimating equation, $\boldsymbol{\psi}_1(D_1; \hat{\boldsymbol{\beta}}_1, \hat{\alpha}_1) = \mathbf{0}$. When D_2 arrives, we hope to obtain the renewable estimator, $\hat{\boldsymbol{\beta}}_2^\star$, that satisfies the following estimating equation:

$$\boldsymbol{\psi}_1(D_1; \hat{\boldsymbol{\beta}}_2^\star, \hat{\alpha}_2^\star) + \boldsymbol{\psi}_2(D_2; \hat{\boldsymbol{\beta}}_2^\star, \hat{\alpha}_2^\star) = \mathbf{0}, \tag{B.1}$$

where $\boldsymbol{\psi}_b(D_b; \boldsymbol{\beta}, \alpha) = \sum_{i \in D_b} \boldsymbol{D}_i^T \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}_i)$ is the estimating function for the current data batch D_b, and the corresponding sensitivity and variability matrices are denoted by $\boldsymbol{S}_b(D_b; \boldsymbol{\beta}, \alpha) = \sum_{i \in D_b} \boldsymbol{D}_i^T \boldsymbol{\Sigma}_i^{-1} \boldsymbol{D}_i$ and $\boldsymbol{V}_b(D_b; \boldsymbol{\beta}, \alpha) = \sum_{i \in D_b} \boldsymbol{D}_i^T \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}_i)(\boldsymbol{y}_i - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} \boldsymbol{D}_i$, respectively, $b = 1, 2, \ldots$. However, solving (B.1) for $\hat{\boldsymbol{\beta}}_2^\star$ involves the use of subject-level data in both data batches D_1 and D_2. To derive a renewable version of estimation, we take first-order Taylor expansion of the first term $\boldsymbol{\psi}_1(D_1; \hat{\boldsymbol{\beta}}_2^\star)$ in (B.1) around $\hat{\boldsymbol{\beta}}_1$,

$$\boldsymbol{\psi}_1(D_1; \hat{\boldsymbol{\beta}}_1) + \boldsymbol{S}_1(D_1; \hat{\boldsymbol{\beta}}_1)(\hat{\boldsymbol{\beta}}_1 - \hat{\boldsymbol{\beta}}_2^\star) + \boldsymbol{\psi}_2(D_2; \hat{\boldsymbol{\beta}}_2^\star) + O_p(n_1 \|\hat{\boldsymbol{\beta}}_2^\star - \hat{\boldsymbol{\beta}}_1\|^2) = \mathbf{0}. \tag{B.2}$$

The error term $O_p(n_1\|\hat{\boldsymbol{\beta}}_2^{\star} - \hat{\boldsymbol{\beta}}_1\|^2)$ in (B.2) may be asymptotically ignored because under some mild conditions, both $\hat{\boldsymbol{\beta}}_1$ and $\hat{\boldsymbol{\beta}}_2^{\star}$ are consistent estimator of $\boldsymbol{\beta}_0$. Removing such term, we propose a new estimator $\tilde{\boldsymbol{\beta}}_2$ as a solution to the equation of the form:

$$\boldsymbol{\psi}_1(D_1;\hat{\boldsymbol{\beta}}_1) + \boldsymbol{S}_1(D_1;\hat{\boldsymbol{\beta}}_1)(\hat{\boldsymbol{\beta}}_1 - \tilde{\boldsymbol{\beta}}_2) + \boldsymbol{\psi}_2(D_2;\tilde{\boldsymbol{\beta}}_2) = \mathbf{0},$$

where $\boldsymbol{\psi}_1(D_1;\hat{\boldsymbol{\beta}}_1) = \mathbf{0}$. Thus, the proposed estimator $\tilde{\boldsymbol{\beta}}_2$ satisfies the following estimating equation:

$$\boldsymbol{S}_1(D_1;\hat{\boldsymbol{\beta}}_1)(\hat{\boldsymbol{\beta}}_1 - \tilde{\boldsymbol{\beta}}_2) + \boldsymbol{\psi}_2(D_2;\tilde{\boldsymbol{\beta}}_2) = \mathbf{0}. \tag{B.3}$$

Note that $\tilde{\boldsymbol{\beta}}_2$ approximates the oracle estimator $\hat{\boldsymbol{\beta}}_2^{\star}$ up to the second order asymptotic errors. Through (B.3), the initial $\hat{\boldsymbol{\beta}}_1$ is renewed by $\tilde{\boldsymbol{\beta}}_2$. Because of this, $\tilde{\boldsymbol{\beta}}_2$ is *a renewable estimator* of $\boldsymbol{\beta}_0$, and equation (B.3) is termed as *an incremental estimating equation*. Numerically, it is rather straightforward to find $\tilde{\boldsymbol{\beta}}_2$ by, for example, the Newton-Raphson algorithm. That is, at the $(r+1)$-th iteration,

$$\begin{aligned}\tilde{\boldsymbol{\beta}}_2^{(r+1)} &= \tilde{\boldsymbol{\beta}}_2^{(r)} + \Big\{\boldsymbol{S}_1(D_1;\hat{\boldsymbol{\beta}}_1) + \boldsymbol{S}_2(D_2;\tilde{\boldsymbol{\beta}}_2^{(r)})\Big\}^{-1}\Big\{\boldsymbol{S}_1(D_1;\hat{\boldsymbol{\beta}}_1)(\hat{\boldsymbol{\beta}}_1 - \tilde{\boldsymbol{\beta}}_2^{(r)}) + \boldsymbol{\psi}_2(D_2;\tilde{\boldsymbol{\beta}}_2^{(r)})\Big\} \\ &= \tilde{\boldsymbol{\beta}}_2^{(r)} + \Big\{\tilde{\boldsymbol{S}}_2(\hat{\boldsymbol{\beta}}_1,\tilde{\boldsymbol{\beta}}_2^{r})\Big\}^{-1}\tilde{\boldsymbol{\psi}}_2(D_2;\tilde{\boldsymbol{\beta}}_2^{(r)}),\end{aligned} \tag{B.4}$$

where no subject-level data of D_1, but only the prior estimate $\hat{\boldsymbol{\beta}}_1$ and the prior sensitivity matrix $\boldsymbol{S}_1(D_1;\hat{\boldsymbol{\beta}}_1)$ are used in the above iterative algorithm. In equation (B.4), $\tilde{\boldsymbol{\beta}}_2$ is iteratively solved by using the adjusted estimating function $\tilde{\boldsymbol{\psi}}_2$ and the aggregated information matrix $\Big\{\boldsymbol{S}_1(\hat{\boldsymbol{\beta}}_1) + \boldsymbol{S}_2(\tilde{\boldsymbol{\beta}}_2)\Big\}$.

The oracle estimator of nuisance parameters α and ϕ are:

$$\hat{\boldsymbol{\alpha}}_2^\star = s\left\{\hat{R}_2^\star(\hat{\boldsymbol{\beta}}_2^\star)\right\} = \frac{1}{\sum_{b=1}^{2}\sum_{i=1}^{n_b} m_{b,i} - p} s\left(\sum_{b=1}^{2}\sum_{i=1}^{n_b} \hat{\boldsymbol{r}}_{b,i}^\star \hat{\boldsymbol{r}}_{b,i}^{T\star}\right)$$
$$\hat{\phi}_2^\star = \frac{1}{\sum_{b=1}^{2}\sum_{i=1}^{n_b} m_{b,i} - p} \sum_{b=1}^{2}\sum_{i=1}^{n_b} \hat{\boldsymbol{r}}_{b,i}^{T\star} \hat{\boldsymbol{r}}_{b,i}^\star,$$

where $\hat{\boldsymbol{r}}_{b,i}^{T\star} = (\hat{r}_{b,i1}, \ldots, \hat{r}_{b,im_i})^T = \left\{\frac{y_{b,i1} - h(\boldsymbol{x}_{b,i1}^T \hat{\boldsymbol{\beta}}_2^\star)}{v(\boldsymbol{x}_{b,i1}^T \hat{\boldsymbol{\beta}}_2^\star)}, \ldots, \frac{y_{b,im_i} - h(\boldsymbol{x}_{b,im_i}^T \hat{\boldsymbol{\beta}}_2^\star)}{v(\boldsymbol{x}_{b,im_i}^T \hat{\boldsymbol{\beta}}_2^\star)}\right\}^T$ is the Pearson residual for cluster i in data batch D_b, $s(\cdot)$ is the function that links the residual matrix to the estimator of correlation parameter and it depends on the working correlation structure. $\hat{\boldsymbol{R}}$ is the unstructured working correlation determined by residuals.

The renewable estimators $\tilde{\boldsymbol{\alpha}}_2$ and $\tilde{\phi}_2$ are calculated using the following equations:

$$\begin{aligned}
\tilde{\boldsymbol{\alpha}}_2 &= \frac{1}{\sum_{b=1}^{2}\sum_{i=1}^{n_b} m_{b,i} - p}\left\{s\left(\sum_{i=1}^{n_1} \hat{\boldsymbol{r}}_{1,i}\hat{\boldsymbol{r}}_{1,i}^T\right) + s\left(\sum_{i=1}^{n_2} \tilde{\boldsymbol{r}}_{2,i}\tilde{\boldsymbol{r}}_{2,i}^T\right)\right\} \\
&= \frac{\sum_{i=1}^{n_1} m_{1,i} - p}{\sum_{b=1}^{2}\sum_{i=1}^{n_b} m_{b,i} - p} s\left\{\hat{\boldsymbol{R}}_1(\hat{\boldsymbol{\beta}}_1)\right\} + \frac{\sum_{i=1}^{n_2} m_{2,i} - p}{\sum_{b=1}^{2}\sum_{i=1}^{n_b} m_{b,i} - p} s\left\{\hat{\boldsymbol{R}}_2(\tilde{\boldsymbol{\beta}}_2)\right\} \\
&= \frac{\sum_{i=1}^{n_1} m_{1,i} - p}{\sum_{b=1}^{2}\sum_{i=1}^{n_b} m_{b,i} - p}\hat{\boldsymbol{\alpha}}_1 + \frac{\sum_{i=1}^{n_2} m_{2,i} - p}{\sum_{b=1}^{2}\sum_{i=1}^{n_b} m_{b,i} - p}\hat{\boldsymbol{\alpha}}_2 \qquad \text{(B.5)} \\
\tilde{\phi}_2 &= \frac{1}{\sum_{b=1}^{2}\sum_{i=1}^{n_b} m_{b,i} - p}\left(\sum_{i=1}^{n_1} \hat{\boldsymbol{r}}_{1,i}^T\hat{\boldsymbol{r}}_{1,i} + \sum_{i=1}^{n_2} \tilde{\boldsymbol{r}}_{2,i}^T\tilde{\boldsymbol{r}}_{2,i}\right) \\
&= \frac{\sum_{i=1}^{n_1} m_{1,i} - p}{\sum_{b=1}^{2}\sum_{i=1}^{n_b} m_{b,i} - p}\hat{\phi}_1 + \frac{\sum_{i=1}^{n_2} m_{2,i} - p}{\sum_{b=1}^{2}\sum_{i=1}^{n_b} m_{b,i} - p}\hat{\phi}_2,
\end{aligned}$$

where $\hat{\boldsymbol{r}}_1 = \hat{\boldsymbol{r}}_1(\hat{\boldsymbol{\beta}}_1)$ and $\tilde{\boldsymbol{r}}_2 = \tilde{\boldsymbol{r}}_2(\tilde{\boldsymbol{\beta}}_2)$. □

B.2 Chapter III: Consistency and Normality of Renewable QIF

Proof. The quadratic inference function estimator of the cumulative dataset to time point b is $\hat{\boldsymbol{\beta}}_b^\star = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, Q_b^\star(\boldsymbol{\beta})$.

Let $\boldsymbol{\beta}_0$ be the true parameter and $\tilde{\boldsymbol{\beta}}_b$ be the renewable estimator. Note that for

the prior data batch D_1, we have $\tilde{\boldsymbol{\beta}}_1 = \hat{\boldsymbol{\beta}}_1^{\star} = \hat{\boldsymbol{\beta}}_1$, which is consistent by the existing theory of the generalized method of moments (GMM) estimators. Now we prove the consistency of $\tilde{\boldsymbol{\beta}}_b$ for an arbitrary $b \geq 2$ by the method of induction.

Define a function

$$f_b(\boldsymbol{\beta}) = \frac{1}{N_b}\left\{\sum_{j=1}^{b-1}\boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{G}_b(\boldsymbol{\beta})\right\}^T \left\{\sum_{j=1}^{b-1}\boldsymbol{C}_j(\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{C}_b(\boldsymbol{\beta})\right\}^{-1} \times \left\{\sum_{j=1}^{b-1}\boldsymbol{g}_j(\tilde{\boldsymbol{\beta}}_j) + \sum_{j=1}^{b-1}\boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j)(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}) + \boldsymbol{g}_b(\boldsymbol{\beta})\right\}$$

The renewable estimator $\tilde{\boldsymbol{\beta}}_b$ satisfies:

$$f_b(\tilde{\boldsymbol{\beta}}_b) = \mathbf{0}. \tag{B.6}$$

When $\tilde{\boldsymbol{\beta}}_j$ is consistent for $j = 1, \ldots, b-1$, we have

$$\begin{aligned} f_b(\boldsymbol{\beta}_0) = \frac{1}{N_b}&\left\{\sum_{j=1}^{b-1}\boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{G}_b(\boldsymbol{\beta}_0)\right\}^T \left\{\sum_{j=1}^{b-1}\boldsymbol{C}_j(\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{C}_b(\boldsymbol{\beta}_0)\right\}^{-1} \\ &\times \left\{\sum_{j=1}^{b-1}\boldsymbol{g}_j(\tilde{\boldsymbol{\beta}}_j) + \sum_{j=1}^{b-1}\boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j)(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) + \boldsymbol{g}_b(\boldsymbol{\beta}_0)\right\} = O_p(N_b^{-1/2}) \end{aligned} \tag{B.7}$$

Taking a difference between equations (B.7)-(B.6), we get

$$
\begin{aligned}
f_b(\boldsymbol{\beta}_0) - f_b(\tilde{\boldsymbol{\beta}}_b) = \frac{1}{N_b} & \left\{ \sum_{j=1}^{b-1} \boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{G}_b(\boldsymbol{\beta}_0) \right\}^T \left\{ \sum_{j=1}^{b-1} \boldsymbol{C}_j(\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{C}_b(\boldsymbol{\beta}_0) \right\}^{-1} \\
& \times \left\{ \sum_{j=1}^{b-1} \boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j)(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) + \boldsymbol{g}_b(\boldsymbol{\beta}_0) - \boldsymbol{g}_b(\tilde{\boldsymbol{\beta}}_b) \right\} \\
& + \frac{1}{N_b} \left[\left\{ \sum_{j=1}^{b} \boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j) \right\}^T \left\{ \sum_{j=1}^{b} \boldsymbol{C}_j(\tilde{\boldsymbol{\beta}}_j) \right\}^{-1} \right. \\
& \left. - \left\{ \sum_{j=1}^{b-1} \boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{G}_b(\boldsymbol{\beta}_0) \right\}^T \left\{ \sum_{j=1}^{b-1} \boldsymbol{C}_j(\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{C}_b(\boldsymbol{\beta}_0) \right\}^{-1} \right] \\
& \times \left\{ \sum_{j=1}^{b-1} \boldsymbol{g}_j(\tilde{\boldsymbol{\beta}}_j) + \boldsymbol{g}_b(\tilde{\boldsymbol{\beta}}_b) \right\} = O_p(N_b^{-1/2}).
\end{aligned} \tag{B.8}
$$

Since

$$
\begin{aligned}
\boldsymbol{g}_b(\tilde{\boldsymbol{\beta}}_b) &= \boldsymbol{g}_b(\boldsymbol{\beta}_0) - \boldsymbol{G}_b(\boldsymbol{\xi}_b)(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) \\
&= \boldsymbol{g}_b(\boldsymbol{\beta}_0) - \{\boldsymbol{G}_b(\boldsymbol{\beta}_0) - \boldsymbol{G}_b(\boldsymbol{\beta}_0) + \boldsymbol{G}_b(\boldsymbol{\xi}_b)\} (\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) \\
&= \boldsymbol{g}_b(\boldsymbol{\beta}_0) - \boldsymbol{G}_b(\boldsymbol{\beta}_0)(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) + \{\boldsymbol{G}_b(\boldsymbol{\beta}_0) - \boldsymbol{G}_b(\boldsymbol{\xi}_b)\} (\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0).
\end{aligned}
$$

Since $\boldsymbol{G}_b$ is Lipschitz continuous in Θ, there exists $M(D_b) > 0$ such that $\|\boldsymbol{G}_b(\boldsymbol{\beta}_0) - \boldsymbol{G}_b(\boldsymbol{\xi}_b)\| \leq M(D_b)\|\boldsymbol{\xi}_b - \boldsymbol{\beta}_0\| \leq M(D_b)\|\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0\|$, and it follows that

$$
\boldsymbol{g}_b(\tilde{\boldsymbol{\beta}}_b) - \boldsymbol{g}_b(\boldsymbol{\beta}_0) = -\boldsymbol{G}_b(\boldsymbol{\beta}_0)(\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0) + O_p(n_b\|\tilde{\boldsymbol{\beta}}_b - \boldsymbol{\beta}_0\|^2). \tag{B.9}
$$

Since $\boldsymbol{g}_b$ is continuous differentiable, $\boldsymbol{C}_b$ is also continuous in Θ. Plug in (B.9) into (B.8), we get

$$\begin{aligned}
&\frac{1}{N_b}\left\{\sum_{j=1}^{b-1}\boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j)+\boldsymbol{G}_b(\boldsymbol{\beta}_0)\right\}^T\left\{\sum_{j=1}^{b-1}\boldsymbol{C}_j(\tilde{\boldsymbol{\beta}}_j)+\boldsymbol{C}_b(\boldsymbol{\beta}_0)\right\}^{-1}\\
&\times\left\{\sum_{j=1}^{b-1}\boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j)+\boldsymbol{G}_b(\boldsymbol{\beta}_0)\right\}(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)\\
&+O_p\left(\frac{n_b}{\sqrt{N_b}}\|\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0\|^2\right)+O_p(N_b^{-1/2})=\mathbf{0}.
\end{aligned}\tag{B.10}$$

Under the assumption that $\tilde{\boldsymbol{\beta}}_j$ is consistent and $\tilde{\boldsymbol{\beta}}_j \in \mathbb{N}_\delta$ for $j = 1, \ldots, b-1$, and by condition (C3), $\boldsymbol{C}_j(\tilde{\boldsymbol{\beta}}_j)$ is positive-definite, it follows that

$$\tilde{\boldsymbol{\beta}}_b \xrightarrow{p} \boldsymbol{\beta}_0,\ N_b \to \infty.$$

Furthermore, by Theorem III.1, all $\tilde{\boldsymbol{\beta}}_j$ are consistent for $j = 1, \ldots, b-1$, and by condition (C2), the Continuous Mapping Theorem implies that

$$\begin{aligned}
&\frac{1}{N_b}\left\{\sum_{j=1}^{b-1}\boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j)+\boldsymbol{G}_b(\boldsymbol{\beta}_0)\right\}^T\left\{\sum_{j=1}^{b-1}\boldsymbol{C}_j(\tilde{\boldsymbol{\beta}}_j)+\boldsymbol{C}_b(\boldsymbol{\beta}_0)\right\}^{-1}\\
&\times\left\{\sum_{j=1}^{b-1}\boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j)+\boldsymbol{G}_b(\boldsymbol{\beta}_0)\right\}\xrightarrow{p}\boldsymbol{J}(\boldsymbol{\beta}_0),\ N_b\to\infty,
\end{aligned}$$

where $\boldsymbol{J}(\boldsymbol{\beta}_0) = \boldsymbol{G}^T(\boldsymbol{\beta}_0)\boldsymbol{C}^{-1}(\boldsymbol{\beta}_0)\boldsymbol{G}(\boldsymbol{\beta}_0)$. By Condition (C3), $\boldsymbol{C}_b$ is positive-definite, and the Central Limit Theorem implies

$$\sqrt{N_b}(\tilde{\boldsymbol{\beta}}_b-\boldsymbol{\beta}_0)\xrightarrow{d}\mathcal{N}(\mathbf{0},\boldsymbol{J}^{-1}(\boldsymbol{\beta}_0))\tag{B.11}$$

□

B.3 Chapter III: Asymptotic Equivalency Between the Renewable QIF and the Oracle Estimators

Proof. The oracle QIF estimator $\hat{\boldsymbol{\beta}}_b^\star$ for the cumulative dataset $D_b^\star$ satisfies:

$$\frac{1}{N_b}\left\{\sum_{j=1}^{b}\boldsymbol{G}_j(\hat{\boldsymbol{\beta}}_b^\star)\right\}^T\left\{\sum_{j=1}^{b}\boldsymbol{C}_j(\hat{\boldsymbol{\beta}}_b^\star)\right\}^{-1}\left\{\sum_{j=1}^{b}\boldsymbol{g}_j(\hat{\boldsymbol{\beta}}_b^\star)\right\}=\mathbf{0}.$$

Taking the first-order Taylor expansion around $\boldsymbol{\beta}_0$ leads to

$$\begin{aligned}&\frac{1}{N_b}\left\{\sum_{j=1}^{b}\boldsymbol{G}_j(\hat{\boldsymbol{\beta}}_b^\star)\right\}^T\left\{\sum_{j=1}^{b}\boldsymbol{C}_j(\hat{\boldsymbol{\beta}}_b^\star)\right\}^{-1}\\&\times\left\{\sum_{j=1}^{b}\boldsymbol{g}_j(\boldsymbol{\beta}_0)+\sum_{j=1}^{b}\boldsymbol{G}_j(\boldsymbol{\beta}_0)(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_b^\star)+O_p\left(\sum_{j=1}^{b}n_j\|\hat{\boldsymbol{\beta}}_b^\star-\boldsymbol{\beta}_0\|^2\right)\right\}=\mathbf{0}.\end{aligned}\tag{B.12}$$

Since $N_b^{-1}\sum_{j=1}^{b}\boldsymbol{G}_j(\hat{\boldsymbol{\beta}}_b^\star)=\boldsymbol{G}(\boldsymbol{\beta}_0)+o_p(1)$ and $N_b^{-1}\sum_{j=1}^{b}\boldsymbol{C}_j(\hat{\boldsymbol{\beta}}_b^\star)=\boldsymbol{C}(\boldsymbol{\beta}_0)+o_p(1)$, equation (B.12) becomes

$$\begin{aligned}&\frac{1}{N_b}\left\{\sum_{j=1}^{b}\boldsymbol{G}_j(\boldsymbol{\beta}_0)\right\}^T\left\{\sum_{j=1}^{b}\boldsymbol{C}_j(\boldsymbol{\beta}_0)\right\}^{-1}\\&\times\left\{\sum_{j=1}^{b}\boldsymbol{g}_j(\boldsymbol{\beta}_0)+\sum_{j=1}^{b}\boldsymbol{G}_j(\boldsymbol{\beta}_0)(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_b^\star)+O_p\left(\sum_{j=1}^{b}n_j\|\hat{\boldsymbol{\beta}}_b^\star-\boldsymbol{\beta}_0\|^2\right)\right\}=\mathbf{0}.\end{aligned}\tag{B.13}$$

Since we have

$$f_b(\tilde{\boldsymbol{\beta}}_b)=\frac{1}{N_b}\left\{\sum_{j=1}^{b}\boldsymbol{G}_j(\tilde{\boldsymbol{\beta}}_j)\right\}^T\left\{\sum_{j=1}^{b}\boldsymbol{C}_j(\tilde{\boldsymbol{\beta}}_j)\right\}^{-1}\left\{\sum_{j=1}^{b}\boldsymbol{g}_j(\tilde{\boldsymbol{\beta}}_j)\right\}=\mathbf{0}.\tag{B.14}$$

Taking the first-order Taylor expansion of $\sum_{j=1}^{b}\boldsymbol{g}_j(\tilde{\boldsymbol{\beta}}_j)$ around $\boldsymbol{\beta}_0$, we obtain

$$\sum_{j=1}^{b}\boldsymbol{g}_j(\tilde{\boldsymbol{\beta}}_j)=\sum_{j=1}^{b}\boldsymbol{g}_j(\boldsymbol{\beta}_0)+\sum_{j=1}^{b}\boldsymbol{G}_j(\boldsymbol{\beta}_0)(\boldsymbol{\beta}_0-\tilde{\boldsymbol{\beta}}_b)+O_p\left(\sum_{j=1}^{b}n_j\|\tilde{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_0\|^2\right)\tag{B.15}$$

Plugging (B.15) into equation (B.14), and by the Continuous Mapping Theorem, we obtain

$$\begin{aligned}&\frac{1}{N_b}\left\{\sum_{j=1}^{b}\boldsymbol{G}_j(\boldsymbol{\beta}_0)\right\}^T\left\{\sum_{j=1}^{b}\boldsymbol{C}_j(\boldsymbol{\beta}_0)\right\}^{-1}\\&\times\left\{\sum_{j=1}^{b}\boldsymbol{g}_j(\boldsymbol{\beta}_0)+\sum_{j=1}^{b}\boldsymbol{G}_j(\boldsymbol{\beta}_0)(\boldsymbol{\beta}_0-\tilde{\boldsymbol{\beta}}_b)+O_p\left(\sum_{j=1}^{b}n_j\|\tilde{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_0\|^2\right)\right\}=\mathbf{0}.\end{aligned}\tag{B.16}$$

Take the difference of equations (B.13) and (B.16) implies that

$$\begin{aligned}&\frac{1}{N_b}\left\{\sum_{j=1}^{b}\boldsymbol{G}_j(\boldsymbol{\beta}_0)\right\}^T\left\{\sum_{j=1}^{b}\boldsymbol{C}_j(\boldsymbol{\beta}_0)\right\}^{-1}\left\{\sum_{j=1}^{b}\boldsymbol{G}_j(\boldsymbol{\beta}_0)\right\}(\tilde{\boldsymbol{\beta}}_b-\hat{\boldsymbol{\beta}}_b^\star)\\=&O_p\left(\sum_{j=1}^{b}\frac{n_j}{N_b}\|\tilde{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_0\|^2+\|\hat{\boldsymbol{\beta}}_b^\star-\boldsymbol{\beta}_0\|^2\right)\\=&O_p(1/N_b).\end{aligned}\tag{B.17}$$

It is easy to see that

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

□

APPENDIX C

Appendices for Chapter IV

C.1 Chapter IV: Asymptotic Normality

Proof. We take the first-order Taylor expansion of the aggregated estimating equation $\tilde{\boldsymbol{U}}_b(\tilde{\boldsymbol{\alpha}}_b)$ around $\boldsymbol{\alpha}_0$, $\tilde{\boldsymbol{U}}_b(\tilde{\boldsymbol{\alpha}}_b) = \tilde{\boldsymbol{U}}_b(\boldsymbol{\alpha}_0) + \frac{\partial \tilde{\boldsymbol{U}}_b(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}^T}(\tilde{\boldsymbol{\alpha}}_b - \boldsymbol{\alpha}_0) = \mathbf{0}$. Therefore, we have

$$\sqrt{N_b}(\tilde{\boldsymbol{\alpha}}_b - \boldsymbol{\alpha}_0) = \left\{ -\frac{1}{N_b}\frac{\partial \tilde{\boldsymbol{U}}_b(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}^T} \right\}^{-1} \left\{ \frac{1}{\sqrt{N_b}} \tilde{\boldsymbol{U}}_b(\boldsymbol{\alpha}_0) \right\}, \tag{C.1}$$

where $\tilde{\boldsymbol{S}}_b = \mathbb{E}\left\{ -\frac{\partial \tilde{\boldsymbol{U}}_b(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}^T} \right\} = \sum_{j=1}^{b} \boldsymbol{X}_j^T \{ \boldsymbol{X}_j + \boldsymbol{Z}_j \boldsymbol{L}_j(\boldsymbol{\alpha}) \}$.

The second term on the right-hand side of equation (C.1) may be written as follows,

$$\frac{1}{\sqrt{N_b}} \tilde{\boldsymbol{U}}_b(\boldsymbol{\alpha}) = \frac{1}{\sqrt{N_b}} \sum_{j=1}^{b} \boldsymbol{X}_j^T (\boldsymbol{y}_j - \boldsymbol{X}_j \boldsymbol{\alpha} - \boldsymbol{Z}_j \boldsymbol{m}_j).$$

Denote $\boldsymbol{U}_j = \boldsymbol{X}_j^T (\boldsymbol{y}_j - \boldsymbol{X}_j \boldsymbol{\alpha} - \boldsymbol{Z}_j \boldsymbol{m}_j) = \sum_{i \in D_j} \boldsymbol{u}_{ji} = \sum_{i \in D_j} \boldsymbol{x}_{ji}(y_{ji} - \boldsymbol{x}_{ji}^T \boldsymbol{\alpha} - \boldsymbol{z}_{ji}^T \boldsymbol{m}_j)$, then $\tilde{\boldsymbol{U}}_b = \sum_{j=1}^{b} \boldsymbol{U}_j$. Let $\mathscr{F}_j$ represent the σ-field generated by $D_j^\star$. Since $\mathbb{E}[\boldsymbol{U}_j \mid \mathscr{F}_{j-1}] = \mathbf{0}$, then $\{\boldsymbol{U}_j, \mathscr{F}_j, j = 1, 2, \dots\}$ forms a sequence of martingale difference with mean $\mathbf{0}$.

To derive the joint distribution of $\tilde{\boldsymbol{U}}_b$, we apply the Cramér-Wold theorem (*Cramér and Wold*, 1936). For any nonrandom nonzero vector $\boldsymbol{a} = (a_1, \ldots, a_p)^T \in \mathbb{R}^p$. Let $\boldsymbol{u}_{ji} = (u_{ji,1}, \ldots, u_{ji,p})^T$, we write

$$\boldsymbol{a}^T \tilde{\boldsymbol{U}}_b = \sum_{i=1}^{N_b} \sum_{d=1}^{p} a_d u_{i,d} = \sum_{i=1}^{N_b} u_i^\star. \tag{C.2}$$

Since $\{\boldsymbol{\beta}_b\}$ is a stationary AR(1) process, it is a ϕ-mixing process *Billingsley* (1968). Given $\boldsymbol{\beta}_j$, $\{\boldsymbol{u}_{ji}\}$ in (C.2) is conditionally independent with $\mathbb{E}[\boldsymbol{u}_{ji}] = \boldsymbol{0}$, and thus the $\{\boldsymbol{u}_{ji}\}$ is a ϕ-mixing centered process (*Billingsley*, 1968). It follows that $u_1^\star, u_2^\star, \ldots$ is also a stationary ϕ-mixing centered stochastic process whose second moments are given by

$$\sigma_{N_b}^2 = \operatorname{var}\left(\sum_{i=1}^{N_b} u_i^\star\right) \to \infty, \text{ as } N_b \to \infty.$$

Now we check the Lindeberg condition, for any $\epsilon > 0$,

$$\begin{aligned}
\sum_{i=1}^{N_b} \mathbb{E}\left\{(u_i^\star)^2 \mathbf{1}[|u_i^\star| > \epsilon \sigma_{N_b}]\right\} &= \sum_{i=1}^{N_b} \mathbb{E}\left\{\left(\sum_{d=1}^{p} a_d u_{i,d}\right)^2 \mathbf{1}\left[\sum_{d=1}^{p} |a_d u_{i,d}| > \epsilon \sigma_{N_b}\right]\right\} \\
&\leq \sum_{i=1}^{N_b} \sum_{d=1}^{p} a_d^2 \mathbb{E}\left\{u_{i,d}^2 \mathbf{1}\left[\sum_{d=1}^{p} |u_{i,d}| > \frac{\epsilon \sigma_{N_b}}{\max\limits_{d} |a_d|}\right]\right\}
\end{aligned}$$

Since $\sigma_{N_b} \to \infty$ and $\max |a_d| < \infty$, it follows that $\mathbf{1}\left[\sum_{d=1}^{p} |u_{i,d}| > \frac{\epsilon \sigma_{N_b}}{\max\limits_{i} |a_i|}\right] \xrightarrow{a.s.} 0$. Additionally, $\mathbb{E}[u_{i,d}^2] < \infty$, and $P(u_{i,d} = \infty) = 0$, it follows that

$$\sum_{i=1}^{N_b} \mathbb{E}\left\{(u_i^\star)^2 \mathbf{1}[|u_i^\star| > \epsilon \sigma_{N_b}]\right\} \to 0, \text{ as } N_b \to \infty.$$

By the central limit theorem for ϕ-mixing stochastic process $\{u_i^\star\}$ (*Peligrad*, 1986), we have

$$\frac{\sum_{i=1}^{N_b} u_i^\star}{\sigma_{N_b}^2} \xrightarrow{d} \mathcal{N}(0, 1),$$

where $\sigma^2_{N_b} = \boldsymbol{a}^T \text{var}[\tilde{\boldsymbol{U}}_b]\boldsymbol{a}$. Then by Cramér-Wold, we have

$$\frac{1}{\sqrt{N_b}}\tilde{\boldsymbol{U}}_b \xrightarrow{d} \mathcal{N}_p(\boldsymbol{0}, \boldsymbol{V}),$$

where $\boldsymbol{V} = \lim_{b\to\infty} \frac{1}{N_b}\tilde{\boldsymbol{X}}_b^T \text{var}[\tilde{\boldsymbol{U}}_b]\tilde{\boldsymbol{X}}_b = \lim_{b\to\infty} \tilde{\boldsymbol{V}}_b$ and $\tilde{\boldsymbol{X}}_b = (\boldsymbol{X}_1^T, \ldots, \boldsymbol{X}_b^T)^T$ is a combined covariate matrix with dimension $N_b \times p$.

Summarizing the above arguments, we finally prove that

$$\sqrt{N_b}(\boldsymbol{\alpha} - \boldsymbol{\alpha}_0) \xrightarrow{d} \mathcal{N}_p(\boldsymbol{0}, \boldsymbol{\Sigma}(\boldsymbol{\alpha}_0)), \text{ as } N_b \to \infty,$$

with $\boldsymbol{\Sigma}(\boldsymbol{\alpha}_0) = \lim_{N_b} N_b \tilde{\boldsymbol{J}}_b^{-1}(\boldsymbol{\alpha}_0)$, where $\tilde{\boldsymbol{J}}_b(\boldsymbol{\alpha}_0) = \tilde{\boldsymbol{S}}_b^T \tilde{\boldsymbol{V}}_b^{-1} \tilde{\boldsymbol{S}}_b$. □

BIBLIOGRAPHY

BIBLIOGRAPHY

Amari, S.-I., H. Park, and K. Fukumizu (2000), Adaptive method of realizing natural gradient learning for multilayer perceptrons, *Neural Computation*, *12*(6), 1399–1409.

Amin, R., and A. J. Search (1991), A nonparametric exponentially weighted moving average control scheme, *Communications in Statistics – Simulation and Computation*, *20*, 1049–1072.

Amin, R. W., M. R. Reynolds, and S. T. Bakir (1995), Nonparametric quality control charts based on the sign statistic, *Communications in Statistics – Theory and Methods*, *24*, 1597–1623.

Angelosante, D., and G. Giannakis (2012), Group lassoing change-points in piecewise-constant ar processes, *EURASIP J. Adv. Signal Process.*, *2012*, 70.

Basseville, M., and I. Nikiforov (1993), *Detection of Abrupt Changes: Theory and Application*, vol. 104, Prentice Hall Englewood Cliffs: Upper Saddle River, NJ, USA.

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

Billingsley, P. (1968), *Convergence of Probability Measures*, Wiley, New York.

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

Broderick, T., N. Boyd, A. Wibisono, A. C. Wilson, and M. I. Jordan (2013), Streaming variational bayes, *in Advances in Neural Information Processing Systems*, pp. 1727 – 1735.

Bucak, S. S., and B. Gunsel (2009), Incremental subspace learning via non-negative matrix factorization, *Pattern Recognition*, *42*(5), 788–797.

Cappé, O. (2011), Online em algorithm for hidden markov models, *Journal of Computational and Graphical Statistics*, *20*(3), 728 – 749.

Cappé, O., and E. Moulines (2009), Online expectation-maximization algorithm for latent data models, *Journal of Royal Statistical Society: Series B*, *71*(3), 593–613.

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

Chen, W., L. Chen, Z. Chen, and S. Tu (2005), A realtime dynamic traffic control system based on wireless sensor network, in *2005 International Conference on Parallel Processing Workshops (ICPPW'05)*, pp. 258–264.

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

Chen, X., W. Liu, and Y. Zhang (2018), Quantile regression under memory constraint, arXiv:1810.08264v1.

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

Ciuciu, P., P. Abry, C. Rabrait, and H. Wendt (2008), Log wavelet leaders cumulant based multifractal analysis of evi fmri time series: Evidence of scaling in ongoing and evoked brain activity, *IEEE Journal of Selected Topics in Signal Processing*, *2*(6), 929–943.

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

Cramér, H., and H. Wold (1936), Some theorems on distribution functions, *Journal of the London Mathematical Society*, *11*(4), 290 – 294.

Czado, C., and P. X.-K. Song (2008), State space mixed models for longitudinal observations with binary and binomial responses, *Statistical Papers*, *49*, 691–714.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977), Maximum likelihood from incomplete data via the em algorithm, *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*(1), 1 – 38.

Dias, D., and J. a. P. S. Cunha (2018), Wearable health devices – vital sign monitoring, systems and technologies, *Sensors (Basel)*, *18*(8), 2414, doi:10.3390/s18082414.

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

Enea, M., R. Meiri, and T. Kalimi (2015), *speedglm: Fitting linear and generalized linear models to large data sets*.

Fahrmeir, L., and H. Kaufmann (1985), Consistency and asymptotic normality of the maximum likelihood estimator in generalized linear models, *The Annals of Statistics*, *13*(1), 342–368.

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

Frigola, R., Y. Chen, and C. E. Rasmussen (2014), Variational gaussian process state-space models, *in Advances in Neural Information Processing Systems*, pp. 3680 – 3688.

Gaber, M. M., A. Zaslavsky, and S. Krishnaswamy (2005), Mining data streams: a review, *ACM SIGMOD Record*, *34*(2), 18–26.

Goel, A. L., and S. M. Wu (1971), Determination of a.r.l. and a contour nomogram for cusum charts to control normal mean, *Technometrics*, *13*(2), 221–230.

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

Hansen, L. P. (1982), Large sample properties of generalized method of moments estimators, *Econometrica*, *50*(4), 1029–1054.

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

Harchaoui, Z., and C. L. Leduc (2010), Multiple change-point estimation with a total variation penalty, *J. Amer. Statist. Assoc.*, *105*, 1480–1493.

Harvey, A. C. (1981), *Time Series Models*, Allan, Oxford.

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

Horowitz, J. L. (1998), Bootstrap methods for median regression models, *Econometrica*, *66*(6), 1327–1351.

Johari, R., L. Pekelis, and D. J. Walsh (2016), Always valid inference: Bringing sequential analysis to a/b testing, ArXiv:1512.04922v2.

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

Jørgensen, B., and P. X.-K. Song (2007), Stationary state space models for longitudinal data, *The Canadian Journal of Statistics*, *35*(4), 461 – 483.

Jørgensen, B., S. Lundbye-Christensen, P. X.-K. Song, and L. Sun (1999), A state-space models for multivariate longitudinal count data, *Biometrika*, *86*, 169 – 181.

Kitagawa, G. (1987), Non-gaussian state-space modeling of non stationary time series (with discussion), *Journal of the American Statistical Association*, *82*, 1032 – 1063.

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

Koenker, R. (2005), *Quantile regression*, Cambridge University Press.

Lai, T. L. (2004), Likelihood ratio identities and their applications to sequential analysis, *Sequential Analysis*, *23*(4), 467–497.

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

Liang, K.-Y., and S. Zeger (1986), Longitudinal data analysis using generalized linear models, *Biometrika*, *73*, 13–22.

Lin, N., and R. Xi (2011), Aggregated estimating equation estimation, *Statistics and Its Interface*, *4*(1), 73–83.

Lindsay, B. G., and A. Qu (2003), Inference functions and quadratic score tests, *Statistical Science*, *18*(3), 394 – 410.

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

Liu, S., A. Wright, and M. Hauskrecht (2017), Change-point detection method for clinical decision support system rule monitoring, *Artif Intell Med*, *10259*, 126–135.

Lumley, T. (2013), *biglm: Bounded memory linear and generalized linear models*.

Luo, L., and P. X.-K. Song (2020), Renewable estimation and incremental inference in generalized linear models with streaming datasets, *Journal of the Royal Statistical Society: Series B*, *82*, 69–97.

L'Heureux, A., K. Grolinger, H. F. Elyamany, and M. A. M. Capretz (2017), Machine learning with big data: Challenges and approaches, *IEEE Access*, *5*, 7776–7797.

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

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

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

Miller, A. J., B. A. Kiberd, I. P. Alwayn, A. Odutayo, and K. K. Tennankore (2017), Donor-recipient weight and sex mismatch and the risk of graft loss in renal transplantation, *Clinical Journal of the American Society of Nephrology*, *12*(4), 669–676.

Nion, D., and N. D. Sidiropoulos (2009), Adaptive algorithms to track the parafac decomposition of third-order tensor, *IEEE Transactions on Signal Processing*, *57*(6), 2299–2310.

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

Page, E. S. (1954), Continuous inspection schemes, *Biometrika*, *41*(1), 100–115.

Peek, N., J. H. Holmes, and J. Sun (2014), Technical challenges for big data in biomedicine and health: Data sources, infrastructure, and analytics, *IMIA Yearbook of Medical Informatics*, pp. 42–47.

Peligrad, M. (1986), *Recent advances in the central limit theorem and its weak invariance principle for mixing sequences of random variables (a survey)*, Birkhäuser, Boston, MA.

Pollak, M., and D. Siegmund (1991), Sequential detection of a change in a normal mean when the initial value is unknown, *The Annals of Statistics*, *19*(1), 394–416.

Poor, H. V., and O. Hadjiliadis (2008), *Quickest Detection*, Cambridge, U.K.: Cambridge Univ. Press.

Priyanka, K., and N. Kulennavar (2014), A survey on big data analytics in health care, *International Journal of Computer Science and Information Technologies*, *5*(4), 5865–5868.

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

Qian, J., and L. Su (2013), Shrinkage estimation of regression models with multilple structural changes.

Qu, A., and P. X.-K. Song (2004), Assessing robustness of generalised estimating equations an quadratic inference functions, *Biometrika*, *91*(2), 447–459.

Qu, A., B. Lindsay, and B. Li (2000), Improving generalised estimating equations using quadratic inference functions, *Biometrika*, *87*, 823–76.

Robbins, H., and S. Monro (1951), A stochastic approximation method, *The Annals of Mathematical Statistics*, *22*(3), 400–407.

Roberts, S. W. (1966), A comparison of some control chart procedures, *Technometrics*, *8*, 411–430.

Robinson, G. K. (1991), That blup is a good thing: The estimation of random effects, *Statist. Sci.*, *6*(1), 15–32, doi:10.1214/ss/1177011926.

Rojas, C. R., and B. Wahlberg (2014), On change point detection using the fused lasso method, arXiv:1401.5408v1.

Sadik, S., L. Gruenwald, and E. Leal (2018), Wadjet: Finding outliers in multiple multi-dimensional heterogeneous data streams, in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 1232–1235.

Sahoo, P. K., S. K. Mohapatra, and S.-L. Wu (2016), Analyzing healthcare big data with prediction for future health condition, *IEEE Access*, *4*, 9786 – 9799.

Sakrison, D. J. (1965), Efficient recursive estimation: application to estimating the parameter of a covariance function, *International journal of engineering science*, *3*(4), 461–483.

Schifano, E. D., J. Wu, C. Wang, J. Yan, and M.-H. Chen (2016), Online updating of statistical inference in the big data setting, *Technometrics*, *58*(3), 393–403.

Schraudolph, N. N., J. Yu, and S. Günter (2007), A Stochastic Quasi-Newton Method for Online Convex Optimization, in *Proc. 11th Intl. Conf. Artificial Intelligence and Statistics (AIstats)*, *Workshop and Conference Proceedings*, vol. 2, edited by M. Meila and X. Shen, pp. 436–443, Journal of Machine Learning Research, San Juan, Puerto Rico.

Shiryayev, A. N. (1963), On optimal method in earliest detection problems, *Theory Probab. Appl.*, *8*, 26–51.

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

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

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

Sundaram, H., A. Maureen, W. S.Cherikh, C. B.Tolleris, B. A.Bresnahan, and C. P.Johnson (2002), Post-transplant renal function in the first year predicts long-term kidney transplant survival, *Kidney International*, *62*(1), 311–318.

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

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

Tillmann, F.-P., I. Quack, M. Woznowski, and L. C. Rump (2019), Effect of recipient-donor sex and weight mismatch on graft survival after deceased donor renal transplantation, *PLoS One*, *14*(3), doi:e0214048.

Titterington, D. M. (1984), Recursive parameter estimation using incomplete data, *Journal of the Royal Statistical Society. Series B (Methodology)*, *46*(2), 257 – 267.

Toulis, P., and E. M. Airold (2017), Asymptotic and finite-sample properties of estimators based on stochastic gradients, *The Annals of Statistics*, *45*(4), 1694–1727.

Toulis, P., and E. M. Airoldi (2015), Scalable estimation strategies based on stochastic approximations: classical results and new insights, *Statistics and computing*, *25*(4), 781–795.

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

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

Wald, A. (1945), Sequential tests of statistical hypotheses, *The Annals of Mathematical Statistics*, *16*(2), 117 – 186.

West, M., and P. J. Harrison (1997), *Bayesian Forecasting and Dynamic Models*, 2nd ed., Springer-Verlag, New York.

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

Zhang, B., J. Geng, and L. Lai (2015), Multiple change-points estimation in linear regression models via sparse group lasso, *IEEE Transactions on Signal Processing*, *63*(9).

Zhang, Y., B. J. Jansen, and A. Spink (2009), Time series analysis of a web search engine transaction log, *Information Processing and Management*, *45*(2), 230–245.

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

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