Sampling for Streaming Data

by

Rui Xie

(Under the Direction of Ping Ma and T. N. Sriram)

Abstract

Advances in data acquisition technology pose challenges in analyzing large volumes of streaming data. Sampling is a natural yet powerful tool for analyzing such data sets due to their competent estimation accuracy and low computational cost. Unfortunately, sampling methods and their statistical properties for streaming data, especially streaming time series data, are not well studied in the literature. Meanwhile, estimating the dependence structure of multidimensional streaming time series data in real-time is challenging. With large volumes of streaming data, the problem becomes more difficult when the multidimensional data are collected asynchronously across distributed nodes, which motivates us to sample representative data points from streams. Here we propose a series of leverage score based sampling methods for streaming time series data. The simulation studies and real data analysis are conducted to validate the proposed methods. The theoretical analysis of the asymptotic behaviors of the least squares estimator is developed based on the subsamples.

We extended the proposed sampling methods to the application of learning velocity model of Full-Waveform Inversion (FWI), which is a high-resolution seismic imaging technique for geophysical site characterization.

INDEX WORDS: Leverage score, Streaming data, Sampling, Stopping rule, Online randomized algorithm, Time Series

Sampling for Streaming Data

by

Rui Xie

B.S., Xiamen University, China 2011

M.S., Georgia Institute of Technology, 2013

M.S., The University of Georgia, 2013

A Dissertation Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2019

Sampling for Streaming Data

by

Rui Xie

Approved:

Major Professors:	Ping Ma
	T.N. Sriram
Committee:	Wenxuan Zhong
	Abhyuday Mandal
	WenZhan Song
	Shuyang Bai
	Yao Xie

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
May 2019

Dedicated to my beloved parents & family

Acknowledgments

I would like to express my deepest appreciation to my major professors Ping Ma and T. N. Sriram, who have jointly guided me to explore the world of statistics. Ping Ma has the attitude and the substance of a truly great advisor. He has always been supportive and encouraging not only for my research but also for my future career. T. N. Sriram continually guided me to become a thoughtful scientist, and convincingly conveyed a spirit of excellent scholarship. This dissertation would not be possible without their guidance. I also want to express my sincere gratitude to my committee members: Wenxuan Zhong, Shuyang (Ray) Bai, Abhyuday Mandal, WenZhan Song and Yao Xie for their advice, help, and encouragement the dissertation research. It is my honor to have them all on my committee.

I want to appreciate my collaborators, co-authors of my conference and journal papers including Fangyu Li, Zengyan Wang, Yanfei Lu, Wei Biao Wu and Yuk Fai Leung for their contributions. This work would not be possible without their excellent collaboration. Especially, I want to express my thanks to all labmates of Big Data Analytics Lab (BDAL) for their supports: Xiaoxiao Sun, Yiwen Liu, Xin Xing, Xinlian Zhang, Wei Xu, Ye Wang, Honghe Jin, Cheng Meng, Jingyi Zhang, Di Xiao,

Mengrui Zhang, Jinyang Chen, Lexiang Ji, Huimin Cheng and many others whose names are not listed. I benefited tremendously from intellectual discussions with colleagues from Department of Statistics and other departments at the University of Georgia, including Chao Song, Wenqian Kong, and Xianyan Chen.

This dissertation research also received support from many staff members at Department of Statistics, including Nikki Rowden, Melissa Pettigrew, Megan T. Weatherford, Nathan T Cooper, and Mollie Johnson Hicks. Part of the dissertation research is supported by the United States National Science Foundation (DMS-1222718, DMS-1438957, DMS-1228288, DMS-1440037, and DMS-1228288) and the United States National Institutes of Health (R01 GM113242 and R01GM122080).

Finally, I want to thank my parents and family for their love, endless support, encouragement and sacrifices throughout my PhD studies and in my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Motivation

1.1 Overview

With the recent advances of data acquisition technology, massive datasets are generated and collected by all fields of science and engineering. As a standard representation of the data, we denote the data as a set of n data units in the p-dimensional space. For massive data sets, either p or n or both are large, which may pose computational challenges to traditional methods. Subsampling of both or one of the rows and columns of the $n \times p$ data matrix has been widely used to reduce the size or dimension of the large data sets. Recently, in a linear regression model, an innovative and effective importance sampling scheme based on leverage scores of the matrix has been proposed, which concentrates more on select a set of observations (subsample) that contains more information. The estimator based on such a subsample has been demonstrated to give a reasonable approximation to the estimator based on the full

data, and this subsampling approach yields a better performance than the simple random sampling method. Even though a linear statistical model is applicable in many statistical problems, the key limitation of it is that the observations are assumed to be independent. In practice, complex data, such as time series and spatial data, naturally arise from many areas, where the observations are dependent.

In modern massive data sets, the data size is either very large or it expands continuously in a streaming fashion. In these instances, conventional methods face computational challenges. For dependent data, the subsample could lead to improved estimates, such as variance reduction. However, we need to overcome some major challenges in order to develop leverage-based sampling methods for these data. Sampling the time series or spatial data that ignores the dependence structure will lead to a systematic biased estimation on the model parameter and inference. Therefore, carelessly applying leveraging methods from linear regression by sampling individual data points may destroy the dependence structure of the data. Developing theoretically justifiable and computationally scalable methods for large-sale **streaming dependent data** is our primary motivation. This research focuses on developing a series of **sampling methods** for temporally or spatially dependent data in the streaming setting and the decentralized data acquisition environment.

Due to variations in real data, an essential statistical question is that whether the leveraging sampling method has competitive statistical properties such as yielding a better mean squared error (MSE) when compared to other estimators. To answer this question in the context of streaming dependent data analysis, the proposal develops statistical leverage sampling theory and methods for streaming dependent

data such as autoregressive models in streaming time series data, and spatial data. The motivation of the proposed research is to address the emerging computational and analytical issues in big data analytics. The proposed methods produce innovative methodologies as well as inspire new lines of quantitative investigations in many disciplines.

One of the major challenges of the big data analytics is that we are still lacking the statistical and computational methods considering the computational resource constraints. One emerging method for dealing with large-scale data sets and heavy computational burden problems is subsampling. In subsampling approach, one first chooses a small portion of the full data, and then uses this sample as a surrogate to carry out computations of interest for the full data. For example, one might randomly sample a small number of rows from an input matrix and use those rows to construct a low-rank approximation to the original matrix, or one might randomly sample a small number of data points in a regression problem and then compute estimates of regression parameters using the subsample. For many problems, it is possible to construct the "worst-case" input for which the *uniform* random sampling will perform very poorly. Motivated by this, there has been a great deal of work on developing algorithms for matrix-based machine learning and data analysis problems that construct the random sample in a *nonuniform* data-dependent fashion Mahoney (2011).

Of particular interest is when that data-adaptive sampling process selects rows or columns from the input matrix according to a probability distribution that depends on the empirical statistical leverage scores of that matrix. In the regression set

up, this recently-developed approach of *statistical leveraging* has been applied to least squares approximation Drineas et al. (2006a, 2010), least absolute deviations regression Clarkson et al. (2013); Meng and Mahoney (2013), and low-rank matrix approximation Mahoney and Drineas (2009a); Clarkson and Woodruff (2013).

In spite of these impressive algorithmic results, works addressing statistical aspects of leveraging or leverage-based sampling are still lacking for dependent streaming data. Ma et al. (2014, 2015) bridged that gap by providing the first statistical analysis of the leveraging methods. They did so in the context of parameter estimation in fitting linear regression models for large sample independent data. The main theoretical contribution is that they provided an analytic framework for evaluating the statistical properties of algorithmic leveraging. Based on these theoretical results, they proposed and analyzed two new leveraging algorithms designed to improve upon vanilla leveraging and uniform sampling algorithms in terms of bias and variance. In both cases, they obtained the algorithmic benefits of leverage-based sampling, while achieves improved statistical performance.

1.2 Leveraging Methods in Linear Regression

We briefly review leveraging methods in linear models of the form

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1.1}$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ is a response vector, $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$ is an $n \times p$ design matrix, and $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^T$ is an error vector with $\boldsymbol{\varepsilon} \sim N(0, \sigma^2 I)$. The ordinary

least squares (OLS) estimator $\hat{\boldsymbol{\beta}}_{ols}$ of $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}}_{ols} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2 = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}, \tag{1.2}$$

where $\|\cdot\|$ is the Euclidean norm. The OLS is a linear estimator, i.e, the ith predicted response $\hat{y}_i$ can be written as $\hat{y}_i = \sum_{j=1}^n h_{ij}y_j$. In vector-matrix form, $\hat{\boldsymbol{y}} = \mathbf{H}\boldsymbol{y}$, where the hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$. The ith diagonal element of the hat matrix $\mathbf{H}$, $h_{ii} = \boldsymbol{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{x}_i$, is called the **leverage score** of the ith observation. The $\hat{\boldsymbol{\beta}}_{ols}$ can be calculated using the singular value decomposition (SVD) algorithm in Golub and Van Loan (1996). By SVD for $\mathbf{X}$, $\mathbf{H}$ is alternatively expressed as $\mathbf{H} = \mathbf{U}\mathbf{U}^T$, where $\mathbf{U}$ is the $n \times d$ left singular vector matrix of $\mathbf{X}$ and $d = \text{rank}(\mathbf{X})$. Then, the leverage score of the ith observation is expressed as

$$h_{ii} = \|\boldsymbol{u}_i\|^2, \tag{1.3}$$

where $\boldsymbol{u}_i$ is the ith row of $\mathbf{U}$.

Algorithm 1: Statistical Leveraging Algorithm in Linear Regression

---

1: **Subsampling**. Draw a random subsample of size $r \ll n$, denoted as $(\mathbf{X}^*, \boldsymbol{y}^*)$, i.e., draw r rows from the original data $(\mathbf{X}, \boldsymbol{y})$ according to the probability $\{\pi_i\}_{i=1}^n$. Record the corresponding sampling probability matrix $\boldsymbol{\Phi}^* = \text{diag}\{\pi_k^*\}_{k=1}^r$.

2: **Least squares**. Calculate the least squares estimate, $\tilde{\boldsymbol{\beta}}$, on the subsample,

$$\tilde{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{\Phi}^{*-1/2}\boldsymbol{y}^* - \boldsymbol{\Phi}^{*-1/2}\mathbf{X}^*\boldsymbol{\beta}\|^2 = (\mathbf{X}^{*T}\boldsymbol{\Phi}^{*-1}\mathbf{X}^*)^{-1}\mathbf{X}^{*T}\boldsymbol{\Phi}^{*-1}\boldsymbol{y}^*. \tag{1.4}$$

One key component of Algorithm 1 is the sampling probability $\{\pi_i\}_{i=1}^n$ in Step 1.

The leveraging algorithm constructs sampling probabilities by giving preference to "influential data points". Note that

$$h_{ii} = \frac{\partial \hat{y}_i}{\partial y_i},$$

which measures the oscillation amount of prediction $\hat{y}_i$ by a small perturbation of y_i. Therefore, in linear regression, h_{ii} is used as "influential" index. Following this line of thinking, one may draw the subsample according to a sampling distribution that is proportional to the leverage scores i.e., $\pi_i = h_{ii} / \sum_{i=1}^{n} h_{ii} = h_{ii}/p$. This is the rationale of the leveraging method Drineas et al. (2006a, 2010). To reduce the computational cost, one typically computes the leverage scores h_{ii} by random approximation Drineas et al. (2012a); Clarkson et al. (2013), or otherwise a random projection Ailon and Chazelle (2010); Clarkson et al. (2013) is used to precondition by approximately uniformizing them Drineas et al. (2010); Avron et al. (2010); Meng et al. (2014).

A distinguishing feature of sampling is that it can improve the performance of some estimates based on full sample complex data. For dependent data, the subsample could lead to a better estimate in terms of variance Näther (1985); Dette et al. (2013).

In spite of the advantages, there are significant challenges in developing leverage-based sampling methods for dependent data. In time series and spatial data, statistical analysis ignoring dependence leads to systematic bias in estimation and inference. Directly applying leveraging methods of linear regression by sampling individual data points will destroy the dependence structure of the data. Thus, novel theory and methods are needed to surmount these challenges. The central question that remains

to be answered for the proposed methods is

- Under what conditions on data and the underlying model, is the resulting leverage-based estimator a competitive estimator for statistical inference on a big data?

Organization

The rest of the thesis is organized as follows. In Chapter 2, we propose an online leverage-based sequential sampling algorithm for streaming time series data, which is assumed to come from an autoregressive model of order $p \geq 1$ (AR(p)). In Chapter 3, we propose a *leverage score sampling* (LSS) method for efficient online inference of the streaming vector autoregressive (VAR) model. In Chapter 4, we combine randomized subsampling techniques with a second-order optimization algorithm to propose the Sub-Sampled Newton (SSN) method for learning velocity model of Full-Waveform Inversion (FWI), which is a high-resolution seismic imaging technique for geophysical site characterization.

Chapter 2

Online Sequential Leveraging Sampling Method for Univariate Streaming Data[1]

In this chapter, we propose an online leverage-based sequential sampling algorithm for streaming time series data, which is assumed to come from an autoregressive model of order $p \geq 1$ (AR(p)). The proposed *sequential leveraging sampling* (SLS) method samples only one consecutively recorded block from the data stream for inference. While the starting point of the SLS scheme is chosen using a random mechanism based on leverage scores of the data, the subsample size is decided by a sequential stopping rule. We show that an appropriately normalized least squares

[1]Xie, R, Sriram, T. N., Wu, W. B., and Ma, P. (2019) Online sequential leveraging sampling method for streaming data, Journal of the Royal Statistical Society: Series B (Statistical Methodology), under review.

estimator based on the SLS block of the AR parameter vector is uniformly asymptotically normally distributed for non-explosive AR(p) model. Simulation studies and real data examples are presented to evaluate the empirical performance of the proposed SLS method.

2.1 Introduction

Advances in technology and discoveries in science have led to the rapid expansion of demands on analyzing vast volumes of data. In the meantime, new challenges have been posed for the modern data analysis tasks. First and foremost, as the data size grows enormously, many fundamental statistical methods such as least squares (LS) estimation and maximum likelihood estimation (MLE) become computationally infeasible, especially in instances where the data are acquired continuously over time in an online fashion or when computing devices are unable to load the entire dataset into the working memory. In such cases, data may take the form of data streams rather than finite stored data set (Babu and Widom, 2001). Analyzing such streaming data poses computational and efficiency challenges, which calls for online algorithms that can sequentially update or process data in batches.

We introduce a new method called *Sequential Leveraging Sampling* (SLS), an online batch learning method, as a solution to the aforementioned computational and efficiency challenges. SLS is designed to extract important information from the streaming data with three features: (1) sequential batch sampling for possibly temporally correlated data, (2) capability of dealing with high-frequency streams,

and (3) online estimation for the streaming data. With limited time and computing resources, SLS compresses and down-sizes the data stream through a randomized sampling technique, which keeps enough information for high precision estimation and speeds up computation by only selecting influential data points or batches from the stream. More specifically, the SLS, as a online random sampling method, aims to provide a way to summarize data streams. The goal of SLS is to obtain a small-size representation of the data streams by sampling important data points or batches.

The first feature of the SLS enables us to retain temporal dependence information among data points in the sampling procedure. When the data stream follows an autoregressive time series model, our SLS extracts important information from the temporal correlated data streams through the sequential block sampling technique. We use the Internet traffic stream as a way to motivate the usefulness of the SLS. It is well known that Internet traffics are autocorrelated (Cao et al., 2001; Cortez et al., 2012; Leland et al., 1993). In Figure 2.1, we show how the SLS can sample a block of data from an Internet traffic stream.

Example 2.1.1. Internet Traffic Data

Monitoring the Internet traffic is vital for the network security and management. Internet traffic data measures the flow of information across the network, such as the volumes of data packets exchange at a certain Internet node. The volume of traffic flow across a specific node is empirically used as an indicator of the Internet traffic, such as an Internet service provider (ISP), or a local network (Liang et al., 2006).

We use the SLS to sample and analyze an Internet traffic data (details in Section 2.6), which measures the TCP/IP protocol packets (in bits per second) exchanged

in a given time interval. This data is from the United Kingdom Education and Research Networking Association (UKERNA) and reflects aggregated traffic of the UK academic network backbone (Cortez et al., 2012). The Internet traffic stream was observed from 19th November 2004, 9:30 AM, to 27th January 2005, 11:11 AM every five minutes.

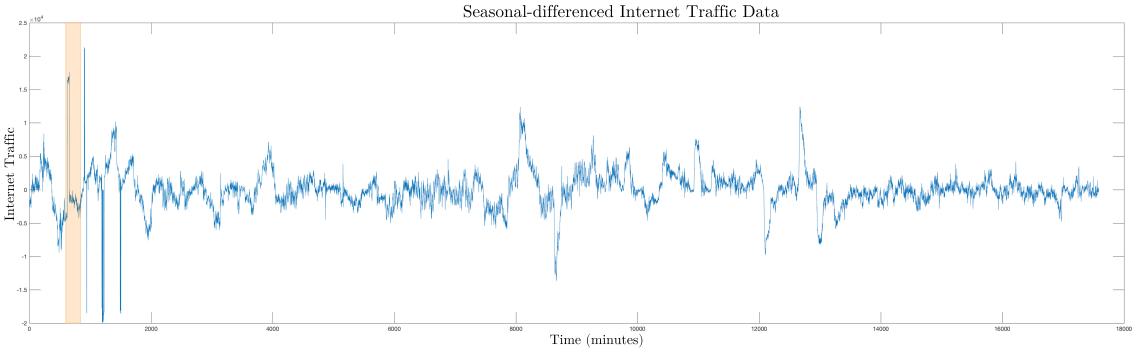

Figure 2.1: Scatter plot of the seasonal-differenced Internet traffic data. Data were observed in every five minutes from 19th November 2004, 9:30 AM, to 27th January 2005, 11:11 AM. As an example, the highlighted area indicates one of the 500 sequential leveraging subsamples that were sampled.

The second feature of the SLS distinguishes itself from the family of recursive estimation methods including recursive least squares and Kalman filter (Solo, 1981; Ljung and Söderström, 1983; Harvey, 1990; Guo, 1994; Young, 2012). These recursive methods obtain observations sequentially from the stream and update estimates with each new observation. When data arrive at a high frequency, such an update demands more computational resources to avoid a delayed update and catch up with the rate of data arrival. Specifically, as an example, the computational complexity of the recursive least square algorithm is $O(p^2)$ operations (flops) per iteration with p as the dimension of parameter vector (Haykin et al., 1996). Practically, the average running

time is 1.543×10^{-4} second per iteration from our simulation study, which is based on recursive least square estimation of AR(1) time series with 512 observations with 1000 independent replicates. This simulation study was conducted using MATLAB build-in function *recursiveAR* (The MathWorks, 2018) through a battery powered laptop computer with 2.4 GHz Intel Core i5 CPU. If the data arrives much faster than the speed of estimation update, e.g. data acquisition frequency (sample rate) in excess of 7 kHz in our simulation, the recursive update methods becomes impracticable to deal with such high-frequency stream. The real applications involving high frequency data includes medical image analysis (15.7 kHz, Yun et al. (2003)), acoustic signal monitoring (200 kHz, Wiggins and Hildebrand (2007)), seismic data acquisition (1000 MHz Davis and Annan (1989), 1500 MHz Hubbard et al. (2002)) etc. Moreover, in many real applications such as Internet of Things (IoT), computing devices have limited processing and energy resources that do not afford high frequency computing operations (Botta et al., 2016).

The growing popularity of high-frequency data and the increasing importance of the real time analysis make the computational efficiency an inevitable aspect in algorithm design. One possible remedy for the computational efficiency issue is to reduce the update frequency, which is equivalent to selecting data points or blocks used to update the estimation from data streams (Christensen et al., 2017). How to efficiently select the data points then becomes a sampling problem that we study in this paper. Sampling data points reduces the data frequency so that the data can be processed fast enough before new data points arrive. By using a computational efficient online sampling approach, the SLS reduces the size of data points used for

analysis and thus speeds up the processing time.

Last but not the least, the SLS allows us to analyze the streaming data in real time as the data arrives sequentially. The leverage-based online sampling approach used in SLS provides an immediate criterion for selecting the underlying data point. Along with the second feature of SLS, it makes the processing time per data point satisfy the real time constraint in streaming analysis. Moreover, the sequential block sampling technique in SLS is memory efficient and does not require us to access the past data in the stream. Based on these constraints, the SLS method produces a real time summary of the data stream.

By using a leverage-based non-uniform importance sampling, we construct a small "sketch" of the data stream, and use the sketch (SLS block) as a surrogate input to establish the computational task and statistical inference. This novel approach is introduced in detail in Section 2.3. In this article, we also establish theoretical properties of SLS such as parameter estimation and construction of a confidence region.

Related work

The study of streaming data originally came from the field of computer science and engineering. The ubiquitousness of streaming data is from the invention of smart instruments and sensors in cyber physical systems, which can automatically collect enormous volumes of data in real time. For example, in the weather station, medical facility, financial industry, transportation industry, and online retail, tons of data are collected in every second endlessly (Himberg et al., 2001; Zhu and Shasha, 2002;

Gaber et al., 2005; Moreira-Matias et al., 2013; Chen et al., 2002; Hu et al., 2013). There is intensive research in the field of computer science and engineering involving streaming data acquisition, storage, visualization, and information query in the communities of database, signal processing and pattern recognition of computer science and engineering (Fu, 2011; Papadimitriou et al., 2005; Woodruff, 2014; Garofalakis et al., 2016). There is also a proliferation of literature on the online analysis for streaming data, which usually requires real time analysis without the entire input data being available. Some representative examples are on methodology and software development (Bifet et al., 2010; Hoffman et al., 2010; Elhamifar and Kaluza, 2017b), on applications in different fields (Olivier et al., 2002; Mathioudakis and Koudas, 2010), and on online algorithms (Keogh et al., 2001; Kossmann et al., 2002).

Some of the recently developed family of randomized sampling methods aim to overcome the computational challenges in analyzing super-large-scale data including the streaming data. Sketching through random sampling is a popular tool in large-scale data analysis that has drawn a lot of attention in algorithmic development (Mahoney et al., 2011). There is a proliferation of literature in topics such as matrix approximation (Drineas et al., 2006b, 2012b; Woodruff et al., 2014), general least squares estimation (Ma et al., 2014; Raskutti and Mahoney, 2016), compressed sensing in streaming setting (Gilbert et al., 2007), and streaming anomaly detection (Huang and Kasiviswanathan, 2015), streaming network sampling (Ahmed et al., 2014; Gama et al., 2016). More specifically, in the context of linear regression, Drineas et al. (2006, 2012b) used the empirical statistical leverage scores as an importance sampling distribution and introduced the *algorithmic leveraging* method,

which samples and rescales full data to reduce the sample size before performing computations on the subproblem. Ma et al. (2014, 2015a) showed that the leverage-based sampling method is a viable alternative to the uniform sampling method in the context of big data. Raskutti and Mahoney (2016) extended the statistical analysis of randomized sketching to the general least squares problem.

It is important to note that randomized sketching methods cannot be directly applied to streaming data because these methods are almost exclusively developed for independent data. Whereas, most of the streaming data are inherently time-dependent. Moreover, the non-uniform random sampling methods, such as the leverage-based sampling method, rely on the calculation of sampling probabilities, e.g. leverage scores, based on the full sample. However, sampling probabilities cannot be calculated for streaming data as the observations are continuously evolving. Finally, statistical inference for streaming data and the related sampling algorithm are not readily available. To the best of our knowledge, the literature on statistical inference for and associated analysis of online streaming data is still at an early stage of development. For instance, Michalak et al. (2012) proposed a strategy for developing systems for real time streaming analysis; and Luts et al. (2014) developed real time semiparametric regression algorithms in a Bayesian framework.

In terms of taking samples from dependent data, the general sampling approach is to consider blocks of consecutive data rather than single data points (Politis et al., 1999; Zhang et al., 2013; Hall et al., 1995). Traditional resampling and subsampling methods are based on non-overlapping blocks or moving blocks as the sampling units (Lahiri, 2003). However, those methods are computationally intensive since

they usually need to go over the entire data set more than once.

To determine the sample (block) size for data stream sampling, a sequential approach is preferred in the context of online analysis since the sample size should be data dependent rather than prespecified (Grambsch, 1983; Lai and Siegmund, 1983; Barndorff-Nielsen and Cox, 1984). The SLS method keeps expanding the leveraging sample until the accumulated information reaches the prespecified level.

Organization

The rest of the article is organized as follows. In Section 2.2, we briefly review the autoregressive model for fixed sample size time series. In Section 2.3, we propose the sequential leveraging sampling method for the streaming AR(p) series. The theoretical results are established in Section 2.4. In Section 2.5, we present simulation results to support the theorems presented in the previous section. Two real data examples using sequential leveraging sampling method are discussed in Section 2.6. A brief discussion on the potential directions is included in Section 2.7. Proofs of our main results are included in the Appendix.

2.2 Overview of the Problem and Preliminaries

We consider a linear time series model for the streaming data $\{X_i\}_{i=-\infty}^{\infty}$, i.e. $p\ (\geq 1)$-th order autoregressive model (AR(p)),

$$X_i = \beta_1 X_{i-1} + \beta_2 X_{i-2} + \ldots + \beta_p X_{i-p} + \varepsilon_i, \tag{2.1}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ is the unknown parameter vector, the innovations $\{\varepsilon_i\}$ are assumed to be a sequence of independent and identically distributed (i.i.d.) random variables. We assume that our observed data starts at $X_1, \ldots, X_p$, and the innovations $\{\varepsilon_i\}_{i=1}^{\infty}$ are independent of these starting values with $\mathrm{E}(\varepsilon_i) = 0$ and $0 < \mathrm{Var}(\varepsilon_i) = \sigma^2 < \infty$. For data observed up to time n, let $\mathbf{z}_i = (X_{i-1}, \ldots, X_{i-p})^T$ and define the design matrix as

$$\boldsymbol{\Gamma}_n = \begin{bmatrix} \mathbf{z}_{p+1}^T \\ \mathbf{z}_{p+2}^T \\ \vdots \\ \mathbf{z}_n^T \end{bmatrix}. \tag{2.2}$$

Using this notation, we can write the AR(p) model observed up to time n as

$$\mathbf{x}_n = \boldsymbol{\Gamma}_n \boldsymbol{\beta} + \boldsymbol{\varepsilon}_n, \tag{2.3}$$

where $\mathbf{x}_n = (X_{p+1}, \ldots, X_n)^T$ and $\boldsymbol{\varepsilon}_n = (\varepsilon_{p+1}, \ldots, \varepsilon_n)^T$. The least square (LS) method (Anderson and Taylor, 1976) fits the AR(p) model by solving the optimization problem:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} ||\mathbf{x}_n - \boldsymbol{\Gamma}_n \boldsymbol{\beta}||^2, \tag{2.4}$$

where $||\cdot||$ is the ℓ_2 norm. For continuously observed streaming data, the "sample size" n is infinite. Thus, the actual observe data can be arbitrarily large, making the exact LS solution $\widehat{\boldsymbol{\beta}}_{n,LS} = (\boldsymbol{\Gamma}_n^T \boldsymbol{\Gamma}_n)^{\dagger} \boldsymbol{\Gamma}_n^T \mathbf{x}_n$ computationally challenging. Here $(\cdot)^{\dagger}$ is the Moore-Penrose pseudoinverse (Ben-Israel and Greville, 2003).

Random sampling is a popular approach to reduce the computational cost on large scale problems. In the SLS, we carefully design a "sketch" operator S, the leverage-based sequential block sampling, in the streaming setting and then implement the LS estimation on a much smaller sub-problem on the sketched stream $S\mathbf{x}_n$. That is, instead of solving problem (4.4) on the data stream $(\mathbf{x}_n, \Gamma_n)$, which is computational challenging or even impractical, we construct a sketched data $(S\mathbf{x}_n, S\Gamma_n)$ and then implement the LS estimation based on the sketched problem

$$\widehat{\boldsymbol{\beta}}_S = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} ||S\mathbf{x}_n - S\boldsymbol{\Gamma}_n\boldsymbol{\beta}||^2, \tag{2.5}$$

where $\widehat{\boldsymbol{\beta}}_S$ can also be an accurate estimator of the true parameter and we can obtain it in a computationally efficient way.

Next, we give some well known details concerning autoregressive (AR) model and statistical leverage scores (see, e.g., Box et al. (2011); Brockwell and Davis (2013)).

2.2.1 Fisher Information for an AR(p) Model

The observed Fisher information matrix, $\boldsymbol{J}_n$, of $\boldsymbol{\beta}$ for a sample $\{X_1, \ldots, X_n\}$ from the AR(p) model is

$$\boldsymbol{J}_n = \boldsymbol{\Gamma}_n^T\boldsymbol{\Gamma}_n, \tag{2.6}$$

and its trace is given by

$$\mathrm{tr}(\boldsymbol{J}_n) = \mathrm{tr}(\boldsymbol{\Gamma}_n^T\boldsymbol{\Gamma}_n) = \sum_{i=p+1}^{n} ||\mathbf{z}_i||^2. \tag{2.7}$$

For example, when $p = 1$, the observed Fisher information about β_1 contained in $\{X_1, \ldots, X_n\}$ is

$$\boldsymbol{J}_n = \boldsymbol{\Gamma}_n^T \boldsymbol{\Gamma}_n = -\frac{d^2}{d\beta^2} \left(\beta \sum_{i=2}^{n} X_{i-1} X_i - \frac{1}{2} \beta^2 \sum_{i=2}^{n} X_{i-1}^2 \right) = \sum_{i=2}^{n} X_{i-1}^2. \tag{2.8}$$

The observed Fisher information is crucial for the development of our Sequential Leveraging Sampling method.

2.2.2 Statistical Leverage Scores for an AR(p) Model

For an AR(p) model, the fitted values are expressed as $\widehat{\mathbf{x}}_n = \boldsymbol{\Gamma}_n \widehat{\boldsymbol{\beta}}_n = \mathbf{H}_n \mathbf{x}_n$, where $\mathbf{H}_n = \boldsymbol{\Gamma}_n (\boldsymbol{\Gamma}_n^T \boldsymbol{\Gamma}_n)^\dagger \boldsymbol{\Gamma}_n^T$ is the so-called *hat matrix* (Hau and Tong, 1989). For $\mathbf{z}_i$ defined above, the i^{th} diagonal element of $\mathbf{H}_n$,

$$h_{ii} = \mathbf{z}_i^T (\boldsymbol{\Gamma}_n^T \boldsymbol{\Gamma}_n)^\dagger \mathbf{z}_i, \tag{2.9}$$

is called the *statistical leverage* of the ith observation. Hau and Tong (1989) showed that h_{ii} may be interpreted as the amount of leverage or influence exerted on $\widehat{X}_i$ by X_i and nh_{ii} is interpreted as the Mahalanobis distance between $\mathbf{z}_i$ and the zero mean vector. Furthermore, they established various properties of the hat matrix including that $0 \leq h_{ii} \leq 1$. Motivated by the work of Drineas et al. (2006, 2012b) and Ma et al. (2014, 2015a) mentioned earlier, in Section 2.3 we will introduce a leverage-based sampling method for streaming time series.

2.2.3 Known Results for AR(p) Models

Suppose we denote the *characteristic polynomial* of an AR(p) model as

$$\phi(\boldsymbol{\beta}, \lambda) = \lambda^p - \beta_1 \lambda^{p-1} - \ldots - \beta_p, \tag{2.10}$$

and, for $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^T$, $\lambda_i = \lambda_i(\boldsymbol{\beta})$, $i = 1, \ldots, p$ denote the roots of the characteristic polynomial (2.10). If all the roots of the polynomial lie strictly inside the unit circle, then the AR(p) series is said to be *stable*, where the stability region Λ_p is defined as

$$\Lambda_p = \{\boldsymbol{\beta} \in \mathbb{R}^p : |\lambda_i(\boldsymbol{\beta})| < 1, i = 1, \ldots, p\}. \tag{2.11}$$

When the roots of the polynomial are inside the unit circle with at least one root on the unit circle, the AR(p) series is said to be *unstable*. The unstable AR(p) series thus contains at least one unit root (Dickey and Fuller, 1979, 1981). Finally, an AR(p) series is said to be purely *explosive* if all roots of the polynomial lie outside the unit circle.

In order to make inference about $\boldsymbol{\beta}$, it is natural to consider the randomly normalized quantity

$$\mathbf{V}_{n,\boldsymbol{\beta}} = (\boldsymbol{\Gamma}_n^T \boldsymbol{\Gamma}_n)^{1/2} (\widehat{\boldsymbol{\beta}}_{n,LS} - \boldsymbol{\beta}). \tag{2.12}$$

It is well known, however, that the limiting distribution of $\mathbf{V}_{n,\boldsymbol{\beta}}$ is drastically different for the three cases: stable, unstable, and explosive (Mann and Wald, 1943; Anderson, 1959; Anderson and Taylor, 1979; Chan and Wei, 1988; Jeganathan, 1988;

Lai and Wei, 1983). Let us denote the cumulative distribution function (c.d.f) of a p-dimensional random vector $\mathbf{V} = (V_1, \ldots, V_p)$ as function $\mathbf{F}_{\mathbf{V}}(\boldsymbol{v}) : \mathbb{R}^p \to \mathbb{R}$ such that

$$\mathbf{F}_{\mathbf{V}}(\boldsymbol{v}) = \mathrm{P}(V_1 \leq v_1, \ldots, V_p \leq v_p). \tag{2.13}$$

When the parameter vector $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^T \in \Lambda_p$ defined in (2.11), then it can be shown that the matrix

$$\frac{1}{n}\boldsymbol{J}_n = \frac{1}{n}\boldsymbol{\Gamma}_n^T\boldsymbol{\Gamma}_n \xrightarrow{a.s.} F, \ n \to \infty, \tag{2.14}$$

where $\xrightarrow{a.s.}$ indicates almost surely convergence and $F = F(\boldsymbol{\beta})$ is a positive definite matrix (Anderson, 2011). Furthermore, for $\mathbf{V}_{n,\boldsymbol{\beta}}$ defined in (2.12) and each $\boldsymbol{x} \in \mathbb{R}^p$, we have

$$\lim_{n\to\infty} \left|\mathbf{F}_{\mathbf{V}_{n,\boldsymbol{\beta}}}(\boldsymbol{x}) - \boldsymbol{\Phi}(\boldsymbol{x}/\sigma)\right| = 0, \tag{2.15}$$

which implies that

$$\frac{1}{\sigma^2}(\widehat{\boldsymbol{\beta}}_n - \boldsymbol{\beta})^T(\boldsymbol{\Gamma}_n^T\boldsymbol{\Gamma}_n)(\widehat{\boldsymbol{\beta}}_n - \boldsymbol{\beta}) \xrightarrow{\mathcal{L}} \chi_p^2, \text{ as } n \to \infty, \tag{2.16}$$

where χ_p^2 is the χ^2 distribution with p degrees of freedom and $\xrightarrow{\mathcal{L}}$ indicates convergence in law.

Under the streaming data setting, it is essential to design an efficient sampling method that can handle large volumes of streaming time series data and provide an accurate estimate of model parameters. That is, from the perspective of statistical inference, we aim to obtain an approximation to the sampling distribution of $\mathbf{V}_{n,\boldsymbol{\beta}}$

which would be valid regardless of whether the roots of $\phi(\boldsymbol{\beta}, z)$ lie inside or on the unit circle.

2.3 Sequential Leveraging Sampling Method for Streaming AR(p) Series

In SLS, we propose to adaptively take only one block of consecutive data points, named *Sequential Leveraging Sampling* block, as a sketch of the streaming data. The SLS block consists of two key components: the starting point and sequential block size. First, we use a leverage-based random sampling method to obtain a starting point X_l at time l of the SLS block. Then, the starting point X_l is expanded adaptively to form the SLS block $\mathbf{x}_{\boldsymbol{\tau_c}} = \{X_l, \ldots, X_{\tau_c}\}$, where the stopping time τ_c is decided according to sequential stopping rule.

When designing the SLS method, we take into account several characteristics of streaming time series. Streaming time series is an uninterrupted and infinite collection of observations collected at discrete time points, where observed data points are correlated and the sample size keeps on increasing. Consequently,

- Different from sampling individual data points in independent data case, we sample a block of consecutive data points as our subsample so that the correlation information can be kept in the subsample.
- On the choice of subsample, we prefer to choose the block containing high leverage score points as the SLS block. The leverage-based sampling tends to

be more efficient than the simple random (uniform) sampling on parameter estimation (Ma et al., 2015a).

- We use the sequential stopping rule to decide the size of the SLS block, which provides a fixed accuracy result for parameter estimation. In SLS, the subsample size is not fixed in advance. Instead, after determining a starting time, we evaluate each data point as it is collected and expand the SLS block until the sampling is stopped. We stop the sequential sampling as soon as the accumulated information contained in the active SLS block reaches a pre-specified significant level.

The detailed procedure of the SLS can be described as following. Suppose we observe a streaming time series $\{X_1, X_2, \ldots\}$, where $\{X_i\}$ follows a streaming AR(p) model in (2.1). First, we take a pilot subsample $\{X_1, \ldots, X_{n_0}\}$ from the streaming data, which can easily be stored in the working memory, and compute what we call the "observed information", $K = \sum_{i=p+1}^{n_0} ||\mathbf{z}_i||^2$ with $\mathbf{z}_i = (X_{i-1}, \ldots, X_{i-p})^T$.

Along with the data streaming through the working memory, i.e. time $j \geq n_0+1$, we conduct independent Bernoulli trials as described in Step 1 of Algorithm 1 (see below) in order to determine the starting value of the SLS block. More specifically, we calculate the success probability of the independent Bernoulli trial at time j, $\pi_j = \frac{||\mathbf{z}_j||^2}{\gamma K}$, which approximates the leverage score h_{jj} defined in (2.9) for a pre-specified leveraging parameter $\gamma \geq 1$.

We suppose that time l is the starting time determined by the independent Bernoulli trials. Then starting with X_l, we keep collecting consecutive data points to expand the SLS block until the sequential stopping rule is triggered at some time

Algorithm 2: Sequential Leveraging Sampling Algorithm

Input: Collect a pilot subsample $\{X_1, \ldots, X_{n_0}\}$, determine the autoregressive order p, and calculate $K = \sum_{i=p+1}^{n_0} ||\mathbf{z}_i||^2$. Information threshold $c > 0$ and leveraging parameter $\gamma > 1$.

Start the **online algorithm** for time $j \geq n_0 + 1$:

1: **Starting value via independent Bernoulli trials**: For the subsequent data point X_j, draw an independent Bernoulli variable B_j with success probability $\pi_j = \frac{||\mathbf{z}_j||^2}{\gamma K} \approx h_{jj}$;
if $B_j = 0$ **then** $j \leftarrow j + 1$, go back to step 1;
else $B_j = 1$ **return** starting time $l = j$ and start the sequential expansion;

2: **Sequential expansion**: Expand the input stream, starting with X_{l-p}, to form a block of consecutive observations $\{X_l, \ldots, X_{\tau_c}\}$, collected according to the sequential leveraging sampling rule $\tau_c = \inf\{t \geq l : \sum_{i=l}^{t} ||\mathbf{z}_i||^2 \geq c\}$.

3: **Least Squares Estimation on subproblem**: Calculate the LS estimator, $\widehat{\boldsymbol{\beta}}_{\tau_c} = (\boldsymbol{\Gamma}_{\tau_c}^T \boldsymbol{\Gamma}_{\tau_c})^{\dagger} \boldsymbol{\Gamma}_{\tau_c}^T \mathbf{x}_{\tau_\mathbf{c}}$, where $\boldsymbol{\Gamma}_{\tau_c}$ is defined in (2.18) and $\mathbf{x}_{\tau_\mathbf{c}} = \{X_l, \ldots, X_{\tau_c}\}$.

τ_c:

$$\tau_c = \inf\{t \geq l : \sum_{i=l}^{t} ||\mathbf{z}_i||^2 \geq c\}, \tag{2.17}$$

where $c(> 0)$ is a pre-specified constant called information threshold. Note that $\sum_{i=l}^{t} ||\mathbf{z}_i||^2$ is the trace of the observed Fisher information matrix for the block $\{X_l, \ldots, X_t\}$, if the ε's are normally distributed. Accordingly, we define the design matrix of the SLS block as

$$\boldsymbol{\Gamma}_{\tau_c} = \begin{bmatrix} \mathbf{z}_l^T \\ \vdots \\ \mathbf{z}_s^T \\ \vdots \\ \mathbf{z}_{\tau_c}^T \end{bmatrix}, \tag{2.18}$$

for the stopping time τ_c.

Finally, based on the SLS block $\mathbf{x}_{\tau_\mathbf{c}} = \{X_l, \ldots, X_{\tau_c}\}$ and its design matrix $\mathbf{\Gamma}_{\tau_c}$, we get the least squares estimator of $\boldsymbol{\beta}$ on the subproblem, $\widehat{\boldsymbol{\beta}}_{\tau_c} = (\mathbf{\Gamma}_{\tau_c}^T \mathbf{\Gamma}_{\tau_c})^\dagger \mathbf{\Gamma}_{\tau_c}^T \mathbf{x}_{\tau_\mathbf{c}}$. Algorithm 2 and Figure 2.2 summarize the sequential leveraging sampling algorithm.

Remark 2.3.1. *Pilot subsample.*

The pilot subsample helps decide several hyper-parameters in the SLS. The size of the pilot subsample n_0 depends on the availability of working memory. It will not influence the performance of the SLS if other hyper-parameters are properly chosen. As long as the pilot subsample fits in the working memory, the information contained in the pilot subsample can be used to make immediate decision on the choice of the hyper-parameters.

Once we have collected a pilot subsample, the value of model order p can be decided through methods such as the autocorrelation and partial autocorrelation plots (Box et al., 2011), minimum description length criterion (Rissanen, 2000) or information criterion such as AIC (Akaike, 1998) or BIC (Schwarz et al., 1978). The pilot subsample can also provide a good initial value for iterative optimization algorithms such as the gradient descent or the Newton–Raphson methods in the estimation of $\boldsymbol{\beta}$ (Step 3 of Algorithm 2).

Remark 2.3.2. *Leverage score approximation and the choice of γ.*

Sampling with probabilities proportional to leverage scores of data matrix yields a high precision approximation to the original data matrix (Drineas et al., 2008). However, leverage-based importance sampling is difficult to adapt naturally to data streams because the leverage scores themselves are not easy to compute in a streaming

setting. The computation of exact leverage scores is not only expensive, but also impossible in the streaming setting because the leverage scores depend on all data points, including those that have not yet been observed in the data stream.

An important note is that we can approximate the leverage scores to achieve a similar goal of importance sampling in a streaming setting. The key idea is to sample data points according to their coarse overestimates of the true leverage scores with respect to the stream observed so far. Using such overestimates of leverage scores in sampling has been shown to be successful in a variety of sampling problems (Koutis et al., 2010; Cohen et al., 2015, 2016; Calandriello et al., 2017a,b).

We specify the Bernoulli success probability π_j to be proportional to the true leverage score h_{jj} of the jth data point by choosing $\gamma \geq 1$ so that $\pi_j = \frac{||\mathbf{z}_j||^2}{\gamma K}$, which is an overestimate of the leverage score. Since K is related to the leverage scores of pilot subsample through $(\mathbf{\Gamma}_{n_0}^T \mathbf{\Gamma}_{n_0})^\dagger$, a conservative choice of γ is 1. It will make the π_j's as overestimates of true leverage scores of pilot subsamples. The rationale for matching π_j with leverage h_{jj} score of the jth data point is to exploit the fact that observations with higher leverage score will have a higher probability of being selected as the starting point of our SLS block.

Remark 2.3.3. *Information threshold c.*

The information threshold c is pre-specified by the user. One guidance to the choice of c is related to width of the confidence region for $\boldsymbol{\beta}$. As an example, in AR(1) case, the $(1-2\alpha)$-level confidence interval for β is $\widehat{\beta}_{\tau_c} \pm c^{-1/2}\sigma\Phi^{-1}(1-\alpha)$, where $\Phi(\cdot)$ is the standard normal distribution function (Lai, 2001). In Theorem 2.4.4 and Proposition 2.4.5, we provide the construction of confidence region for SLS esti-

mator $\widehat{\boldsymbol{\beta}}_{\tau_c}$ *for general AR(p) streams. The information threshold c is involved in the construction of fixed width confidence region based on the stopping rule (2.17).*

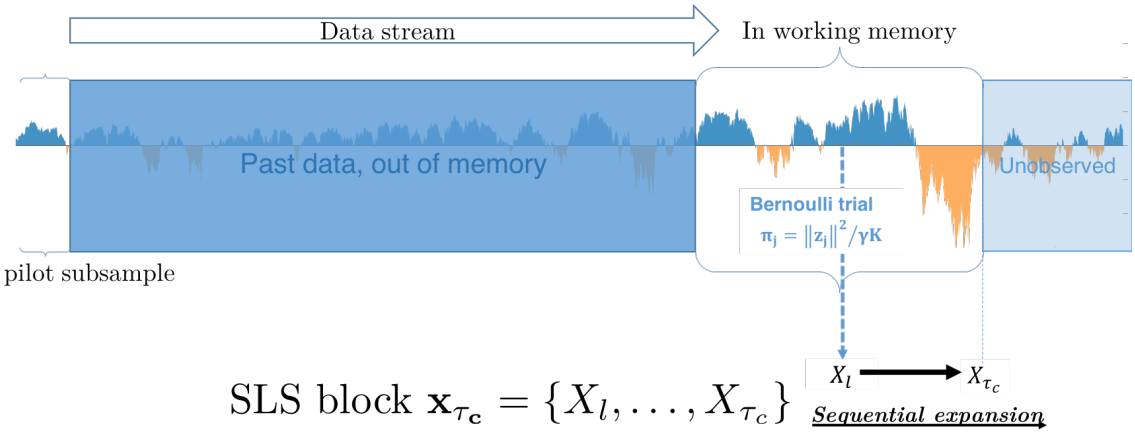

Figure 2.2: An Illustration of SLS Algorithm 2. The sequence labeled with "Data stream" is the streaming time series we are observing. The SLS block, as a subset of the data points in the working memory, with starting point X_l, selected according to leveraged-based independent Bernoulli trial (2.9), and stopping point X_{τ_c} according to sequential stopping rule (2.17).

2.4 Theoretical Results

In this section, we demonstrate that our proposed SLS method is accurate in terms of providing a unified statistical inference for the parameters in an AR model. We state two main theorems which establish a conditional uniform asymptotic normality result for the normalized least squares estimator based on the SLS method introduced in Section 2.3. We first present the main result for a streaming AR(1) series and then the results for the general AR(p) model.

It is well known that, under the regular sampling setting, i.e., when the sample

size is not random, the limit distribution of the least squares estimator of the AR parameter vector drastically changes according to whether the AR(p) series is stable, unstable or explosive; see, for instance, Anderson (1959); Chan and Wei (1988); Jeganathan (1988); Lai and Wei (1983). When the series is unstable, it requires a more complex procedure to obtain a unified asymptotic result, as discussed in Lahiri (2003). For the AR(1) model, Lai and Siegmund (1983) established a unified limiting distribution for non-explosive AR(1) models, if the sample size is determined sequentially using a stopping rule. Later, Galtchouk and Konev (2011) extended this result to the unstable AR(p) series with $p \geq 1$. In order to provide a unified inference for streaming time series data coming from a p-th order autoregressive model (AR(p)) with $p \geq 1$, we study the theoretical properties of our SLS method along the lines of Lai and Siegmund (1983) and Galtchouk and Konev (2011).

2.4.1 Sequential Leveraging for AR(1) Series

In the case of streaming first order autoregressive process AR(1), we have $\mathbf{z}_i = X_{i-1}$. We follow the sequential leveraging algorithm in Section 2.3 to decide the starting time l of the SLS block. This leads us to the sequential sampling rule

$$\tau_c = \inf\{t \geq l : \sum_{i=l}^{t} X_{i-1}^2 \geq c\}, \tag{2.19}$$

where $c(> 0)$ is the information threshold. Note that if ε's are normally distributed, $\sum_{i=l}^{t} X_{i-1}^2$ is the observed Fisher information for the subsample $\{X_l, \ldots, X_t\}$.

Before we state the main theorem, we state in Lemma 2.4.1 below a conditional

martingale central limit theorem (Freedman, 1971, pp. $90-92$).

Lemma 2.4.1. *Let $X_j, \varepsilon_j, j = l-1, l, \ldots$ be random variables adapted to the increasing sequence of σ-algebras $\mathscr{F}_j, j = l-1, l, \cdots$. Let $\{\mathrm{P}_\beta, \beta \in [-1,1]\}$ be a family of conditional probability measures conditioned on $\mathscr{F}_{l-1}$ such that, under every P_β, we have*

$$\varepsilon_l, \varepsilon_{l+1}, \ldots \quad are \quad i.i.d. \quad with \quad E_\beta \varepsilon_l = 0, \quad E_\beta \varepsilon_l^2 = \sigma^2; \tag{2.20}$$

$$\sup_\beta E_\beta[\varepsilon_l^2; |\varepsilon_l| > a] \to 0 \text{ as } a \to \infty; \tag{2.21}$$

$$\varepsilon_j \text{ is independent of } \mathscr{F}_{j-1} \text{ for each } j \geq l; \tag{2.22}$$

$$\mathrm{P}_\beta \left(\sum_{j=l}^{\infty} X_{j-1}^2 = \infty \right) = 1; \tag{2.23}$$

$$\sup_\beta \mathrm{P}_\beta \left(X_j^2 > a \right) \to 0 \quad as \ a \to \infty \quad for \ each \quad j \geq l-1; \tag{2.24}$$

and for each $\zeta > 0$

$$\sup_\beta \mathrm{P}_\beta \left(X_j^2 \geq \zeta \sum_{i=l}^{j} X_{i-1}^2 \text{ for some } j \geq m \right) < \epsilon \text{ for any } \epsilon > 0 \text{ as } m \to \infty. \tag{2.25}$$

Let

$$\tau_c = \inf\{j : \sum_{i=l}^{j} X_{i-1}^2 \geq c\},\ c > 0. \tag{2.26}$$

Then uniformly in $\beta \in [-1, 1]$ *and* $-\infty < t < \infty$*, we have*

$$\lim_{c\to\infty} \sup_{|\beta|\leq 1} \sup_{t\in\mathbb{R}} \left| \mathrm{P}_\beta \left(c^{-1/2} \sum_{j=l}^{\tau_c} X_{j-1}\varepsilon_j \leq t \right) - \Phi(t) \right| = 0. \tag{2.27}$$

Recall that the estimator of β based on the subsample $\{X_l, \cdots, X_{\tau_c}\}$ is given by $\widehat{\beta}_{\tau_c} = \sum_{j=l}^{\tau_c} X_{j-1}X_j / \sum_{j=l}^{\tau_c} X_{j-1}^2$. We now state the main theorem.

Theorem 2.4.2. *If* $\varepsilon_l, \varepsilon_{l+1}, \ldots,$ *are i.i.d. with mean* 0 *and variance* σ^2*, and the sequence* $\{\varepsilon_i : i \geq l\}$ *is independent of* X_{l-1} *(defined in Algorithm 2), then*

$$\lim_{c\to\infty} \sup_{|\beta|\leq 1} \sup_{x\in\mathbb{R}} \left| \mathrm{P}_\beta \left[\sqrt{\sum_{i=l}^{\tau_c} X_{i-1}^2} \left(\widehat{\beta}_{\tau_c} - \beta \right) \leq x \right] - \Phi(x/\sigma) \right| = 0, \tag{2.28}$$

where P_β *is the conditional probability measure defined in Lemma 2.4.1.*

The proofs of Lemma 2.4.1 and Theorem 2.4.2 have been relegated to the Appendix. The proof of Theorem 2.4.2 is along the line of Theorem 2.1 of Lai and Siegmund (1983) with appropriate conditional probability measures. Theorem 2.4.2 provides a uniform asymptotic normality result given the starting time l. That is, based on the uniformity result of (2.28), the sequential leveraging sampling establishes a unified approach for the estimation of β regardless of whether $|\beta| < 1$ or $|\beta| = 1$, which provides a strong large sample result.

2.4.2 Sequential Leveraging for AR(p) Series

In this section, we consider the uniform asymptotic normality properties of SLS least square estimate that is eligible for both the stable and the unstable AR(p) processes. We use the conditions proposed in Galtchouk and Konev (2011) to define the unstable region for the AR(p) processes.

Condition 1. Parameter $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^T$ satisfies that all roots $\lambda_i = \lambda_i(\boldsymbol{\beta})$ of the characteristic polynomial (2.10) lie inside or on the unit circle, which indicates stable or unstable AR(p) processes.

Condition 2. The unit roots $\{\lambda_i = \lambda_i(\boldsymbol{\beta}) : |\lambda_i(\boldsymbol{\beta})| = 1, \boldsymbol{\beta} \in \mathbb{R}^p, i = 1, \ldots, p\}$ are real numbers.

Condition 3. The system of linear equations with respect to $Y_1, Y_2, \ldots, Y_{p-1}$

$$\begin{cases} Y_1 = \beta_1 + \sum_{i=2}^{p} \beta_i Y_{i-1} \\ Y_j = \beta_j + \sum_{k=1}^{j-1} \beta_{j-k} + \sum_{k=1}^{p-j} \beta_{k+j} Y_k, \quad 2 \le j \le p-1 \end{cases} \tag{2.29}$$

has a unique solution $(Y_1, \ldots, Y_{p-1})$ denoted as $Y_i = \kappa_i(\boldsymbol{\beta}), 1 \le i \le p-1$, so that the transformation matrix is positive definite

$$L(\boldsymbol{\beta}) = \begin{bmatrix} 1 & \kappa_1(\boldsymbol{\beta}) & \kappa_2(\boldsymbol{\beta}) & \cdots & \kappa_{p-1}(\boldsymbol{\beta}) \\ \kappa_1(\boldsymbol{\beta}) & 1 & \kappa_1(\boldsymbol{\beta}) & \cdots & \kappa_{p-2}(\boldsymbol{\beta}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \kappa_{p-1}(\boldsymbol{\beta}) & \kappa_{p-2}(\boldsymbol{\beta}) & \cdots & \kappa_1(\boldsymbol{\beta}) & 1 \end{bmatrix}. \tag{2.30}$$

We denote Λ_p as all $\boldsymbol{\beta}$'s satisfying Condition 1 as previously defined, which in-

cludes the stable and unstable cases. We denote $\check{\Lambda}_p$ as all $\boldsymbol{\beta}$'s satisfying both Conditions 1 and 2, which excludes the complex unit roots. Whereas the AR processes with complex unit roots have a persistent cyclical behavior (Bierens, 2001). Lastly, we denote $\tilde{\Lambda}_p$ as all $\boldsymbol{\beta}$'s satisfying Conditions 1,2 and 3, which further excludes the some extreme cases such as the first lag coefficient $\beta_1 = 0$.

Now we establish the main result of this section, which provides the uniform asymptotic normality of the SLS estimator $\boldsymbol{\beta}_{\tau_c}$.

Theorem 2.4.3. *Let $\hat{\boldsymbol{\beta}}_{\tau_c}$ be the least squares estimate of $\boldsymbol{\beta}$ based on design matrix (2.18), τ_c defined in (2.17) and $\mathbf{V}_{\tau_c,\boldsymbol{\beta}} = (\boldsymbol{\Gamma}_{\tau_c}^T \boldsymbol{\Gamma}_{\tau_c})^{1/2}(\widehat{\boldsymbol{\beta}}_{\tau_c} - \boldsymbol{\beta})$. Assume that $\boldsymbol{\beta}$ satisfies Conditions 1, 2 and 3. If $\varepsilon_l, \varepsilon_{l+1}, \ldots,$ are i.i.d. with mean 0 and variance σ^2, and the sequence $\{\varepsilon_i : i \geq l\}$ is independent of the starting state $\mathbf{z}_l = (X_{l-1}, \ldots, X_{l-p})^T$ defined in Algorithm 2, then*

$$\lim_{c\to\infty} \sup_{\boldsymbol{\beta}\in K} \sup_{\boldsymbol{x}\in\mathbb{R}^p} \left|\mathbf{F}_{\mathbf{V}_{\tau_c,\boldsymbol{\beta}}}(\boldsymbol{x}) - \boldsymbol{\Phi}(\boldsymbol{x}/\sigma)\right| = 0, \tag{2.31}$$

for any compact set $K \subset \tilde{\Lambda}_p$.

With the result of Theorem 2.4.3, the following theorem can be proved easily, which is omitted.

Theorem 2.4.4. *Let $\hat{\boldsymbol{\beta}}_{\tau_c}$ be the least squares estimate of $\boldsymbol{\beta}$ based on design matrix (2.18) and τ_c defined in (2.17). Assume that $\boldsymbol{\beta}$ satisfies Conditions 1, 2 and 3. If $\varepsilon_l, \varepsilon_{l+1}, \ldots,$ are i.i.d. with mean 0 and variance σ^2, and the sequence $\{\varepsilon_i : i \geq l\}$ is*

independent of the starting state $\mathbf{z}_l = (X_{l-1}, \ldots, X_{l-p})^T$ *defined in Algorithm 2, then*

$$\frac{1}{\sigma^2}(\widehat{\boldsymbol{\beta}}_{\tau_c} - \boldsymbol{\beta})^T(\boldsymbol{\Gamma}_{\tau_c}^T\boldsymbol{\Gamma}_{\tau_c})(\widehat{\boldsymbol{\beta}}_{\tau_c} - \boldsymbol{\beta}) \to \chi_p^2, \ \ as\ c \to \infty, \tag{2.32}$$

uniformly in $\boldsymbol{\beta} \in K$ *for any compact set* $K \subset \tilde{\Lambda}_p$.

Based on Theorem 2.4.4, for any $d > 0$, let

$$\boldsymbol{R}_n = \{\boldsymbol{\beta} : (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}_n)^T(\boldsymbol{\Gamma}_n^T\boldsymbol{\Gamma}_n)(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}_n) \le d^2\mathrm{tr}(\boldsymbol{\Gamma}_n^T\boldsymbol{\Gamma}_n)\}, \tag{2.33}$$

where $\mathrm{tr}(\boldsymbol{\Gamma}_n^T\boldsymbol{\Gamma}_n)$ is the trace of observed Fisher information matrix $\boldsymbol{J}_n = \boldsymbol{\Gamma}_n^T\boldsymbol{\Gamma}_n$. The ellipsoid defined by $\boldsymbol{R}_n$ has length of the major axis of $2d$, which is in the sense that the size of the ellipsoid is fixed. Now, given any $\alpha \in (0, 1)$, we define

$$n_0(d) = \left\lceil \sigma^2 a^2 / [d^2 \mathrm{tr}(\mathrm{E}_{\boldsymbol{\beta}}(\boldsymbol{\Gamma}_n^T\boldsymbol{\Gamma}_n))] \right\rceil, \tag{2.34}$$

where $\lceil \cdot \rceil$ is the ceiling function, which maps a real number to the least succeeding integer, and a^2 satisfies $\mathrm{P}[\chi_2^2 \le a^2] = 1 - \alpha$, then we have

$$\lim_{d \to 0} \mathrm{P}\left(\boldsymbol{\beta} \in \boldsymbol{R}_{n_0(d)}\right) = 1 - \alpha. \tag{2.35}$$

In practice, we use the sequential stopping rule in (2.17) to determine the stopping time τ_c instead of $n_0(d)$ in (2.34) since $n_0(d)$ depends on the unknown parameters, where $c = \sigma^2 a^2 / d^2$.

The next proposition follows immediately from Theorem 2.4.4, which provides

the confidence region of our SLS estimators $\widehat{\boldsymbol{\beta}}_{\tau_c}$.

Proposition 2.4.5. *Let $\widehat{\boldsymbol{\beta}}_{\tau_c}$ be the least squares estimate of $\boldsymbol{\beta} \in \Lambda_p$ based on design matrix (2.18) and τ_c defined in (2.17). If $\varepsilon_l, \varepsilon_{l+1}, \ldots,$ are i.i.d. with mean 0 and variance σ^2, and the sequence $\{\varepsilon_i : i \geq l\}$ is independent of the starting state $\mathbf{z}_l = (X_{l-1}, \ldots, X_{l-p})^T$ defined in Algorithm 2, then*

$$\limsup_{d \to 0} \sup_{\boldsymbol{\beta} \in K} \mathrm{P}\left(\boldsymbol{\beta} \in \boldsymbol{R}_{\tau_c(d)}\right) = 1 - \alpha, \tag{2.36}$$

where $\boldsymbol{R}_{\tau_c(d)} = \{\boldsymbol{\beta} : (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}_{\tau_c(d)})^T (\boldsymbol{\Gamma}^T_{\tau_c(d)} \boldsymbol{\Gamma}_{\tau_c(d)})(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}_{\tau_c(d)}) \leq d^2 \mathrm{tr}(\boldsymbol{\Gamma}^T_{\tau_c(d)} \boldsymbol{\Gamma}_{\tau_c(d)})\}$.

Proposition 2.4.5 provides a fixed width confidence region of the SLS estimate of $\boldsymbol{\beta} \in \Lambda_p$.

2.5 Simulation Studies

In this section, we demonstrate the empirical performance of the SLS method for the streaming AR(1) and AR(2) data. To verify the asymptotic normality results from Section 2.4, we present comprehensive simulation studies for the synthetic streaming AR(1) and AR(2) time series with various parameter settings.

2.5.1 AR(1) Series

Model specification. We present some simulation studies to illustrate the validity of the asymptotic normality result stated in Theorem 2.4.2. We generate the data stream from an AR(1) model with five different values of β gradually changing from

stable to unstable cases, i.e., $\beta = 0.2, 0.5, 0.9, 0.99, 1$. The innovations are generated from t-distribution with 4 degrees of freedom. We collect a pilot subsample of size $n_0 = 50$, and compute $K = \sum_{i=1}^{n_0} X_{i-1}^2$. We choose the value of γ so that the independent Bernoulli trial success probability $\pi_j = X_{j-1}^2/\gamma K$ is close to the leverage score h_{jj} computed using $100,000$ (additionally) simulated values from an AR(1) series. For the AR(1) model, the leverage scores are $h_{ii} = X_{i-1}^2/\sum_{i=2}^{N} X_{i-1}^2 \in [0,1]$, $i = 2, \ldots, N$ for a fixed sample size N. The values of information threshold c are reported in Table 2.1. There are $1,000$ replications for each setting of the parameter, i.e. $n_{rep} = 1000$.

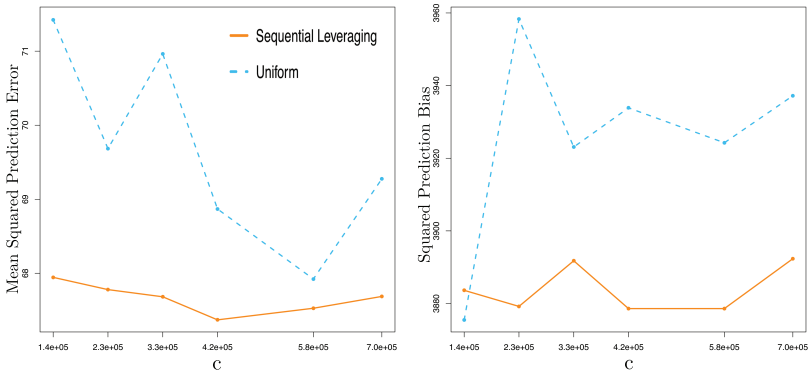

Figure 2.3: AR(1) synthetic data with $\beta = 1$. **MSPE** (left) and **bias**^{2} (right) of AR(1) model on the test data for SLS (solid) and uniform subsample (dashed) at different information threshold c levels.

Assessment criteria. We follow Algorithm 2 and the stopping rule in (2.19) with information threshold c to collect a SLS block $\{X_l, \ldots, X_{\tau_c}\}$ from the respective streaming time series. The assessment of the parameter estimation based on Theorem 2.4.2 consists of computing frequency count of the number of times the normalized estimator, $\sqrt{\sum_{i=l}^{\tau_c} X_{i-1}^2}\left(\widehat{\boldsymbol{\beta}}_{\tau_c} - \boldsymbol{\beta}\right)/\sigma > z$ or $\sqrt{\sum_{i=l}^{\tau_c} X_{i-1}^2}\left(\widehat{\boldsymbol{\beta}}_{\tau_c} - \boldsymbol{\beta}\right)/\sigma < -z$

for standard normal quantiles z, where σ is the standard deviation of the innovation ε. In Table 2.1, we set $z = 1.28$, for which the (one-tailed) probabilities are nominally $p_r = p_l = 0.10$. We also carry out simulation studies for other values of z but do not report it here. In addition, we assess the prediction error based on the parameter estimation using the mean squared prediction error, MSPE $= \frac{1}{n_{rep}} \sum_{i=1}^{n_{rep}} ||\mathbf{\Gamma}_{\text{test}} \hat{\boldsymbol{\beta}}_i - \mathbf{x}_{\text{test}}||^2$, and the corresponding squared prediction bias, bias$^2 = ||\frac{1}{n_{rep}} \sum_{i=1}^{n_{rep}} (\mathbf{\Gamma}_{\text{test}} \hat{\boldsymbol{\beta}}_i - \mathbf{x}_{\text{test}})||^2$, on the additional test data $\mathbf{x}_{\text{test}}$ of length $2,000$.

Comparison of methods. Table 2.1 summarizes the simulation results. In Table 2.1, "Uniform Sequential Sampling" refers to sequential sampling with starting points chosen randomly with equal probability (Bernoulli trials with equal probability), and "Fixed Length Sampling" refers to fixed subsample size sampling starting at $t = n_0 + 1$. By comparing with uniform sequential sampling, we demonstrate the advantage of leverage-based independent Bernoulli trials for choosing the starting points as an online algorithm; while by comparing with the fixed length sampling, we illustrate the efficiency of the SLS method.

Discussion. Table 2.1 shows that for each value of β, the SLS performs as well or better than other methods, and the fixed length sampling is even worse especially for $|\beta|$ near 1. It is important to note from Table 2.1 we see that the average sequential subsample size, $E_\beta N_c$, is significantly smaller for the SLS block than that of the uniform sequential sampling. It demonstrates the efficiency of the SLS from the perspective of subsample size. For each setting of β, the case that has smaller average subsample size has been highlighted in column $\text{E}_\beta N_c$. Nevertheless, the SLS

yields the same or more accurate right and left tail estimated probabilities ($\hat{p_r}$ and $\hat{p_l}$) than those of the uniform sequential sampling and the fixed length sampling. In terms of prediction error, results of MSPE and bias^2 also demonstrate that our SLS method outperforms the uniform sequential sampling, where Figure 2.3 shows a particular case when $\beta = 1$ with six increasing values of c.

Table 2.1: Sequential Leveraging Sampling of AR(1) with Varying Information Threshold c

	No.	β	c	$\hat{p}_r$	$\hat{p}_l$	$\mathrm{E}_\beta N_c$	$sd_\beta(N_c)$
	1	0.2	500	0.09	0.09	**235.76**	55.55
	2	0.5	800	0.12	0.10	**291.61**	70.36
SLS	*3*	0.9	3500	0.12	0.08	**328.68**	98.52
(Online)	*4*	0.99	20000	0.12	0.11	**236.42**	158.31
	5	1	1.5×10^6	0.09	0.10	**146.66**	336.59
	6	1	2.8×10^6	0.09	0.11	**209.93**	477.19
	1	0.2	500	0.11	0.10	248.50	47.70
Uniform	*2*	0.5	800	0.09	0.11	309.73	55.91
Sequential	*3*	0.9	3500	0.09	0.10	348.39	89.15
Sampling	*4*	0.99	20000	0.10	0.11	277.28	153.74
(Online)	*5*	1	1.5×10^6	0.08	0.09	450.06	731.46
	6	1	2.8×10^6	0.11	0.12	643.22	1018.08
	No.	β	c	$\hat{p}_r$	$\hat{p}_l$	N	
	1	0.2	500	0.09	0.11	236	
Fixed	*2*	0.5	800	0.10	0.10	292	
Length	*3*	0.9	3500	0.08	0.11	329	
Sampling	*4*	0.99	20000	0.04	0.15	236	
(Offline)	*5*	1	1.5×10^6	0.08	0.11	146	
	6	1	2.8×10^6	0.08	0.13	210	

NOTE: The columns $\hat{p_r}$ and $\hat{p_l}$ give the estimated percentage of excesses in the right and left tails of the distributions. The columns $\mathrm{E}_\beta N_c$ and $sd_\beta(N_c)$ report the average and standard deviation of the subsample size, respectively. The case that has smaller average subsample size in each setting has been highlighted in column $\mathrm{E}_\beta N_c$.

2.5.2 AR(2) Series

Model specification. The simulation studies for the AR(2) streaming data are reported in Table 2.2 to check whether the asymptotic normality result proposed in Theorem 2.4.3 and Theorem 2.4.4 are valid. We generate the data stream from an AR(2) model with 14 different values of $\boldsymbol{\beta}$ that cover various stable and unstable cases (See Figure 2.4, the Nos. *1*, *4*, *5*, *8*, *11* and *12* are stable cases, while Nos. *2*, *3*, *6*, *7*, *9*, *10*, *13* and *14* are unstable cases). The innovations are generated from t ($df = 4$) distribution. We collect the pilot subsample of size $n_0 = 50$, and compute $K = \sum_{i=1}^{n_0}(X_{i-1}^2 + X_{i-2}^2)$. We choose the value of γ carefully so that the independent Bernoulli trial success probability $\pi_j = \frac{X_{j-1}^2 + X_{j-2}^2}{\gamma K}$ is approximated to the exact leverage scores h_{jj} computed by simulating additionally $50,000$ values from an AR(2) series. For the AR(2) model, it can be shown that the leverage scores are:

$$h_{ii} = \frac{X_{i-1}^2 \sum X_{i-2}^2 - 2X_{i-1}X_{i-2}\sum X_{i-1}X_{i-2} + X_{i-2}^2 \sum X_{i-1}^2}{\sum X_{i-2}^2 \sum X_{i-1}^2 - (\sum X_{i-1}X_{i-2})^2} \approx \frac{X_{i-1}^2 + X_{i-2}^2}{\gamma \sum_{i=1}^{n_0}(X_{i-1}^2 + X_{i-2}^2)} \tag{2.37}$$

with a scale parameter $\gamma > 0$ and $i = 2, \ldots, N$ for a fixed sample size N. There are $1,000$ replications for each $\boldsymbol{\beta}$ and c, i.e. $n_{rep} = 1000$.

Table 2.2 summarizes the simulation results, where the notation is the same as the AR(1) case. The parameter setting of $\boldsymbol{\beta}$ is denoted in column "No." of Table 2.2 and Figure 2.4 correspondingly. The stable region of the AR(2) process is inside the triangular area in Figure 2.4, and the boundary $-2 < \beta_1 < 2,\ \beta_2 = -1$ (the bottom side (blue) of the triangle in Figure 2.4) defines the unstable region that satisfies Condition 1 - 3.

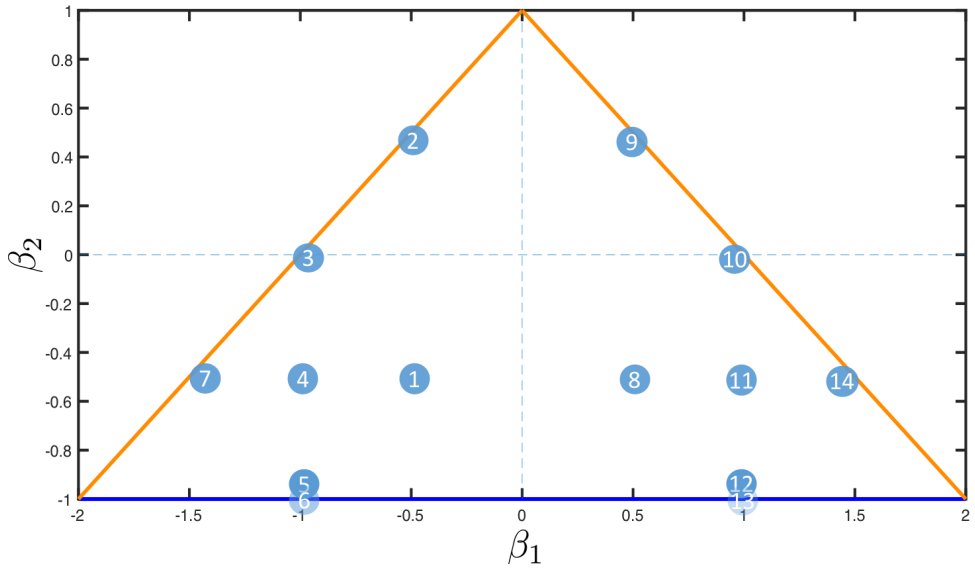

Figure 2.4: An Illustration of Simulation Cases in Table 2.2 on Different Region for the AR(2) process. The stable AR(2) process requires that the roots of (2.10) have to be inside the triangle, while the unstable cases are on the boundary of the triangle. Particularly, the Nos. *1*, *4*, *5*, *8*, *11* and *12* are stable cases, while Nos. *2*, *3*, *6*, *7*, *9*, *10*, *13* and *14* are unstable cases.

Assessment criteria. We follow Algorithm 2 and the stopping rule in (2.17) with respective information threshold c to take a SLS block $\{X_l, \ldots, X_{\tau_c}\}$ from the corresponding streaming time series. Instead of measuring the multidimensional asymptotic normality of $(\mathbf{\Gamma}_{\tau_c}^T \mathbf{\Gamma}_{\tau_c})^{1/2}(\widehat{\boldsymbol{\beta}}_{\tau_c} - \boldsymbol{\beta})$, we transform the quantity of interest into $\frac{1}{\sigma^2}(\widehat{\boldsymbol{\beta}}_{\tau_c} - \boldsymbol{\beta})^T(\mathbf{\Gamma}_{\tau_c}^T \mathbf{\Gamma}_{\tau_c})(\widehat{\boldsymbol{\beta}}_{\tau_c} - \boldsymbol{\beta})$, which converges to chi-squared distribution with 2 degrees of freedom ($\chi^2(df = 2)$) asymptotically. The closeness between distributions of $\frac{1}{\sigma^2}(\widehat{\boldsymbol{\beta}}_{\tau_c} - \boldsymbol{\beta})^T(\mathbf{\Gamma}_{\tau_c}^T \mathbf{\Gamma}_{\tau_c})(\widehat{\boldsymbol{\beta}}_{\tau_c} - \boldsymbol{\beta})$ and $\chi^2(df = 2)$ distribution is measured through Kullback-Leibler divergence and tail probability coverages. The Kullback-Leibler

(KL) divergence of distribution Q from distribution P is defined as the integral $\int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx$, where p and q denote the densities of P and Q. In addition, we assess the fitted error based on the parameter estimation using the MSPE and the corresponding bias^2 on an additional test data $\mathbf{x}_{\text{test}}$ of length $2,000$.

Table 2.2: Sequential Leveraging Sampling of AR(2) with Varying Information Threshold c

	No.	β_1	β_2	c	KL	$\mathrm{E}_{\boldsymbol{\beta}} N_c$	$sd_{\boldsymbol{\beta}}(N_c)$
	1	-0.50	-0.50	1000	**0.96**	**165.79**	44.07
	2	-0.50	0.49	19000	0.98	**112.22**	107.43
	3	-1.00	-0.01	30000	**1.07**	**88.39**	85.46
SLS	*4*	-1.00	-0.50	1000	1.07	**105.41**	33.76
(Online)	*5*	-1.00	-0.99	35000	**0.97**	**135.83**	102.01
	6	-1.00	-1.00	1500000	**1.01**	355.98	373.51
	7	-1.49	-0.50	40000	0.99	**86.07**	71.26
	8	0.50	-0.50	600	0.97	**102.23**	31.32
	9	0.50	0.49	18000	**0.98**	**137.94**	126.94
	10	1.00	-0.01	18000	1.08	**69.90**	72.16
	11	1.00	-0.50	2900	**1.05**	**298.38**	67.30
	12	1.00	-0.99	29000	1.06	**108.92**	85.93
	13	1.00	-1.00	780000	**0.97**	321.35	314.85
	14	1.49	-0.50	50000	1.03	**140.37**	92.25
	1	-0.50	-0.50	1000	1.13	174.28	41.66
	2	-0.50	0.49	19000	**0.97**	232.61	143.09
Uniform	*3*	-1.00	-0.01	30000	1.08	219.89	139.61
Sequential	*4*	-1.00	-0.50	1000	1.09	111.37	31.18
Sampling	*5*	-1.00	-0.99	35000	1.06	192.68	116.93
(Online)	*6*	-1.00	-1.00	1500000	1.19	**105.66**	297.47
	7	-1.49	-0.50	40000	1.07	146.35	85.37
	8	0.50	-0.50	600	1.09	105.56	29.62
	9	0.50	0.49	18000	1.03	215.54	143.16
	10	1.00	-0.01	18000	1.08	148.40	98.40
	11	1.00	-0.50	2900	1.09	310.02	66.22
	12	1.00	-0.99	29000	1.10	159.23	101.52
	13	1.00	-1.00	780000	1.12	**97.30**	210.69
	14	1.49	-0.50	50000	1.05	173.28	90.87
	No.	β_1	β_2	c	KL	N	
	1	-0.50	-0.50	1000	1.05	166	
	2	-0.50	0.49	19000	1.06	112	
Fixed	*3*	-1.00	-0.01	30000	1.14	88	
Length	*4*	-1.00	-0.50	1000	**1.02**	105	
Sampling	*5*	-1.00	-0.99	35000	1.09	136	
(Offline)	*6*	-1.00	-1.00	1500000	1.02	356	
	7	-1.49	-0.50	40000	**0.98**	86	
	8	0.50	-0.50	600	**0.94**	102	
	9	0.50	0.49	18000	1.08	138	
	10	1.00	-0.01	18000	**0.98**	70	
	11	1.00	-0.50	2900	1.08	298	
	12	1.00	-0.99	29000	**1.04**	109	
	13	1.00	-1.00	780000	1.02	321	
	14	1.49	-0.50	50000	**1.00**	140	

NOTE: The values of $\boldsymbol{\beta}$ for the AR(2) model that cover various stable and unstable cases (See Figure 2.4, the Nos. *1*, *4*, *5*, *8*, *11* and *12* are stable cases, while Nos. *2*, *3*, *6*, *7*, *9*, *10*, *13* and *14* are unstable cases). The column KL reports the Kullback-Leibler divergence of distributions of $\frac{1}{\sigma^2}(\widehat{\boldsymbol{\beta}}_{\tau_c} - \boldsymbol{\beta})^T(\boldsymbol{\Gamma}_{\tau_c}^T \boldsymbol{\Gamma}_{\tau_c})(\widehat{\boldsymbol{\beta}}_{\tau_c} - \boldsymbol{\beta})$ from the $\chi^2(df = 2)$ distributions. The case that has smaller Kullback-Leibler divergence has been highlighted in column KL. The columns $\mathrm{E}_{\boldsymbol{\beta}} N_c$ and $sd_{\boldsymbol{\beta}}(N_c)$ report average and standard deviation of sequential subsample size N_c, respectively. The case that has smaller average subsample size in each setting has been highlighted in column $\mathrm{E}_{\beta} N_c$.

Discussion. Table 2.2 shows that, for most of the settings of $\boldsymbol{\beta}$, the SLS outperforms uniform sequential sampling and fixed length sampling no matter $\boldsymbol{\beta}$ is in the stable or unstable region. The KL divergence of our method is smaller than or as good as other methods. In most settings of $\boldsymbol{\beta}$, the distributions of $\frac{1}{\sigma^2}(\widehat{\boldsymbol{\beta}}_{\tau_c} - \boldsymbol{\beta})^T(\boldsymbol{\Gamma}_{\tau_c}^T\boldsymbol{\Gamma}_{\tau_c})(\widehat{\boldsymbol{\beta}}_{\tau_c} - \boldsymbol{\beta})$ is closer to theoretical value of $\chi^2(df = 2)$ distributions than other two methods. It is important to note from Table 2.2 that the average sequential subsample size, $E_{\boldsymbol{\beta}}N_c$, is significantly smaller for the SLS (as we highlighted) than that of the uniform sequential sampling for most of the cases. The saving of subsample size for the sequential leveraging sampling, i.e. the efficiency of SLS, is especially obvious when the processes come from the unstable region. The standard deviation, $sd_{\boldsymbol{\beta}}(N_c)$, of subsample size is, in most cases, also smaller for the SLS than that of the uniform sequential sampling. Similarly, results of MSPE and bias^2 shown in Figure 2.5 demonstrate that, as an example, when $\beta_1 = 1,\ \beta_2 = -1$ with 14 increasing values of c, our SLS method outperforms the uniform sequential sampling in prediction error.

2.6 Real Data Analysis

In this section, we apply the SLS method to two real data sets, the Internet traffic data from Cortez et al. (2012) and the seismic data from Chen et al. (2016). We treat the two datasets as if they are streaming time series data for demonstration purpose even though all the observations are available. Such treatment is reasonable because Internet traffic and seismic wave are data streams in real world application

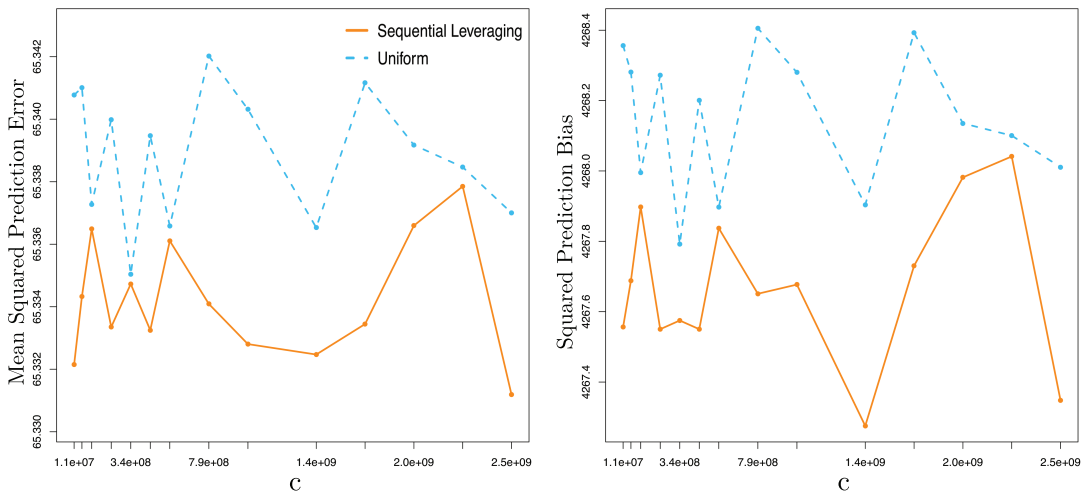

Figure 2.5: AR(2) synthetic data with $\beta_1 = \beta_2 = -1$. **MSPE** (left) and **bias**^{2} (right) of AR(2) model on the test data for SLS (solid) and uniform subsample (dashed) at different information threshold c levels.

and are usually analyzed in an online fashion.

2.6.1 Internet Traffic Data

For data demonstrated in Example 2.1.1, detrending seasonality is needed so that we have an unstable or stable streaming time series satisfying Conditions 1, 2 and 3 in Section 2.4.2. To get rid of possible complex unit roots specified in Condition 2, we need to remove the possible cyclical part caused by seasonality. Based on the information collected in the pilot subsample, seasonal differencing is applied in real time to remove the daily and weekly seasonality, resulting in a time series of length $17,584$ (See Figure 2.1). Since the network traffic appears unstable at multi-second

time scales (Karagiannis et al., 2004), the uniformity of our SLS method is well suited for the Internet traffic stream. We focus on the estimation of the first order partial autocorrelation. and thus fit the AR(1) model. We partition the Internet traffic data into training set and test set. The training set, which contains the first $15,584$ time points, is used to estimate the model parameter β through our SLS method and the uniform sequential sampling method. The test set, which contains the last $2,000$ time points, is used to evaluate the estimation accuracy via the prediction errors.

We perform the SLS method for the AR(1) model 200 times to get sketches of the stream with $\gamma = 2$ and $n_0 = 100$. We choose 11 different values of information threshold c ranging from 1.4×10^9 to 2×10^{10}. The starting points of SLS blocks spread out the stream due to the independent Bernoulli random starting mechanism. The right panel of Figure 2.6 shows the histogram of the SLS estimates of the AR(1) parameter. We observe that most of the estimates of the AR(1) parameter are around the neighborhood of 1. This is consistent with fact of the nonstationarity of the network traffic (Cao et al., 2001; Cortez et al., 2012; Leland et al., 1993).

The prediction error on test data is used to evaluate the empirical performance of the SLS method. In the Internet traffic data, we focus on predicting the future value based on observed data so that we can monitor the Internet traffic and detect the possible threat in real time. To evaluate the accuracy of our parameter estimation, we calculate the MSPE and bias^2 on the test data with $n_{rep} = 200$ replicates.

As illustrated in Section 2.5, the MSPE and bias^2 on test data have similar trends as the parameter estimation, which can be used as a valid performance assessment of sampling methods. In Figure 2.6, the MSPE and bias^2 of SLS method are consistently

lower than those of the uniform sequential sampling. The average subsample size for SLS blocks is 518.3 with standard deviation 349.7, which is much smaller than the corresponding average (standard deviation) subsample size 1616.1(1450.7) for uniform sequential sampling. We see that our SLS method outperforms the uniform sampling with higher prediction accuracy and smaller subsample size.

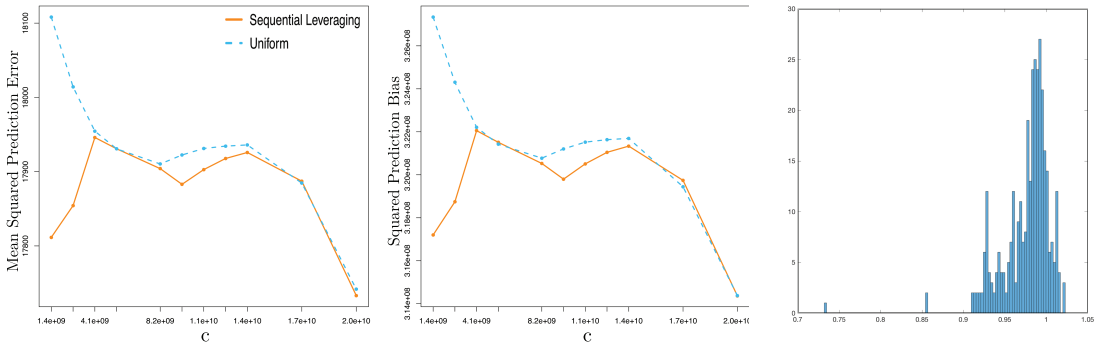

Figure 2.6: Internet Traffic Data. **MSPE** (left) and **bias**^{2} (middle) of AR(1) model on the test data for SLS (solid) and uniform subsample (dashed) at different information threshold c levels. Right: **Histogram** of the AR(1) model parameter estimation for SLS method.

2.6.2 Seismic Data

Seismic data is the recording of earth motion as a function of time that provides the "time snapshot" of subsurface structure. Seismic waves have been continuously recorded since the early 20th century with high acquisition frequency. In exploration geophysics, seismic data is usually continuously acquired every 1 to 4 milliseconds, i.e. 1000 to 250 Hz frequency respectively, and is continuously recorded (Yilmaz, 2001), which results in a huge quantity of data. That amount of data is beyond

the current storage capability for most of the mobile or personal computing devices. Thus, seismic data is naturally treated as streaming time series data in the field works, which needs to utilize the online algorithms for collecting and processing data samples. An efficient sampling algorithm that can automatically capture possible seismic events or ambient vibrations is preferred over the simple uniform sampling. The seismic data can be modeled as an AR(p) process and the model parameter

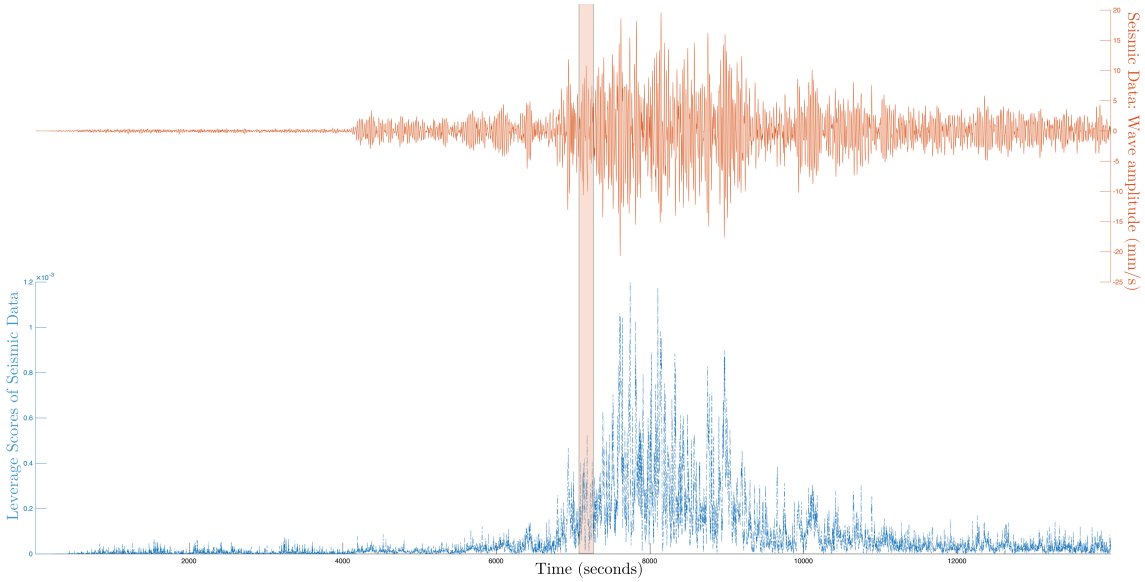

Figure 2.7: An illustration of streaming seismic data that has one seismic event. Top: Scatter plot of the seismic data (solid line). Bottom: Corresponding leverage scores for the AR(4) model (dash-dot line). As an example, the highlighted area indicates one of the 100 SLS blocks.

$\boldsymbol{\beta}$ can help us understand the properties of earthquake source and seismic wave prorogation. The model parameter $\boldsymbol{\beta}$ is mainly influenced by the earthquake focal mechanisms and the seismic wave propagation (Isacks and Molnar, 1971; Gephart and Forsyth, 1984). Thus the estimation of model parameter is essential in the

analysis of seismic data. However, in the geophysics community, the AR model and parameter estimation are rarely used under the current data acquisition system. With large sample size, high acquisition frequency and possible high dimensionality of the model, the computational cost of analyzing such seismic data is very high. Especially in the field, the real time analysis is required but computing power of the portable battery powered device is limited. The sampling method thus is needed to reduce the sample size as well as keep the parameter estimation accuracy. Due to the nature of seismic data, the first few autoregressive parameters is mainly determined by the seismic events, such as the earthquakes and ambient vibrations. In order to capture the seismic events in the subsamples, the leverage-based sampling approach is preferred here.

The seismic data analyzed in this section was well-recorded earthquake sequences (wave amplitude, mm/s) in Oklahoma that were collected on October 26, 2014. We refer Chen et al. (2016) for details of the seismic data. The total sample size for the earthquake sequence is $16,000$. The seismic data is modeled as an AR(p) process with p chosen to be 4 ($p = 4$). From Figure 2.7 , the plot of leverage scores of the seismic data (lower part, dash-dot line), we observe that the one seismic event is clearly illustrated by the picks. The starting point sampling probability constructed based on leverage scores will boost the sequential leveraging method to capture the seismic events.

In our analysis, we treated the seismic data as streaming time series and divided the data into training set and test set. The training set, which contains the first $14,000$ time points, is used to estimate the model parameters using our online se-

quential leveraging sampling method and the uniform sequential sampling method. The uniform sequential sampling refers to sequential sampling with starting points chosen randomly with equal probability (Bernoulli trials with equal probability). The test set, which contains the last 2000 time points, is used to evaluate the estimation accuracy via the prediction errors.

We perform the SLS method for the AR(4) model 100 times to get sketches of the seismic stream with $\gamma = 50,000$ and $n_0 = 200$. There are 11 different values, ranging from 1.2×10^3 to 2.5×10^3, of information threshold c evaluated, and each setting of c is calculated with $n_{rep} = 100$ independent replicates. The SLS results are demonstrated in the Figure 2.8 comparing with uniform sequential sampling, where the MSPE and bias^2 are plotted.

The MSPE and bias^2 (Figure 2.8) of the SLS method are consistently lower than those of the uniform sequential sampling at all information threshold levels. The average subsample size for SLS blocks is 328.5 with the standard deviation 156.5, which is much smaller than the corresponding average (standard deviation) subsample size 1560.3 (1448.4) for uniform sequential sampling. We see that our SLS method outperforms the uniform sampling with higher prediction accuracy and smaller subsample size.

2.7 Summary

In this article, we present an online sampling method, the Sequential Leveraging Sampling (SLS), for streaming time series data. The SLS takes one block of consec-

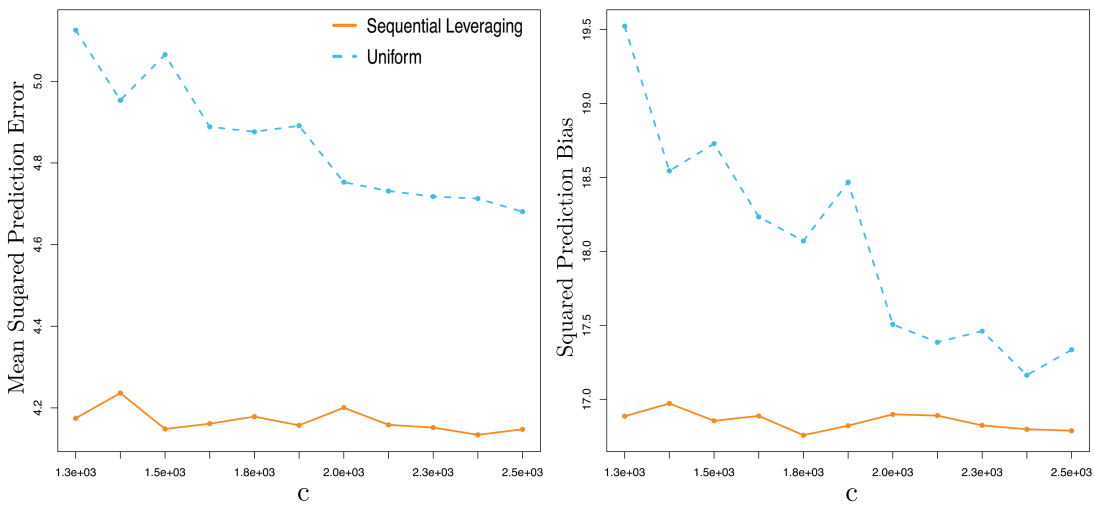

Figure 2.8: Seismic Data. **MSPE** (left) and **bias**^{2} (right) of AR(4) model on the test data for SLS (solid) and uniform subsample (dashed) at different information threshold c levels.

utive time points, called SLS block, as a subsample/snapshot of the streaming data, which maintains the dependence structure of the subsampled data. The sampling algorithm consists of choices of starting time and stopping time. The starting time of the subsample is decided according to leverage scores of the streaming data, which captures the influential time points of the data stream. The stopping time is chosen according to the sequential stopping rule, which provides the theoretical guarantee of estimate properties of the time series model base on the SLS block. The simulation and real data analysis reveal that our method is capable of processing the streaming times series data in real time.

The proposed SLS method establishes an example of using leverage scores as

importance sampling distribution in dependent data. It provides an efficient and accuracy alternative to the simple uniform sampling for time series data. The algorithm design of the SLS method and its empirical performance in saving sequential subsample size also demonstrate the advantage of our method in an online analysis for streaming data. As a natural generalization, the SLS can be applied to multivariate time series model and varying coefficient AR model (Rao, 1970; Hallin, 1978; Dahlhaus et al., 1997; Dahlhaus and Giraitis, 1998; Tiao and Tsay, 1989). The idea of the leverage-based importance sampling can also be extended into the analysis of massive dependent data, for example, spatial or spatio-temporal data. The leverage-based method also has the potential to be incorporated into the study of complex data including nonparametric regression, kernel learning or matrix approximation problems.

2.8 Proofs of Theorem

Lemma 2.8.1 states the martingale central limit theorem from (Freedman, 1971, pages $90-92$) and (Lai and Siegmund, 1983, Lemma 2.1).

Lemma 2.8.1. *Let* $\{u_j, \mathscr{F}_j, j \geq 0\}$ *be a martingale difference sequence, and for* $0 < \zeta < 1$ *and* $r > 0$*, if*

$$|u_j| \leq \zeta \quad \textit{for all} \quad j \tag{2.38}$$

and

$$\mathrm{P}\left(\sum_{j=1}^{\infty} E(u_j^2|\mathscr{F}_{j-1}) > r\right) = 1, \tag{2.39}$$

let

$$\tau = \inf\{j : \sum_{i=1}^{j} E(u_i^2|\mathscr{F}_{i-1}) \geq r\}, \tag{2.40}$$

then there exists a function $\rho : (0,\infty) \to [0,2]$ *that is independent on the distribution of the martingale difference sequence, such that* $\lim_{x\to 0}\rho(x) = 0$ *and*

$$\sup_x |\mathrm{P}(\sum_{i=1}^{\tau} u_i \leq x) - \Phi(x/r^{1/2})| \leq \rho(\zeta/r^{1/2}), \tag{2.41}$$

where Φ *is the standard normal distribution function.*

Lemma 2.8.2 (Lai and Siegmund, 1983, Lemma 2.2) states that the martingale $\sum_{j=l}^{\tau_c} X_{j-1}\varepsilon_j$ is bounded above, which provides a connection between Lemma 2.4.1 and Theorem 2.4.2.

Lemma 2.8.2. *Provided that the assumptions (2.20) and (2.22) hold, for every* $\kappa > 1/2, \delta > 0$, *and a sequence of increasing positive number* $c_\tau \to \infty$, *we have*

$$\sup_\beta \mathrm{P}_\beta \left(|\sum_{i=l}^{\tau} X_{i-1}\varepsilon_i| \geq \delta \max(c_\tau, (\sum_{i=l}^{\tau} X_{i-1}^2)^\kappa) \text{ for some } \tau \geq m \right) \to 0, \qquad (2.42)$$

as $m \to \infty$.

2.8.1 Proof of Lemma 2.4.1

Proof

We prove the Lemma by reducing it to martingale central limit theorem as stated in Lemma 2.8.1. Without loss of generality, we assume that $\mathrm{E}_\beta \varepsilon_l^2 = 1$. To construct the bounded martingale difference sequence, we define a bounded sequence $\{\tilde{X}_t : \tilde{X}_t^2 = \min\{X_t^2, \delta^2 c\},\ 0 < \delta < 1\}$ and denote

$$\Omega_c = \{X_t = \tilde{X}_t \text{ for all } l - 1 \leq t < \tau_c\}, \qquad (2.43)$$

where τ_c is defined in (2.26). Then for all β,

$$\begin{aligned}
&\mathrm{P}_\beta \left(X_t \neq \tilde{X}_t \text{ for some } l - 1 \leq t < \tau_c \right) \\
&\quad \leq \sum_{i=l}^{m} \mathrm{P}_\beta (X_{i-1}^2 > \delta^2 c) + \mathrm{P}_\beta \left\{ \tau_c > m, X_t \neq \tilde{X}_t \text{ for some } m \leq t < \tau_c \right\} \\
&\quad \leq \sum_{i=l}^{m} \mathrm{P}_\beta (X_{i-1}^2 > \delta^2 c) + \mathrm{P}_\beta \left\{ X_t^2 \geq \delta^2 \sum_{i=l}^{t} X_i^2 \text{ for some } m \leq t \right\}.
\end{aligned}$$

For arbitrary large number m, from condition (2.23), $\mathrm{P}_\beta(\tau_c < \infty) = 1$; given m large

enough, by (2.25), choosing c large enough and use (2.24), we get, for all $\beta \in [-1, 1]$, $\mathrm{P}_\beta(\lim_{c\to\infty} \tau_c = \infty) = 1$ and

$$\mathrm{P}_\beta\left(X_t \neq \tilde{X}_t \text{ for some } l-1 \leq t < \tau_c\right) < \delta + \delta = 2\delta, \tag{2.44}$$

or equivalently,

$$\mathrm{P}_\beta(\Omega_c) \geq 1 - 2\delta. \tag{2.45}$$

Now based on sequence $\{\tilde{X}_t\}$, we define $\tilde{\varepsilon}_t = \varepsilon_t I_{\{|\varepsilon_t| \leq \delta^{-1/2}\}}$ and further denote $\bar{\varepsilon}_t = \varepsilon_t - \tilde{\varepsilon}_t$, then by construction, $c^{-1/2}\tilde{X}_{t-1}(\tilde{\varepsilon}_t - \mathrm{E}_\beta\tilde{\varepsilon}_t)$ is a bounded martingale difference sequence for all $t \geq l$

$$|c^{-1/2}\tilde{X}_{t-1}(\tilde{\varepsilon}_t - \mathrm{E}_\beta\tilde{\varepsilon}_t)| \leq 2\delta^{1/2}. \tag{2.46}$$

Moreover, by (2.23), we have

$$\mathrm{P}_\beta\left(\sum_{t=l}^{\infty} \tilde{X}_{t-1}^2 = \infty \Big| \mathscr{F}_{t-1}\right) = 1; \tag{2.47}$$

and choosing δ small enough, by (2.21), we have

$$\mathrm{Var}_\beta(\tilde{\varepsilon}_t) \to 1 \quad \text{as } \delta \to 0. \tag{2.48}$$

Thus

$$\mathrm{P}_\beta\left(\sum_{t=l}^{\infty} \mathrm{E}_\beta\left[c^{-1/2}\tilde{X}_{t-1}(\tilde{\varepsilon}_t - \mathrm{E}_\beta\tilde{\varepsilon}_t)\Big|\mathscr{F}_{t-1}\right]^2 > \mathrm{Var}_\beta(\tilde{\varepsilon}_t)\right) = 1, \tag{2.49}$$

uniformly in $\beta \in [-1, 1]$. Now we are ready to apply Lemma 2.8.1, if

$$\begin{aligned}\tau_c &= \inf\{j \geq l : \sum_{t=l}^{j} \mathrm{E}_\beta \left[c^{-1/2} \tilde{X}_{t-1}(\tilde{\varepsilon}_t - \mathrm{E}_\beta \tilde{\varepsilon}_t) \Big| \mathscr{F}_{t-1} \right]^2 \geq \mathrm{Var}_\beta(\tilde{\varepsilon}_t)\} \\ &= \inf\{j \geq l : \sum_{t=l}^{j} \tilde{X}_{t-1}^2 \geq c\},\end{aligned}$$

then uniformly in $\beta \in [-1, 1]$

$$\begin{aligned}\sup_\beta \sup_{x \in \mathbb{R}} \left| \mathrm{P}_\beta \left(\sum_{t=l}^{\tau_c} c^{-1/2} \tilde{X}_{t-1}(\tilde{\varepsilon}_t - \mathrm{E}_\beta \tilde{\varepsilon}_t) \leq x \right) - \Phi\left(x / [\mathrm{Var}_\beta(\tilde{\varepsilon}_t)]^{1/2} \right) \right| \\ \leq \rho\left(2(\delta / \mathrm{Var}_\beta(\tilde{\varepsilon}_t))^{1/2} \right) \to 0 \quad \text{as } \delta \to 0.\end{aligned} \tag{2.50}$$

Further more, by Wald's identity (Chow et al., 1971, page 23) and Equation (2.15) of Lai and Siegmund (1983) , we can establish the relation between $\sum_{t=l}^{\tau_c} c^{-1/2} \tilde{X}_{t-1}(\tilde{\varepsilon}_t - \mathrm{E}_\beta \tilde{\varepsilon}_t)$ and $\sum_{j=l}^{\tau_c} X_{j-1}\varepsilon_j$ on Ω_c

$$c^{-1/2} \left| \sum_{t=l}^{\tau_c} \tilde{X}_{t-1}(\tilde{\varepsilon}_t - \mathrm{E}_\beta \tilde{\varepsilon}_t) - \sum_{j=l}^{\tau_c} X_{j-1}\varepsilon_j \right| = c^{-1/2} \left| \sum_{t=l}^{\tau_c} \tilde{X}_{t-1}(\bar{\varepsilon}_t - \mathrm{E}_\beta \bar{\varepsilon}_t) \right| \to 0 \tag{2.51}$$

as $\delta \to 0$. Hence the Lemma follows. ■

2.8.2 Proof of Theorem 2.4.2

Proof The proof of Theorem 2.4.2 follows from Lemma 2.4.1. Now we need to verify the conditions (2.20)∼(2.25). Under the AR(1) model with $\beta \in [-1, 1]$, it is easy to verify all the conditions of Lemma 2.4.1 except (2.25). The condition (2.25) provides a upper bound δ for the leverage score of the data point in the streaming

AR(1) sereis. We start from the AR(1) model in sequential leveraging subsample $X_i = \beta X_{i-1} + \varepsilon_i,\ i = l, l+1, \cdots$. Without loss of generality, we assume that $\mathrm{E}_\beta \varepsilon_l^2 = 1$. By squaring and summing up to some time point j, we get

$$X_j^2 + (1-\beta^2)\sum_{i=l}^{j} X_{i-1}^2 - X_{l-1}^2 = \sum_{i=l}^{j} \varepsilon_i^2 + 2\beta \sum_{i=l}^{j} X_{i-1}\varepsilon_i. \tag{2.52}$$

Note that $|\beta| \le 1$ and let $0 < \lambda < \sigma^2/4$, then according to Lemma 2.8.2, we define

$$\Omega_{m,\lambda} = \left\{ \left| \frac{\sum_{i=l}^{j} \varepsilon_i^2}{j-l} - \sigma^2 \right| < \lambda,\ \left| \sum_{i=l}^{j} X_{i-1}\varepsilon_i \right| < \max\left[\lambda(j-l), (\sum_{i=l}^{j} X_{i-1}^2)^{2/3}\right] \right. \\ \left. \text{for all } j \ge m+l \right\}. \tag{2.53}$$

On $\Omega_{m,\lambda}$ if $j \ge m+l$ and $X_j^2 \le \lambda(j-l)$, then (2.52) suggests

$$\sum_{i=l}^{j} X_{i-1}^2 \ge (\sigma^2 - \lambda)(j-l) - \lambda(j-l) - 2\max(\lambda(j-l), (\sum_{i=l}^{j} X_{i-1}^2)^{2/3}) \tag{2.54}$$

and so for all m sufficiently large

$$2\sum_{i=l}^{j} X_{i-1}^2 \ge (\sigma^2 - 4\lambda)(j-l) \ge (\sigma^2\lambda^{-1} - 4)X_j^2. \tag{2.55}$$

Note that $|X_{j-1}| \ge |\beta X_{j-1}| \ge |X_j| - |\varepsilon_j|$, then

$$\min_{1\le t\le k-l+1} |X_{j-t}| \ge |X_j| - \sum_{t=0}^{k-l} |\varepsilon_{j-t}| \ge |X_j|\{1 - \sum_{t=0}^{k-l} |\varepsilon_{j-t}|/[\lambda(j-l)]^{1/2}\}; \tag{2.56}$$

if $X_j^2 \geq \lambda(j-l)$, then $j \geq k$ implies

$$\sum_{i=l}^{j} X_{i-1}^2 / X_j^2 \geq (k-l)\left[1 - \sum_{t=0}^{k-l} |\varepsilon_{j-t}| / (\lambda(j-l))^{1/2}\right]^2; \tag{2.57}$$

note that $\sum_{t=0}^{k-l} |\varepsilon_{j-t}|,\ j = k, k+1, \ldots$ are identically distributed with finite variance, thus we have

$$\lim_{k\to\infty} \mathrm{P}\left(\left|(k-l)\left[1 - \sum_{t=0}^{k-l} |\varepsilon_{j-t}| / [\lambda(j-l)]^{1/2}\right]^2 - (k-l)\right| < \epsilon\right) = 1, \tag{2.58}$$

where the probability bound on the right hand side of (2.57) is free of the parameter β. Now combine the results of (2.55) and (2.57) by choosing λ small so that $2/(\sigma^2\lambda - 4) < \zeta$ and k large so that $k > 1/\zeta + l$, we have

$$\sup_{\beta}\left[\mathrm{P}_\beta\left(X_j^2 \geq \zeta \sum_{i=l}^{j} X_{i-1}^2 \text{ for some } j \geq m\right)\right] < \sup_{\beta}\left[\mathrm{P}_\beta(\Omega_{m,\lambda}^c)\right] \to 0, \tag{2.59}$$

as $m \to \infty$ based on the result of Lemma 2.8.2. Thus the condition (2.25) is verified and then Theorem 2.4.2 holds. ■

2.8.3 Proof of Theorem 2.4.3

Proof

We have

$$\mathbf{V}_{\tau_c} = (\mathbf{\Gamma}_{\tau_c}^T \mathbf{\Gamma}_{\tau_c})^{1/2}(\widehat{\boldsymbol{\beta}}_{\tau_c} - \boldsymbol{\beta}) = (\mathbf{\Gamma}_{\tau_c}^T \mathbf{\Gamma}_{\tau_c})^{-1/2} \mathbf{\Gamma}_{\tau_c}^T \boldsymbol{\varepsilon}_{\tau_c}, \tag{2.60}$$

where $\boldsymbol{\varepsilon}_{\tau_c} = (\varepsilon_l, \ldots, \varepsilon_{\tau_c})^T$. Applying the transformation of coefficients $\boldsymbol{\beta}$ using $L(\boldsymbol{\beta})$

in (2.29) and define

$$\mathbf{Y}_c = \frac{1}{\sqrt{c}} L^{-1/2}(\boldsymbol{\beta}) \Gamma_{\tau_c}^T \boldsymbol{\varepsilon}_{\tau_c}, \tag{2.61}$$

and

$$\mathcal{G}_{\tau_c} = L^{-1/2}(\boldsymbol{\beta}) \boldsymbol{\Gamma}_{\tau_c}^T \boldsymbol{\Gamma}_{\tau_c} L^{-1/2}(\boldsymbol{\beta}). \tag{2.62}$$

Then

$$\mathbf{V}_{\tau_c} = \sqrt{c} (\boldsymbol{\Gamma}_{\tau_c}^T \boldsymbol{\Gamma}_{\tau_c})^{-1/2} L^{1/2}(\boldsymbol{\beta}) \mathbf{Y}_c. \tag{2.63}$$

The transformation has the property that

$$\begin{aligned} \left|\left| \frac{1}{\sqrt{c}} L^{-1/2}(\boldsymbol{\beta}) (\boldsymbol{\Gamma}_{\tau_c}^T \boldsymbol{\Gamma}_{\tau_c})^{1/2} - \mathbf{I}_p \right|\right|^2 &= \left|\left| \frac{1}{\sqrt{c}} \mathcal{G}_{\tau_c}^{1/2} - \mathbf{I}_p \right|\right|^2 \leq ||c^{-1} \mathcal{G}_{\tau_c} - \mathbf{I}_p||^2 \\ &\leq \mathrm{tr} L^{-1}(\boldsymbol{\beta}) ||c^{-1} \boldsymbol{\Gamma}_{\tau_c}^T \boldsymbol{\Gamma}_{\tau_c} - L(\boldsymbol{\beta})||^2, \end{aligned} \tag{2.64}$$

where $\mathbf{I_p}$ is the $p \times p$ identity matrix. Use Lemma 2.8.3, for any $\delta > 0$, we get

$$\lim_{c \to \infty} \sup_{\boldsymbol{\beta} \in K} \mathrm{P}_{\boldsymbol{\beta}} \left(||\sqrt{c} (\boldsymbol{\Gamma}_{\tau_c}^T \boldsymbol{\Gamma}_{\tau_c})^{-1/2} L^{1/2}(\boldsymbol{\beta}) - \mathbf{I}_p|| > \delta \right) = 0, \tag{2.65}$$

uniformly for $\boldsymbol{\beta} \in K$ for any compact set $K \subset \tilde{\Lambda}_p$.

We shall find the limiting distribution of $\mathbf{V}_{\tau_c}$ by showing that an arbitrary linear combination of $\mathbf{Y}_c$, say $\mathbf{v}^T \mathbf{Y}_c$ has a limiting normal distribution for any $\mathbf{v} \in \mathbb{R}^p$ with $||\mathbf{v}|| = 1$. Let

$$g_i = \mathbf{v}^T L^{-1/2}(\boldsymbol{\beta}) \mathbf{z}_i,$$

now we can define an auxiliary sequential sampling rule similar to (2.19)

$$\tilde{\tau}_c = \inf\{t \geq l : \sum_{i=l}^{t} g_i^2 \geq c\}. \tag{2.66}$$

Then

$$\mathbf{v}^T \mathbf{Y}_c = \frac{1}{\sqrt{c}} \mathbf{v}^T L^{-1/2}(\boldsymbol{\beta}) \Gamma_{\tau_c}^T \boldsymbol{\varepsilon} = \frac{1}{\sqrt{c}} \sum_{i=l}^{\tau_c} g_i \varepsilon_i = \frac{1}{\sqrt{c}} \sum_{i=l}^{\tilde{\tau}_c} g_i \varepsilon_i + \eta_c + \Delta(c), \tag{2.67}$$

where $\Delta(c) = \frac{1}{\sqrt{c}} \left(g_l \varepsilon_l I_{\{\tau_c = l\}} - g_l \varepsilon_l I_{\{\tilde{\tau}_c = l\}} + g_{\tau_c} \varepsilon_{\tau_c} + g_{\tilde{\tau}_c} \varepsilon_{\tilde{\tau}_c} \right)$ with $\eta_c = \frac{1}{\sqrt{c}} \sum_{i=l}^{\tau_c - 1} g_i \varepsilon_i - \frac{1}{\sqrt{c}} \sum_{i=l}^{\tilde{\tau}_c - 1} g_i \varepsilon_i I_{\{\tilde{\tau}_c > l\}}$. We need to show the uniform asymptotic normality

$$\lim_{c \to \infty} \sup_{\boldsymbol{\beta} \in K} \sup_{x \in \mathbb{R}} |\mathrm{P}_{\boldsymbol{\beta}}(\frac{1}{\sqrt{c}} \sum_{i=l}^{\tilde{\tau}_c} g_i \varepsilon_i \leq x) - \Phi(x)| = 0. \tag{2.68}$$

and for any $\delta > 0$ and any $\delta' > 0$

$$\lim_{c \to \infty} \sup_{\boldsymbol{\beta} \in K} \mathrm{P}_{\boldsymbol{\beta}}(|\eta_c| > \delta) = 0, \tag{2.69}$$

$$\lim_{c \to \infty} \sup_{\boldsymbol{\beta} \in K} \mathrm{P}_{\boldsymbol{\beta}}(|\Delta(c)| > \delta') = 0. \tag{2.70}$$

The proof of (2.68) follows Lemma 2.4.1 through verifying conditions (2.20) to (2.25). All conditions are trivially verified except (2.25), that is, for each given $\zeta > 0$ and arbitrarily large number $m > l$

$$\lim_{m \to \infty} \sup_{\boldsymbol{\beta} \in K} \mathrm{P}_{\boldsymbol{\beta}} \left(g_{j+1}^2 \geq \zeta \sum_{i=l}^{j} g_i^2 \text{ for some } j \geq m \right) = 0. \tag{2.71}$$

Since

$$\sum_{i=l}^{j} g_i^2 = \sum_{i=l}^{j} \mathbf{v}^T L^{-1/2}(\boldsymbol{\beta}) \mathbf{z}_i \mathbf{z}_i^T L^{-1/2}(\boldsymbol{\beta}) \mathbf{v} \tag{2.72}$$

$$= \left[\mathbf{v}^T L^{-1/2}(\boldsymbol{\beta}) \left(\frac{1}{\sum_{i=l}^{j} X_{i-1}^2} \boldsymbol{\Gamma}_{\tau_c}^T \boldsymbol{\Gamma}_{\tau_c} - L(\boldsymbol{\beta}) \right) L^{-1/2}(\boldsymbol{\beta}) \mathbf{v} + 1 \right] \sum_{i=l}^{j} X_{i-1}^2, \tag{2.73}$$

we get

$$\begin{aligned}
&\mathrm{P}_{\boldsymbol{\beta}} \left(g_{j+1}^2 \geq \zeta \sum_{i=l}^{j} g_i^2 \text{ for some } j \geq m \right) \leq \mathrm{P}_{\boldsymbol{\beta}} \left(||\mathbf{z}_{j+1}||^2 \geq \frac{\zeta}{\iota} \sum_{i=l}^{j} g_i^2 \text{ for some } j \geq m \right) \\
&= \mathrm{P}_{\boldsymbol{\beta}} \Bigg(||\mathbf{z}_{j+1}||^2 \geq \frac{\zeta}{\iota} \left[1 + \mathbf{v}^T L^{-1/2}(\boldsymbol{\beta}) \left(\frac{\boldsymbol{\Gamma}_j^T \boldsymbol{\Gamma}_j}{\sum_{i=l}^{j} X_{i-1}^2} - L(\boldsymbol{\beta}) \right) L^{-1/2}(\boldsymbol{\beta}) \mathbf{v} \right] \sum_{i=l}^{j} X_{i-1}^2 \\
&\qquad \text{for some } j \geq m \Bigg) \\
&\leq \mathrm{P}_{\boldsymbol{\beta}} \Bigg(||\mathbf{z}_{j+1}||^2 \geq \frac{\zeta}{\iota} \left[1 - ||\mathbf{v}^T L^{-1/2}(\boldsymbol{\beta})||^2 \left|\left| \frac{\boldsymbol{\Gamma}_j^T \boldsymbol{\Gamma}_j}{\sum_{i=l}^{j} X_{i-1}^2} - L(\boldsymbol{\beta}) \right|\right| \right] \sum_{i=l}^{j} X_{i-1}^2 \\
&\qquad \text{for some } j \geq m \Bigg) \\
&\leq \mathrm{P}_{\boldsymbol{\beta}} \left(||\mathbf{z}_{j+1}||^2 \geq \frac{\zeta}{\iota} \left[1 - \iota \left|\left| \frac{\boldsymbol{\Gamma}_j^T \boldsymbol{\Gamma}_j}{\sum_{i=l}^{j} X_{i-1}^2} - L(\boldsymbol{\beta}) \right|\right| \right] \sum_{i=l}^{j} X_{i-1}^2 \text{ for some } j \geq m \right) \\
&\leq \mathrm{P}_{\boldsymbol{\beta}} \left(\left|\left| \frac{\boldsymbol{\Gamma}_j^T \boldsymbol{\Gamma}_j}{\sum_{i=l}^{j} X_{i-1}^2} - L(\boldsymbol{\beta}) \right|\right| \geq \frac{1}{2\iota} \text{ for some } j \geq m \right) \\
&\qquad + \mathrm{P}_{\boldsymbol{\beta}} \left(||\mathbf{z}_{j+1}||^2 \geq \frac{\zeta}{2\iota} \sum_{i=l}^{j} X_{i-1}^2 \text{ for some } j \geq m \right),
\end{aligned}$$

where $\iota = \sup_{\boldsymbol{\beta} \in K} ||\mathbf{v}^T L^{-1/2}(\boldsymbol{\beta})||^2$.

By Lemma 4.1 of Galtchouk and Konev (2011), we have

$$\lim_{m\to\infty}\sup_{\boldsymbol{\beta}\in K}\mathrm{P}_{\boldsymbol{\beta}}\left(||\mathbf{z}_{j+1}||^2 \geq \frac{\zeta}{2\iota}\sum_{i=l}^{j} X_{i-1}^2 \text{ for some } j \geq m\right) = 0; \tag{2.74}$$

and by Lemma 4.3 of Galtchouk and Konev (2011), we have

$$\lim_{m\to\infty}\sup_{\boldsymbol{\beta}\in K}\mathrm{P}_{\boldsymbol{\beta}}\left(\left|\left|\frac{\boldsymbol{\Gamma}_j^T\boldsymbol{\Gamma}_j}{\sum_{i=l}^{j} X_{i-1}^2} - L(\boldsymbol{\beta})\right|\right| \geq \frac{1}{2\iota} \text{ for some } j \geq m\right) = 0; \tag{2.75}$$

thus the condition (2.71) holds.

With the result of (2.71) , we now show that based on stopping rule (2.66), for any compact set $K \subset \tilde{\Lambda}_p$ and $\forall\zeta > 0$

$$\lim_{m\to\infty}\sup_{\boldsymbol{\beta}\in K}\mathrm{P}_{\boldsymbol{\beta}}\left(g_{\tilde{\tau}_c}^2 \geq \zeta\sum_{i=l}^{\tilde{\tau}_c-1} g_i^2\right) = 0. \tag{2.76}$$

For arbitrarily large number m

$$\sup_{\boldsymbol{\beta}\in K}\mathrm{P}_{\boldsymbol{\beta}}\left(g_{\tilde{\tau}_c}^2 \geq \zeta\sum_{i=l}^{\tilde{\tau}_c-1} g_i^2\right) \leq \mathrm{P}_{\boldsymbol{\beta}}\left(g_t^2 \geq \zeta\sum_{i=l}^{t-1} g_i^2 \text{ for some } t \geq m\right) + \mathrm{P}_{\boldsymbol{\beta}}(\tilde{\tau}_c < m) \tag{2.77}$$

$$\leq \mathrm{P}_{\boldsymbol{\beta}}\left(g_t^2 \geq \zeta\sum_{i=l}^{t-1} g_i^2 \text{ for some } t \geq m\right) + I_{\{(m-l)k>c\}} + \sum_{j=l}^{m}\sup_{\boldsymbol{\beta}\in K}\mathrm{P}_{\boldsymbol{\beta}}\left(||\mathbf{z}_j||^2 \geq k\right). \tag{2.78}$$

The first term of the right hand side of (2.78) follows from (2.71), and the second and the third terms hold for letting $k \to \infty$ and $c \to \infty$ (See details in proof of

Lemma 2.8.3).

The proof of (2.69) starts from checking the variance of η_c, $\mathrm{E}_{\boldsymbol{\beta}}\eta_c^2$. We now show that η_c^2 is bounded from above uniformly for $\boldsymbol{\beta} \in K$,

$$\begin{aligned}
\eta_c^2 &= \frac{1}{c}\left|\sum_{i=l}^{\tau_c-1} g_i^2 - \sum_{i=l}^{\tilde{\tau}_c-1} g_i^2\right| = \frac{1}{c}\left|\sum_{i=l}^{\tau_c-1} \mathbf{v}^T L^{-1/2}(\boldsymbol{\beta})\mathbf{z}_i\mathbf{z}_i^T L^{-1/2}(\boldsymbol{\beta})\mathbf{v} - \sum_{i=l}^{\tilde{\tau}_c-1} g_i^2\right| \\
&= \frac{1}{c}\mathbf{v}^T L^{-1/2}(\boldsymbol{\beta})\left(\frac{\boldsymbol{\Gamma}_{\tau_c}^T\boldsymbol{\Gamma}_{\tau_c}}{\sum_{i=l}^{\tau_c-1} X_{i-1}^2} - L(\boldsymbol{\beta})\right) L^{-1/2}(\boldsymbol{\beta})\mathbf{v}\sum_{i=l}^{\tau_c-1} X_{i-1}^2 + \frac{1}{c}\sum_{i=l}^{\tau_c-1} X_{i-1}^2 - \frac{1}{c}\sum_{i=l}^{\tilde{\tau}_c-1} g_i^2 \\
&\leq \iota\left|\left|\frac{\boldsymbol{\Gamma}_{\tau_c}^T\boldsymbol{\Gamma}_{\tau_c}}{\sum_{i=l}^{\tau_c-1} X_{i-1}^2} - L(\boldsymbol{\beta})\right|\right| + \frac{X_{\tau_c-1}^2}{\sum_{i=l}^{\tau_c-1} X_{i-1}^2} + \frac{g_{\tilde{\tau}_c}^2}{\sum_{i=l}^{\tilde{\tau}_c-1} g_i^2},
\end{aligned}$$

where $\iota = \sup_{\boldsymbol{\beta}\in K} ||\mathbf{v}^T L^{-1/2}(\boldsymbol{\beta})||^2$. By the result of (2.76) and Lemmas 2.8.3, 2.8.4, we get the result of (2.69). Similarly, one can prove (2.70). With the results of (2.68), (2.69) and (2.70), the Theorem 2.4.3 holds.■

2.8.4 More Lemmas

Lemma 2.8.3. *Let $\{\hat{\boldsymbol{\beta}}_{\tau_c}, \tau_c \geq p+1\}$ be the least squares estimate of $\boldsymbol{\beta}$ based on design matrix (2.18), $L(\boldsymbol{\beta})$ defined in (2.30) and τ_c defined in (2.17). Let X_t, ε_t be random variables adapted to the increasing sequence of σ-algebras $\mathscr{F}_t$, $t = \ldots, l-1, l, l+1, \ldots$. If $\varepsilon_l, \varepsilon_{l+1}, \ldots,$ are i.i.d. with mean 0 and variance σ^2, and the sequence $\{\varepsilon_i : i \geq l\}$ is independent of the starting state $\mathbf{z}_l = (X_{l-1}, \ldots, X_{l-p})^T$ defined in Algorithm 2, then given $\mathscr{F}_{l-1}$, uniformly in $\boldsymbol{\beta} \in K$ for any compact set $K \subset \check{\Lambda}_p$ defined in Section 2.4.2, for any $\delta > 0$, we have*

$$\lim_{c\to\infty}\sup_{\boldsymbol{\beta}\in K} \mathrm{P}_{\boldsymbol{\beta}}\left(\left|\left|\frac{\boldsymbol{\Gamma}_{\tau_c}^T\boldsymbol{\Gamma}_{\tau_c}}{c} - L(\boldsymbol{\beta})\right|\right| > \delta\right) = 0. \tag{2.79}$$

Proof By rearranging the terms of interest, we build connection to Lemma 4.3 of Galtchouk and Konev (2011), since

$$\left|\left|\frac{\mathbf{\Gamma}_{\tau_c}^T\mathbf{\Gamma}_{\tau_c}}{c}-L(\boldsymbol{\beta})\right|\right| \leq \left|\left|\frac{\mathbf{\Gamma}_{\tau_c}^T\mathbf{\Gamma}_{\tau_c}}{\sum_{i=l}^{\tau_c}X_{i-1}^2}-L(\boldsymbol{\beta})\right|\right| + \frac{||\mathbf{\Gamma}_{\tau_c}^T\mathbf{\Gamma}_{\tau_c}||}{c} - \frac{||\mathbf{\Gamma}_{\tau_c}^T\mathbf{\Gamma}_{\tau_c}||}{\sum_{i=l}^{\tau_c}X_{i-1}^2}$$

We have

$$\begin{aligned}\mathrm{P}_{\boldsymbol{\beta}}\left(\left|\left|\frac{\mathbf{\Gamma}_{\tau_c}^T\mathbf{\Gamma}_{\tau_c}}{c}-L(\boldsymbol{\beta})\right|\right|>\delta\right) \leq\ & \mathrm{P}_{\boldsymbol{\beta}}\left(\left|\left|\frac{\mathbf{\Gamma}_{\tau_c}^T\mathbf{\Gamma}_{\tau_c}}{\sum_{i=l}^{\tau_c}X_{i-1}^2}-L(\boldsymbol{\beta})\right|\right|>\frac{\delta}{2}\right) \\ & + \mathrm{P}_{\boldsymbol{\beta}}\left(\frac{1}{c}\frac{||\mathbf{\Gamma}_{\tau_c}^T\mathbf{\Gamma}_{\tau_c}||}{\sum_{i=l}^{\tau_c}X_{i-1}^2}(\sum_{i=l}^{\tau_c}X_{i-1}^2-c)>\frac{\delta}{2}\right).\end{aligned} \tag{2.80}$$

The first term of the right hand side of (2.80) can be decomposed as

$$\begin{aligned}&\mathrm{P}_{\boldsymbol{\beta}}\left(\left|\left|\frac{\mathbf{\Gamma}_{\tau_c}^T\mathbf{\Gamma}_{\tau_c}}{\sum_{i=l}^{\tau_c}X_{i-1}^2}-L(\boldsymbol{\beta})\right|\right|>\frac{\delta}{2}\right) \\ &\leq \mathrm{P}_{\boldsymbol{\beta}}(\tau_c<m)+\mathrm{P}_{\boldsymbol{\beta}}\left(\left|\left|\frac{\mathbf{\Gamma}_{t}^T\mathbf{\Gamma}_{t}}{\sum_{i=l}^{t}X_{i-1}^2}-L(\boldsymbol{\beta})\right|\right|>\frac{\delta}{2}\text{ for some } t\geq m\right),\end{aligned}$$

similarly, the second term of the right hand side of (2.80) can be written as

$$\begin{aligned}&\mathrm{P}_{\boldsymbol{\beta}}\left(\frac{1}{c}\frac{||\mathbf{\Gamma}_{\tau_c}^T\mathbf{\Gamma}_{\tau_c}||}{\sum_{i=l}^{\tau_c}X_{i-1}^2}(\sum_{i=l}^{\tau_c}X_{i-1}^2-c)>\frac{\delta}{2}\right) \\ &\leq \mathrm{P}_{\boldsymbol{\beta}}(\tau_c<m)+\mathrm{P}_{\boldsymbol{\beta}}\left(\frac{||\mathbf{\Gamma}_{t}^T\mathbf{\Gamma}_{t}||}{\sum_{i=l}^{t}X_{i-1}^2}\frac{X_{i-1}^2}{\sum_{i=l}^{t}X_{i-1}^2}>\frac{\delta}{2}\text{ for some } t\geq m\right).\end{aligned}$$

By the definition of τ_c (2.17), we have

$$\begin{aligned}
\mathrm{P}_{\boldsymbol{\beta}}(\tau_c < m) &= \mathrm{P}_{\boldsymbol{\beta}}(\sum_{i=l}^{m} ||\mathbf{z}_i||^2 > c) \\
&= \mathrm{P}_{\boldsymbol{\beta}}(\sum_{i=l}^{m} ||\mathbf{z}_i||^2 > c, \max_{l\le j\le m} ||\mathbf{z}_j||^2 < k) + \mathrm{P}_{\boldsymbol{\beta}}(\sum_{i=l}^{m} ||\mathbf{z}_i||^2 > c, \max_{l\le j\le m} ||\mathbf{z}_j||^2 \ge k) \\
&\le \mathrm{P}_{\boldsymbol{\beta}}[(m-l)k > c] + \sum_{i=l}^{m} \mathrm{P}_{\boldsymbol{\beta}}(||\mathbf{z}_i||^2 \ge k).
\end{aligned}$$

By decomposition $||\boldsymbol{\Gamma}_t^T\boldsymbol{\Gamma}_t||/\sum_{i=l}^t X_{i-1}^2 \le ||\boldsymbol{\Gamma}_t^T\boldsymbol{\Gamma}_t/\sum_{i=l}^t X_{i-1}^2 - L(\boldsymbol{\beta})|| + ||L(\boldsymbol{\beta})||$, we have

$$\begin{aligned}
&\mathrm{P}_{\boldsymbol{\beta}}\left(\frac{||\boldsymbol{\Gamma}_t^T\boldsymbol{\Gamma}_t||}{\sum_{i=l}^t X_{i-1}^2}\frac{X_{i-1}^2}{\sum_{i=l}^t X_{i-1}^2} > \frac{\delta}{2} \text{ for some } t \ge m\right) \\
&\quad\le \mathrm{P}_{\boldsymbol{\beta}}\left(\Big|\Big|\frac{\boldsymbol{\Gamma}_t^T\boldsymbol{\Gamma}_t}{\sum_{i=l}^t X_{i-1}^2} - L(\boldsymbol{\beta})\Big|\Big| > \left(\frac{\delta}{4}\right)^{1/2} \text{ for some } t \ge m\right) \\
&\quad+ \mathrm{P}_{\boldsymbol{\beta}}\left(\frac{X_{i-1}^2}{\sum_{i=l}^t X_{i-1}^2}\sup_{\boldsymbol{\beta}\in K}||L(\boldsymbol{\beta})|| > \left(\frac{\delta}{4}\right)^{1/2} \text{ for some } t \ge m\right) \\
&\quad+ \mathrm{P}_{\boldsymbol{\beta}}\left(\frac{X_{i-1}^2}{\sum_{i=l}^t X_{i-1}^2}\sup_{\boldsymbol{\beta}\in K}||L(\boldsymbol{\beta})|| > \frac{\delta}{4} \text{ for some } t \ge m\right).
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathrm{P}_{\boldsymbol{\beta}}\left(\Big|\Big|\frac{\boldsymbol{\Gamma}_{\tau_c}^T\boldsymbol{\Gamma}_{\tau_c}}{c} - L(\boldsymbol{\beta})\Big|\Big| > \delta\right) &\le 2\mathrm{P}_{\boldsymbol{\beta}}[(m-l)k > c] + 2\sum_{i=l}^{m}\sup_{\boldsymbol{\beta}\in K}\mathrm{P}_{\boldsymbol{\beta}}(||\mathbf{z}_i||^2 \ge k) \\
&+ \mathrm{P}_{\boldsymbol{\beta}}\left(\Big|\Big|\frac{\boldsymbol{\Gamma}_t^T\boldsymbol{\Gamma}_t}{\sum_{i=l}^t X_{i-1}^2} - L(\boldsymbol{\beta})\Big|\Big| > \delta' \text{ for some } t \ge m\right) \\
&+ 2\mathrm{P}_{\boldsymbol{\beta}}\left(\frac{X_{i-1}^2}{\sum_{i=l}^t X_{i-1}^2}\sup_{\boldsymbol{\beta}\in K}||L(\boldsymbol{\beta})|| > \delta' \text{ for some } t \ge m\right),
\end{aligned}$$

where $\delta' \leq \frac{1}{2}\min(\delta, \delta^{1/2})$.

For given l, according to our construction $\lim_{c\to\infty}(m-l)k/c = 0$ as $k \to \infty$, $m \to \infty$, $c \to \infty$. Thus, letting $k \to \infty$, $m \to \infty$, and $c \to \infty$, combined with Lemma 4.3 of Galtchouk and Konev (2011), we have (2.79). ■

Lemma 2.8.4. *Let $\{\hat{\boldsymbol{\beta}}_{\tau_c}, \tau_c \geq p+1\}$ be the least squares estimate of $\boldsymbol{\beta}$ based on design matrix (2.18), $L(\boldsymbol{\beta})$ defined in (2.30) and τ_c defined in (2.17). Let X_t, ε_t be random variables adapted to the increasing sequence of σ-algebras $\mathscr{F}_t$, $t = \ldots, l-1, l, l+1, \ldots$. If $\varepsilon_l, \varepsilon_{l+1}, \ldots,$ are i.i.d. with mean 0 and variance σ^2, and the sequence $\{\varepsilon_i : i \geq l\}$ is independent of the starting state $\mathbf{z}_l = (X_{l-1}, \ldots, X_{l-p})^T$ defined in Algorithm 2, then given $\mathscr{F}_{l-1}$, uniformly in $\boldsymbol{\beta} \in K$ for any compact set $K \subset \check{\Lambda}_p$ defined in Section 2.4.2, for any $\delta > 0$, we have*

$$\lim_{c\to\infty} \sup_{\boldsymbol{\beta}\in K} \mathrm{P}_{\boldsymbol{\beta}}\left(\frac{X^2_{\tau_c-1}}{\sum_{i=l}^{\tau_c-1} X^2_{i-1}} > \delta\right) = 0. \tag{2.81}$$

Proof

$$\sup_{\boldsymbol{\beta}\in K} \mathrm{P}_{\boldsymbol{\beta}}\left(\frac{X^2_{\tau_c-1}}{\sum_{i=l}^{\tau_c-1} X^2_{i-1}} > \delta\right) \leq \mathrm{P}_{\boldsymbol{\beta}}[(m-l)k > c] + \sum_{j=l}^{m} \sup_{\boldsymbol{\beta}\in K} \mathrm{P}_{\boldsymbol{\beta}}\left(||\mathbf{z}_j||^2 \geq k\right)$$
$$+ \sup_{\boldsymbol{\beta}\in K} \mathrm{P}_{\boldsymbol{\beta}}\left(X_j^2 \geq \delta \sum_{i=l}^{j} X^2_{i-1} \text{ for some } j \geq m\right).$$

By the result of Theorem 2.4.2 and letting $k \to \infty$, $m \to \infty$ and $c \to \infty$, we have (2.81). Thus complete the proof. ■

Chapter 3

Online Decentralized Leverage Score Sampling for Streaming Multidimensional Time Series[1]

We propose a *leverage score sampling* (LSS) method for efficient online inference of the streaming vector autoregressive (VAR) model. We define the leverage score for the streaming VAR model so that the LSS method selects informative data points in real-time with statistical guarantees of parameter estimation efficiency. Moreover, our LSS method can be directly deployed in an asynchronous decentralized environment, e.g., a sensor network without a fusion center, and produce asynchronous consensus online parameter estimation over time. By exploiting the temporal depen-

[1] Xie, R., Wang, Z., Bai, R., Zhong, W., and Ma, P.(2019) Online decentralized leverage score sampling for streaming multidimensional time series, The 22nd International Conference on Artificial Intelligence and Statistics (AISTATS 2019), accepted.

dence structure of the VAR model, the LSS method selects samples independently on each dimension and thus is able to update the estimation asynchronously. We illustrate the effectiveness of the LSS method in synthetic, gas sensor and seismic datasets.

3.1 Introduction

Understanding the dependence structure of streaming multidimensional time series in real-time is a "space-time" challenge due to (1) the temporal dependency and infinite sample size of data streams in time, and (2) cross-correlation among multidimensional streams and information transition in the data acquisition network on space. The multidimensional streaming data are commonly collected from a network system with each node corresponding to one marginal dimension of the streams. The multidimensional streams contain complex temporal and cross-sectional dependency, usually along with a huge volume of data. Accurately and efficiently estimating the dependence structure is crucial, especially for real-time inference tasks, but the estimating process is time-consuming. Sampling is a natural and efficient way to reduce the data size and speed up the computation. Meanwhile, when the multidimensional streams are collected across distributed nodes asynchronously, it is not practical to transfer all data to one computing node and process them with an increasing data volume. One reasonable approach to retrieve dependence information is to perform asynchronous consensus estimation on each node in the decentralized computing framework Wu et al. (2018); Xiao and Wang (2008). Sampling can relief the storage

pressure and minimize the communication cost in such decentralized network.

The vector autoregressive (VAR) model, one of the most popular and fundamental time series models, provides a mechanism for capturing complex temporal dependency and cross-correlation among the multidimensional time series. Inferring these dependencies requires both efficient methodology and intensive computational efforts. Precisely understanding theses dependencies facilitates the interpretation of the model and improves prediction accuracy.

In this work, we introduce a *leverage score sampling* (LSS) method that can efficiently estimate the dependence structure from asynchronous multidimensional streaming time series. By exploiting the VAR model, we parameterize the temporal dependence (auto/cross-correlation) structure and propose the streaming statistical leverage scores for streaming sampling. We also seek to directly deploy this method to an asynchronous decentralized network, which has limited energy, memory and processing resource. In these cases, finding the informative data points is highly desired for accelerating the estimation process and boosting the transmission of the streaming data in the decentralized network system.

Challenges: In this paper, we focus on designing a sampling strategy that can improve the parameter estimation accuracy and maintain the computation efficiency. We address a few specific challenges in sampling streaming multidimensional time series. First, how do we find a subset of samples that efficiently capture the temporal structure under a multidimensional setting? The proposed sampling method aims to find influential data points, which are highly efficient for estimating the parameter matrix of the VAR model, in real-time to reduce evaluation times without losing too

much accuracy. Second, how do we adapt the importance sampling method to the streaming and decentralized environment? We utilize the VAR model to decompose the dependence structure and distribute it to each node so that the sampling method can be applied on each node independently.

Prior Work: Sampling is an important data reduction approach for *reducing the computational cost and memory usage*, and it is widely used in matrix approximation or sketching Drineas et al. (2008, 2012b); Boutsidis et al. (2014); Zhang et al. (2017), kernel approximation Musco and Musco (2017); Achlioptas et al. (2002), graph sparsification Spielman and Srivastava (2011); Kapralov et al. (2017), linear regression Ma et al. (2014); Derezinski and Warmuth (2017); Raskutti and Mahoney (2015), and etc. Especially, sampling method based on leverage score is one of the most popular techniques Papailiopoulos et al. (2014); Cohen et al. (2017); Mahoney and Drineas (2009b). Random sampling with probability proportional to exact or approximated leverage scores can yield high accuracy on model parameter estimation for linear regression Ma et al. (2015b); Raskutti and Mahoney (2015), logistic regression Wang et al. (2017) and kernel ridge regression Alaoui and Mahoney (2015).

On the other hand, sampling as the subset selectionwhich *optimizes a specified objective function* leads to numerous applications in image, video, speech summarization Elhamifar and Kaluza (2017a,b); Gong et al. (2014); Simon et al. (2007); Lin and Bilmes (2012); Kulesza et al. (2012), and bioinformatics Wu and Wang (2016); Jörnsten and Yu (2003). Most of the existing methods treat the samples independently and ignore the dependence information among the samples, except the most recent work of Elhamifar and Kaluza (2017b) that selects the sequential data based

on Markov models.

Meanwhile, literatures on sampling for *streaming data* have focused on column sampling Cohen et al. (2017), spectral sparsification or subgraph sampling for graph streams Kapralov et al. (2017); Chitnis et al. (2016), data management Cormode et al. (2012); Efraimidis (2015), and clustering Song et al. (2018). To the best of our knowledge, the study on sampling with an objective of recovering the dependence information of streaming data is still lacking.

Paper Contributions: In this paper, we develop a novel sampling method for estimating temporal dependence structure of multidimensional streaming multidimensional time series. Our leverage score sampling (LSS) method is based on the statistical leverage score of vector autoregressive model for online selecting representative data points, which are later used to estimate the VAR model parameter matrix.

1. The LSS differs substantially from other leverage-based sampling methods. The LSS focuses on selecting informative data points that contribute to the estimation efficiency of the VAR model parameter matrix, which is a model-based surrogate for temporal dependence structure of the multidimensional time series streams.

2. We provide a theoretical guarantee that the LSS method yields a better estimation efficiency for the VAR model parameter matrix than naive sampling methods.

3. Not only is the LSS method fast and accurate for estimating temporal dependence structure, but it can also be applied in an asynchronous decentralized environment where traditional leverage-based sampling methods cannot.

As an illustration, we present a single-pass streaming sampling algorithm on the

asynchronous decentralized framework for consensus optimization. We demonstrate the practical effectiveness with such asynchronous decentralized environment, both in parameter estimation on K-dimensional VAR(p) synthetic data streams, as well as in real large-scale sensor data prediction tasks.

3.2 Background

3.2.1 Notation

A curly capital letter $\mathcal{A}$ is used for set and collection of sets. The upper-case letters $\mathbf{A}$ or A are used for matrices and operators. A lower-case bold letter $\mathbf{a}$ is used for vector and a lower-case letter a is used for scalar. Specifically, we reserve $\mathrm{E}(\cdot)$ to denote the expectation operator. The integers are denoted by $\boldsymbol{\beta}Z$, and real numbers are denoted by $\boldsymbol{\beta}R$. We denote the identity matrix of dimension n by $\mathbf{I}_n \in \mathbb{R}^{n\times n}$. We use $1_{\{\cdot\}}$ to denote the indicator function. We use $\boldsymbol{\Sigma}_1 \prec \boldsymbol{\Sigma}_2$, for two non-negative-definite matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, to indicate that $\boldsymbol{\Sigma}_2 - \boldsymbol{\Sigma}_1$ is positive definite. We denote the transpose of a matrix $\mathbf{A}$ as $\mathbf{A}'$, the determinant of a matrix $\mathbf{A}$ as $\det(\mathbf{A})$ and vectorization operator as $\mathrm{vec}(\cdot)$. The Kronecker product is denoted by $\otimes$. Finally, $\xrightarrow{d}$ denotes the convergence in distribution, $\triangleq$ means *defined to be equal to*, $||\cdot||_2$ denotes the vector ℓ_2-norm, and $||\cdot||_F$ denotes the matrix Frobenius norm.

3.2.2 Vector Autoregressive Model

Time series data, one of the most representative classes of dependent data, which contain temporal dependence structure among samples, are often modeled by the

vector autoregressive (VAR) models Box et al. (2015); Harvey (1990); West et al. (1985). VAR model represents a family of time series models that offers a broad framework for capturing complex temporal and longitudinal interrelationship among the multidimensional time series data. A centered time series $\mathbf{y}_t \in \mathbb{R}^K$ follows a K-dimensional vector autoregressive model of order p, i.e., VAR(p), if

$$\mathbf{y}_t = \sum_{i=1}^{p} \mathbf{\Phi}_i \mathbf{y}_{t-i} + \mathbf{e}_t, \quad t \in \boldsymbol{\beta} Z, \tag{3.1}$$

where $\mathbf{\Phi}_i$'s are $K \times K$ matrices, and $\mathbf{e}_t$ is a sequence of independent and identically distributed (i.i.d.) random vectors with mean zero and and finite non-singular covariance matrix $\mathrm{E}[\mathbf{e}_t \mathbf{e}_t'] = \mathbf{\Psi}$. Let $\mathbf{\Phi}(z) = \mathbf{I}_K - \sum_{j=1}^{p} \mathbf{\Phi}_j z^j$ be the the associated characteristic matrix polynomial, where $\mathbf{I}_p$ is the $p \times p$ identity matrix. We assume that $\det(\mathbf{\Phi}(z)) \neq 0$ on the complex unit disk $\{z \in \boldsymbol{\beta} C : \ |z| \leq 1\}$. In this case, there is a unique stationary solution $\mathbf{y}_t$ of (3.1), which is expressed as a causal linear filter of $(\mathbf{e}_t)$ (Theorem 11.3.1 of Brockwell and Davis (2013)).

We rewrite the VAR(p) model as

$$\mathbf{y}_t' = \mathbf{x}_t' \mathbf{B} + \mathbf{e}_t', \quad t \in \boldsymbol{\beta} Z, \tag{3.2}$$

where $\mathbf{B} = [\Phi_1', \Phi_2', \cdots, \Phi_p']'$ is the $Kp \times K$ model parameter matrix, and $\mathbf{x}_t = (\mathbf{y}_{t-1}', \mathbf{y}_{t-2}', \cdots, \mathbf{y}_{t-p}')'$ is a column vector of length $m = Kp$. We assume that the covariance matrix $\Gamma = \mathrm{E}[\mathbf{x}_t \mathbf{x}_t']$ is non-singular. The model parameter estimation can be done through Ordinary Least-Squares (OLS) estimate Tsay (2013), which is $\hat{\mathbf{B}}_{OLS} = \left(\sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t' \right)^{-1} \sum_{t=1}^{T} \mathbf{x}_t \mathbf{y}_t'$. The computational cost of estimating the model

parameter matrix $\mathbf{B}$ is $O(TK^2p^2)$.

3.3 Leverage Score Sampling for Time Series Data

The *leverage score sampling* (LSS) method for streaming time series data utilizes the structure information of the underlying dynamic model to efficiently select informative samples. The information contained in multidimensional time series are projected onto a one-dimensional space through the LSS procedure, which results in an easy-to-implement and efficient sampling criterion.

We use the VAR(p) model to characterize the temporal dependence structure of K-dimensional time series, which keeps the interoperability, compatibility and avoids the overparameterization. For a fixed-sample-size case, suppose we observe the K-dimensional time series at T time points, $\{\mathbf{y}_t | t = 1, \ldots, T\}$. The dependence structure between data points can be modeled through a VAR(p) model. Our goal is to select a subset $\mathcal{S} \subset \{1, \ldots, T\}$ of samples over which the estimation of the model can be efficiently performed. The least square estimator of $\boldsymbol{B}$ based on the selected sample then becomes Hamilton (1994)

$$\hat{\mathbf{B}}_S = \left(\sum_{t \in \mathcal{S}} \mathbf{x}_t \mathbf{x}_t' \right)^{-1} \left(\sum_{t \in \mathcal{S}} \mathbf{x}_t \mathbf{y}_t' \right).$$

The *leverage score sampling* method finds subset $\mathcal{S} = \mathcal{S}_{lev}$ according to the sampling rule

$$\mathcal{S}_{lev} = \{t = 1, \ldots, T |\ h_{tt} \triangleq \mathbf{x}_t' \boldsymbol{\Gamma}^{-1} \mathbf{x}_t > r^2\} \tag{3.3}$$

for some threshold $r > 0$, where we recall that $\mathbf{\Gamma} = \mathrm{E}[\mathbf{x}_t\mathbf{x}_t']$ is the covariance matrix. The choice of r is based on the quantile of a probability distribution of normalized data points. In practice, the unknown $\mathbf{\Gamma}^{-1}$ in (3.3) is replaced by an estimate based on a pilot sample, denoted as Ω, and the regressor vector $\mathbf{x}_t$ is constructed based on the VAR model. The quadratic form $h_{tt} = \mathbf{x}_t'\Omega\mathbf{x}_t$ can be viewed as the (unscaled) statistical leverage score.

The leverage score sampling can be summarized as, if for sample stretch $(\mathbf{x}_t, \mathbf{y}_t)$, the Mahalanobis distance satisfies $\sqrt{\mathbf{x}_t'\Omega\mathbf{x}_t} > r$, then we include t in subset $\mathcal{S}_{lev}$. As illustrated in Fig. 3.1, the normalized data points outside the ellipse are selected into $\mathcal{E}_{lev}$, where the normalization is based on their statistical leverage scores. The rate of sampling, $|\mathcal{S}_{lev}|/T$, is determined by the quantile r that measures the proportion of information selected rather than a prespecified sample size.

LSS simultaneously achieves the following goals

1. Improving the estimation efficiency of $\hat{\mathbf{B}}_{\mathcal{S}_{lev}}$ by reducing its estimation uncertainty;

2. Selecting a small set of samples to improve the computational efficiency;

3. Preserving the dependence structure in the sampling procedure since the data stretch $(\mathbf{x}_t, \mathbf{y}_t) = ((\mathbf{y}_{t-1}', \mathbf{y}_{t-2}', \cdots, \mathbf{y}_{t-p}')', \mathbf{y}_t)$ is the smallest sampling unit for any $t = 1, \ldots, T$;

4. Leading to streaming and decentralized algorithms.

3.3.1 Leverage Score Sampling for Streaming Time Series

The fundamental characteristic of a sampling method in the streaming setting, which distinguishes itself from its off-line version, is that streaming sampling requires a real-time decision-making mechanism. The LSS method utilizes a single-pass streaming procedure that calculates the leverage score in real-time so that one can make an immediate decision on sampling the current data point or not. In the streaming setting, we can first initialize Ω as an estimator of the precision matrix $\mathbf{\Gamma}^{-1}$ (see, e.g., e.g. Chen et al. (2013); Cai et al. (2016)) based on a pilot sample, and then periodically update it using new stream samples, so that Ω is a consistent estimator for the model precision matrix $\mathbf{\Gamma}^{-1}$. For computational advantage, an infrequent update of Ω is desired. Then the streaming leverage score and the corresponding sampling criterion is given by

$$\tilde{h}_{tt} \triangleq \mathbf{x}_t' \Omega \mathbf{x}_t > r^2 \tag{3.4}$$

to select the important data point in real time, which is a single-pass procedure and only requires linear computation time with respect to the model dimension Kp.

Streaming time series also requires an online method to continuously aggregates past data, updating the current estimate of parameter to incorporate the information obtained from the new data. As the streaming data comes in sequentially, we would like to update the estimate of the parameter $\mathbf{B}$, sequentially as well. With a slight abuse of notation, we use $\mathbf{B}_t$ to denote the estimate of the parameter $\mathbf{B}$ using LSS method at time t. Hence, for each time point t in the selected subset $\mathcal{S}_{lev}$ up to

current time T, we find the estimate $\mathbf{B}_t$ through optimizing the ℓ_2 loss,

$$\mathbf{B}_t = \arg\min_{\mathbf{B}} \sum_{i \in \mathcal{S}_{lev} \cap \{1,\ldots,t\}} ||\mathbf{y}_i' - \mathbf{x}_i'\mathbf{B}||_2^2, \tag{3.5}$$

which is in the form of dynamic linear model (DLM) Lai and Wei (1982); Pole et al. (1994), where the observation vector at time t becomes, $\mathbf{y}_t = \mathbf{B}_t'\mathbf{x}_t + \mathbf{e}_t$ and the underlying state vector satisfies $\mathrm{E}\mathbf{B}_t = \mathrm{E}\mathbf{B}_{t-1} = \mathbf{B}_0$ (unbiased).

There are plenty choices to solve the DLM in (3.5) , for example the classical Kalman filter. The Kalman filter Kalman (1960); Grewal (2011) updates the state vector $\mathrm{vec}(\mathbf{B}_t)$ for $\forall\ t \in \mathcal{S}_{lev}$. The updates of the parameter $\mathrm{vec}(\mathbf{B}_t)$ depends on accumulating the corresponding values themselves while streaming, and do not require accessing previous data points. It is important to note that, our LSS method is independent of the choice of the DLM solver in (3.5). The leverage score sampling for streaming time series can, therefore, run in constant memory and at a computational cost constant in time.

3.4 Decentralized Leverage Score Sampling

When the multidimensional streams are observed in a decentralized environment, the LSS method can be efficiently applied in parallel into asynchronous decentralized optimization algorithm by exploiting to VAR model structure. The leverage score and sampling criterion defined in (3.4) can be computed on each dimension in parallel and asynchronously under the decentralized setting.

The decentralized architecture is needed as long as the streams dimension K is

Algorithm 3: Online Asynchronous Decentralized Leverage Score Sampling

Require: Precision matrix Ω (updated periodically), quantile r.
Broadcast initial value of parameter $\mathbf{B}_0$ and covariance $\mathbf{P}_0$.

while $t > 0$ **do**
  while node $j \in [1, \ldots, K]$ in parallel **do**
    Receive the local data $y_t^{(j)}$ *without delay*, and the neighborhood data *with arbitrary delay*
    Send out the local data $y_t^{(j)}$ to neighbors
    Wait until $\mathbf{x}_{\tau_j}$ is complete for some $\tau_j \leq t$
    if $\tilde{h}_{tt}^{(j)} = \mathbf{x}'_{\tau_j} \Omega \mathbf{x}_{\tau_j} > r^2$ **then** {▷ LSS}
      Update $\boldsymbol{\beta}_{\tau_j}^{(j)}$ and $\mathbf{P}_{\tau_j}$ according to the local Kalman filter (3.7) and (3.8)
    else
      $\boldsymbol{\beta}_{\tau_j}^{(j)} = \beta_{\tau_j - 1}^{(j)}$ **and** $P_{\tau_j} = P_{\tau_j - 1}$
    end if
    Transmit the local estimate $\boldsymbol{\beta}^{(j)}$ to neighbors and receive neighbors' estimation
    Set $\tau_j \leftarrow \tau_j + 1$
    return $\mathbf{B}_{\tau_j} = [\boldsymbol{\beta}_{\tau_j}^{(1)}, \ldots, \boldsymbol{\beta}_{\tau_j}^{(j)}, \ldots, \boldsymbol{\beta}_{\tau_j}^{(K)}]$
  end while nodes
end while t

large or distributed physically apart in a network that accessing the data streams on a single machine is impossible. More specifically, a decentralized system lacks a fusion center (a centralized computing node) and may be communication-restricted, which requires a communication-efficient information diffusion strategy in the design of the decentralized algorithm. As illustrated in Fig. 3.2, we use neighborhood-based communication strategy in our sampling method and parameter estimation.

Note that the problem of (3.5) can be decomposed into K subproblems by taking advantage of the VAR model structure. We assume that, without loss of general-

ity, each node in the network observes one dimension data of the multidimensional streams. The selection criterion $\mathcal{S}_{lev}^{(j)}$ for node j becomes,

$$\tilde{h}_{tt}^{(j)} = \mathbf{x}'_{\tau_j} \Omega \mathbf{x}_{\tau_j} > r^2,$$

as long as the node j receives its local copy of data $\mathbf{x}_{\tau_j}$ at local time τ_j. We express the parameter matrix $\mathbf{B}$ as a block matrix with column vectors

$$\mathbf{B} = [\boldsymbol{\beta}^{(1)}, \boldsymbol{\beta}^{(2)}, \ldots, \boldsymbol{\beta}^{(K)}]$$

with $\boldsymbol{\beta}^{(j)}$ being the jth column of $\mathbf{B}$ for $j = 1, \ldots, K$. Hence for node j with its local time τ_j, the j-th subproblem is

$$\boldsymbol{\beta}_{\tau_j}^{(j)} = \arg\min_{\boldsymbol{\beta}^{(j)}} \sum_{\tau_j \in \mathcal{S}_{lev}^{(j)} \cap \{1,\ldots,t\}} ||y_{\tau_j}^{(j)} - \mathbf{x}'_{\tau_j} \boldsymbol{\beta}^{(j)}||_2^2, , \tag{3.6}$$

where $y_{\tau_j}^{(j)}$ is the jth element of $\mathbf{y}_{\tau_j}$, $\boldsymbol{\beta}_{\tau_j}^{(j)}$ is the estimate of $\boldsymbol{\beta}^{(j)}$ at time τ_j, for $j = 1, \ldots, K$. Those $\boldsymbol{\beta}^{(j)}$ can be estimated at each corresponding node locally as soon as $\mathbf{x}_{\tau_j}$ is completed at local time τ_j. From (3.6), we see that the sampling, parameter estimation and communication of nodes are uncoordinated. Each node j has its own local time τ_j and a global clock is not needed, resulting in an asynchronous algorithm. The algorithm then is running over the *ad hoc* network topology Wu and Stojmenovic (2004), i.e. decentralized network without fusion/data center to aggregate data asynchronously, where the nodes communicate with their neighbors and perform the local computation. The data from neighbors arrived sequentially

with delay depends on the distance in the network due to the limited communication, see Fig. 3.2.

Solving (3.6) can be done by various decentralized consensus optimization, e.g. decentralized gradient descent Yuan et al. (2016), decentralized ADMM Shi, Ling, Yuan, Wu, and Yin (Shi et al.), decentralized Kalman filter Olfati-Saber (2005); Cattivelli and Sayed (2010) and references therein, etc. We use diffusion strategies in Cattivelli and Sayed (2010) as an illustration to handle the parameter estimation and sampling, which allow asynchrony and delay in the decentralized consensus optimization. We use the local Kalman filter to estimate the local parameter $\boldsymbol{\beta}_{\tau_j}^{(j)}$ for j-th node and $\tau_j \in \mathcal{S}_{lev}$

$$\mathbf{P}_{\tau_j} = \mathbf{P}_{\tau_j - 1} - \mathbf{k}_{\tau_j} \mathbf{x}'_{\tau_j} \mathbf{P}_{\tau_j - 1} \tag{3.7}$$

$$\boldsymbol{\beta}_{\tau_j}^{(j)} = \boldsymbol{\beta}_{\tau_j - 1}^{(j)} + [y_{\tau_j}^{(j)} - \mathbf{x}'_{\tau_j} \boldsymbol{\beta}_{\tau_j - 1}^{(j)}] \mathbf{k}_{\tau_j}, \tag{3.8}$$

where $\mathbf{k}_{\tau_j} \triangleq \gamma_{\tau_j}^{-1} \mathbf{P}_{\tau_j - 1} \mathbf{x}_{\tau_j}$, and $\gamma_{\tau_j} \triangleq 1 + \mathbf{x}'_{\tau_j} \mathbf{P}_{\tau_j - 1} \mathbf{x}_{\tau_j}$ with $\mathbf{P}_{\tau_j}$ as the j-th local estimate of the precision matrix at local time τ_j. After getting the local estimate $\boldsymbol{\beta}_{\tau_j}^{(j)}$, the node exchanges the local estimate with neighbors to form a complete estimate of $\mathbf{B}_{\tau_j}$ at time τ_j. The theoretical guarantee of the consensus result of the algorithm can be found in Cattivelli and Sayed (2010). The algorithm is summarized in Algorithm 3.

3.5 Theoretical Justification of Leverage Score Sampling

The goal of this section is to provide the theoretical justification on the superiority of the LSS method over the Bernoulli sampling method. In Bernoulli sampling, we take the simple random sampling over time, i.e., conduct Bernoulli trail to with success probability q at each time t to select samples.

For simplicity of theoretical discussions, we shall assume below that $\Omega = \Gamma^{-1}$ exactly, while careful examination of the uncertainty of the Ω estimate is left as a future work.

The following theorem[2] establishes the asymptotic normality of the estimate $\hat{\mathbf{B}}_{\mathcal{S}_{lev}}$ based on the LSS-selected samples indexed by $\mathcal{S}_{lev}$.

Theorem 3.5.1. *Let $m = Kp$ and let $K \times K$ matrix $\Psi = E[\mathbf{e}_t \mathbf{e}_t']$. Define the $m \times m$ matrix*

$$\Gamma(r) = E\left[1_{\{\mathbf{x}_t'\Gamma^{-1}\mathbf{x}_t > r^2\}} \mathbf{x}_t \mathbf{x}_t'\right].$$

and suppose that it is non-singular. Then as $T \to \infty$,

$$\sqrt{T}(\mathrm{vec}(\hat{\mathbf{B}}_{\mathcal{S}}) - \mathrm{vec}(\mathbf{B})) \xrightarrow{d} N(\mathbf{0}, \Psi \otimes \Gamma(r)^{-1}). \tag{3.9}$$

In view of Theorem 3.5.1, the asymptotic covariance matrix of $\mathrm{vec}(\hat{\mathbf{B}}_{\mathcal{S}})$ dropping the scaling T^{-1} is

$$\Psi \otimes \Gamma(r)^{-1}. \tag{3.10}$$

[2]The proofs of all theorems can be found in Supplementary Material.

Our goal is to compare this covariance matrix with those arising from some naive sampling approaches. One option is to directly use a consecutive sample $(\mathbf{x}_t, \mathbf{y}_t)_{1 \le t \le Tq}$, $q \in (0,1)$. Another option is to employ an i.i.d. Bernoulli sampling: for each $t \in \{1 \ldots, T\}$, the sample $(\mathbf{x}_t, \mathbf{y}_t)$ is selected for regression with probability q independently. It turns out that these two options lead to the same asymptotic covariance matrix:

Theorem 3.5.2. *Under either the consecutive sampling or the i.i.d. Bernoulli sampling described above, we have as* $T \to \infty$,

$$\sqrt{T}(\mathrm{vec}(\hat{\mathbf{B}}) - \mathrm{vec}(\mathbf{B})) \xrightarrow{d} N(\mathbf{0}, q^{-1}\Psi \otimes \Gamma^{-1}), \tag{3.11}$$

where

$$\Gamma = E[\mathbf{x}_t \mathbf{x}_t'] = \Gamma(\boldsymbol{\beta} R^m). \tag{3.12}$$

To have a fair comparison with a leveraged-based sampling approach, we shall set

$$q = Q(m, r) = \Pr(\mathbf{x}_t' \Gamma^{-1} \mathbf{x}_t > r^2).$$

This ensures that the average sampling proportions across the different approaches are the same. Now the asymptotic covariance matrix (dropping scaling T^{-1}) of the consecutive or Bernoulli sampling approaches is

$$q^{-1}\Psi \otimes \Gamma^{-1}. \tag{3.13}$$

Comparing (3.10) with (3.13), we want to achieve

$$\Psi \otimes \Gamma(r)^{-1} \prec Q(m,r)^{-1}\Psi \otimes \Gamma^{-1}. \tag{3.14}$$

Relation (3.14) is equivalent to

$$Q(m,r)\Gamma \prec \Gamma(r). \tag{3.15}$$

See items 10.51(b) and 11.1(i) of Seber (2008).

Under the Gaussian assumption, the following theorem provides an expression for the minimum eigenvalue of $Q(m,r)\Gamma - \Gamma(r)$, which implies that (3.15) or equivalently (3.14) holds.

Theorem 3.5.3. *Suppose in (3.2) that* $\mathbf{e}_t$*'s are i.i.d.* $N(\mathbf{0}, \Psi)$*. Let* $m = Kp$ *and let* $\Gamma^{1/2} = P'\Lambda^{1/2}P$ *be a square root of the covariance matrix* Γ *in (3.12), where* P *is an orthogonal matrix and* $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$ *is a diagonal matrix of the eigenvalues of* Γ*. Let* $\mathcal{D}_r = \{\mathbf{x} \in \mathcal{B}R^m : \|\mathbf{x}\| > r\}$*,* $r \geq 0$*. Define* $\mathcal{E}_r$ *as the complement of an ellipsoid:*

$$\mathcal{E}_r = \Gamma^{1/2}\mathcal{D}_r.$$

(a) The (marginal) sampling probability is

$$\Pr(\mathbf{x}_t \in \mathcal{E}_r) = Q(m,r) = \Pr(\chi^2_m > r^2), \tag{3.16}$$

where χ^2_m *denotes a chi-squared random variable with* m *degrees of freedom.*

(b) The minimum eigenvalue of $Q(m,r)\Gamma - \Gamma(r)$ *is*

$$\lambda_{\min}\left[T(m,r) - Q(m,r)\right],$$

where $\lambda_{\min} = \min(\lambda_1, \ldots, \lambda_m) > 0$ *and*

$$T(m,r) = \frac{1}{m} E[\chi^2_m 1_{\{\chi^2_m > r^2\}}].$$

Some elementary calculus entails that $T(m,r) > Q(m,r)$ for any $r > 0$. Hence we have the following.

Corollary 3.5.4. *Under the setup of Theorem 3.5.3, the relation (3.14), namely, the asymptotic superiority of LSS over the Bernoulli sampling holds for any* $r > 0$.

Remark 1. If $\mathbf{e}_t$ is non-Gaussian, then some symmetry explored in the proof of Theorem 3.5.3 is unavailable. Nevertheless, (3.14) is expected hold under a moderate departure from normality.

3.6 Experiments

In this section, we demonstrate the applicability of the LSS method on three experiments: the synthetic data with various settings and two real multidimensional streaming data. In all experiments, we compare our proposed LSS method against Bernoulli sampling method (hereafter, Bernoulli) and vanilla Kalman filter method (hereafter, Vanilla) Cattivelli and Sayed (2010) in a decentralized setting with fixed

network topology structure. The Vanilla method uses full observed data, while the LSS and Bernoulli methods take samples accordingly with the same sampling rate q. We assume that the current node can only access its own data in real time, and the data transition from other nodes is delayed by the distance in the network connectivity to the current node. The results show the distinguishing features of our LSS method: accurate in parameter estimation with faster and better convergence at different sampling rate q, and computation efficiency with shorter execution time.

3.6.1 Synthetic Data

To compare accuracy and efficiency of parameter estimation in the streaming setting at different sampling rates q, dimensions K and lags p, we perform simulation study on synthetic data and report the estimation error $||\mathbf{B}_t - \mathbf{B}||_F$. The simulation data is generated by two settings: the first one (used in Fig. 3.3 (a)-(c)) is a 10-dimensional stationary VAR(3) process for 10 nodes, i.e., $K = 10,\ p = 3$ and the second one (used in Fig. 3.3 (d)-(f)) is a 30-dimensional stationary VAR(1) process for 30 nodes, i.e., $K = 30,\ p = 1$. The topology structure and the connectivity of nodes is created randomly at the beginning of the simulation and then applied to all methods Sayed et al. (2014). The first 200 data points from all nodes are used as pilot samples to obtain the estimate of Ω for each setting. In each subplot of Fig. 3.3, the result is compared by the estimation error, $||\mathbf{B}_t - \mathbf{B}||_F$, against time T, with 100 independent replicates, on different sampling rate $q \in [0.1, 0.2, 0.5]$ and two settings of K and p.

Fig. 3.3 shows that our method converges significantly faster (high accuracy and efficiency) than Bernoulli method, and converge as fast or slightly faster than the

Vanilla method that uses full data points in all test cases. In addition, LSS takes fewer computational steps (require fewer samples) than the Bernoulli method to achieve convergence. Fig. 3.5(d) shows the average elapsed time of 100 trials of the three methods. It can be seen that the time consumption of LSS is much smaller than vanilla Kalman filter and similar to the Bernoulli, while LSS achieves better estimation results than both of the other two methods, especially comparing to Bernoulli sampling. From Fig. 3.3(a) and (c), the advantages of the LSS method are more obvious when the sample size is small.

3.6.2 Real Data

The LSS, Bernoulli and Vanilla methods are implemented on two real datasets to compare the prediction error, $||\mathbf{y}_t - \hat{\mathbf{y}}_t||_2$, since the VAR model parameters are unknown for real data. In both experiments, the first 2000 data points are used as pilot samples.

Seismic Data: We consider the seismic data that records the wave amplitude (mm/s) from earthquake sequences in Oklahoma collected on October 26, 2014 Chen et al. (2016). The data contains 17 sensors with 17,698 time-steps. The VAR(3) model was chosen based on the analysis of the pilot sample. Fig. 3.4 shows the prediction error of the seismic data estimation. LSS outperforms the Bernoulli method, and it can achieve comparable or better prediction than the Vanilla method. From the first-order parameter matrices $\mathbf{\Phi}_1$ shown in Fig. 3.5, we see that LSS (a) and vanilla Kalman filter (c) perform similar estimation, while Bernoulli method has several off-diagonal unusual patterns. Combine Fig. 3.5 and Fig. 3.4, we see that

Bernoulli method failed to capture the correlation information in the seismogram so that the prediction reflects a severer bias and delay.

Gas Sensor Array: We do another experiment on the UCI dynamic gas mixtures dataset Fonollosa and Huerta (2015); Fonollosa et al. (2015). The data uses 16 chemical sensors at a sampling frequency of 100 Hz and records $4,208,261$ time-steps of Ethylene and CO mixture in air. For our experiments, we use data from 15 sensors to build a VAR(3) model[3]. A snapshot of the prediction error is shown in Fig. 3.6. It is clear that LSS captures the correct patterns in streams and performed superior or comparable to the Vanilla method that using the full data points.

3.7 Conclusion

We develop a novel online leverage score sampling method for efficiently estimating the temporal dependence of streaming multidimensional time series in an asynchronous decentralized environment. We prove that leverage score sampling yields a lower parameter estimation variance by selecting informative samples in infinite-sample streaming time series. Our future work includes, from the theoretical perspective, finding an optimal selection criterion under a more general (such as nonlinear or nonparametric) dynamic streaming model, and from the application perspective, extending the sampling scope to irregular-sampled high-dimensional random field streams, such as medical imaging real-time diagnosis, video and audio summarization, and environmental monitoring.

[3]We drop data from one sensor due to incomplete observation and low quality of the data.

3.8 Proofs

Proof of Theorem 3.5.1.

Let $\mathbf{U}_t = 1_{\{\mathbf{X}_t \in \mathcal{E}_r\}} \mathbf{X}_t$. By equation (3.2), we have

$$\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) = \left(\frac{1}{n} \sum_{t=1}^{n} \mathbf{U}_t \mathbf{U}_t' \right)^{-1} \left(\frac{1}{\sqrt{n}} \sum_{t=1}^{n} \mathbf{U}_t \mathbf{e}_t' \right), \tag{3.17}$$

which is understood as $-\sqrt{n}\boldsymbol{\beta}$ if the invertibility fails. Note that

$$\mathrm{E}[\mathrm{vec}(\mathbf{U}_t \mathbf{e}_t') \, \mathrm{vec}((\mathbf{U}_t \mathbf{e}_t')']' = \Omega \otimes \Gamma(r). \tag{3.18}$$

For any column vector $\mathbf{a} \in \mathbb{R}^{K^2 p}$, the linear combination $\mathbf{a}' \, \mathrm{vec}(\mathbf{U}_t) \mathbf{e}_t$ forms a stationary martingale difference in t with respect to the filtration $\mathcal{F}_t = \sigma(\mathbf{e}_i, \ i \leq t)$ since $\mathbf{U}_t$ is $\mathcal{F}_{t-1}$-measurable and $\mathbf{e}_t$ is centered and independent of $\mathcal{F}_{t-1}$. By (3.18) and the Martingale Central Limit Theorem (Theorem 35.12 of [Billingsley 1995 Probability and Measure 3rd ed.]), as $n \to \infty$,

$$\frac{1}{\sqrt{n}} \sum_{t=1}^{n} \mathbf{a}' \, \mathrm{vec}(\mathbf{U}_t \mathbf{e}_t') \xrightarrow{d} N(0, \mathbf{a}' \Omega \otimes \Gamma(r) \mathbf{a}).$$

In view of the Cramer-Wold Device, we have thus shown that as $n \to \infty$,

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} \mathrm{vec}(\mathbf{U}_t \mathbf{e}_t') \xrightarrow{d} N(\mathbf{0}, \Omega \otimes \Gamma(r)). \tag{3.19}$$

On the other hand, each component of the $\mathbf{U}_t$ is a causal linear filter of i.i.d. (thus ergodic) $\mathbf{e}_t$, and is hence an ergodic sequence by Lemma 10.5 of [Kallenberg 2002

Foundations of Modern Probability 2nd ed]. Therefore, by the Birkhoff Ergodic Theorem (Theorem 10.6 of [Kallenberg]) applied to each entry, one has almost surely as $n \to \infty$ that

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{U}_t \mathbf{U}_t' \to \Gamma(r). \tag{3.20}$$

At last, notice that the invertible matrices of a fixed size form an open subset under the product topology. Hence $\frac{1}{n}\sum_{i=1}^{n} \mathbf{U}_t \mathbf{U}_t'$ is invertible with probability tending to one as $n \to \infty$. Combining (3.17), (3.19) and (3.20) yields (3.9). □

Proof of Theorem 3.5.2.

The case of consecutive sampling can be directly deduced from Theorem 3.5.1 by letting $E = \boldsymbol{\beta} R^m$ and substituting n by nq. For the Bernoulli sampling, the proof can be carried out similarly as the proof of Theorem 3.5.1. In particular, the indicator $1_{\{\mathbf{X}_t \in E\}}$ is replaced by i.i.d. Bernoulli(q) variables independent of the time series $(\mathbf{Y}_t)$, which still retains the martingale property used in the proof of Theorem 3.5.1.

□

Proof of Theorem 3.5.3.

(a) Since $\mathbf{e}_t$'s are Gaussian, for each $t \in \boldsymbol{\beta} Z$, $\mathbf{X}_t \sim N(\mathbf{0}, \Gamma)$. Let $\mathbf{X} = (X_1, \ldots, X_m) \overset{d}{:=} \mathbf{X}_t$, and let $\mathbf{Z} = \Gamma^{-1/2}\mathbf{X} \sim N(\mathbf{0}, I_m)$. Then

$$\Pr(\mathbf{X} \in \mathcal{E}_r) = \Pr(\mathbf{Z} \in \mathcal{D}_r) = \Pr(\chi_m^2 > r^2) = Q(m, r).$$

(b) For any column vector $\mathbf{a} \in \boldsymbol{\beta} R^m$ with $\|\mathbf{a}\| = 1$, define

$$\begin{aligned} F(\mathbf{a}; \mathcal{E}_r) &:= \mathbf{a}'(\Gamma(r) - Q(m,r)\Gamma)\mathbf{a} \\ &= \mathrm{E}\left[\left(\sum_{i=1}^m a_i X_i\right)^2 [1_{\{\mathbf{X}\in\mathcal{E}_r\}} - Q(m,r)]\right]. \end{aligned} \tag{3.21}$$

Let ϕ_Γ denote the density of $N(\mathbf{0}, \Gamma)$. Then by a change of variable $\mathbf{x} = \Gamma^{1/2}\mathbf{y}$,

$$\begin{aligned} F(\mathbf{a}; \mathcal{E}_r) &= \int (\mathbf{a}'\mathbf{x})^2 [1_{\mathcal{E}_r}(\mathbf{x}) - Q(m,r)]\phi_\Gamma(\mathbf{x}) d\mathbf{x} \\ &= \int (\mathbf{a}' P' \Lambda^{1/2} P\mathbf{y})^2 [1_{\mathcal{D}_r}(\mathbf{y}) - Q(m,r)]\phi_{I_m}(\mathbf{y}) d\mathbf{y}. \end{aligned}$$

Let $\mathbf{b} = (b_1, \ldots, b_m)' = P\mathbf{a}$. By orthogonality of P, we have $\|\mathbf{b}\| = 1$ as well. By a change of variable $\mathbf{z} = (z_1, \ldots, z_m)' = P\mathbf{y}$, and using the invariance of $d\mathbf{z}$, ϕ_I and $\mathcal{D}_r$ with respect to an orthogonal transform, we have

$$\begin{aligned} & F(\mathbf{a}(\mathbf{b}); \mathcal{E}_r) \\ &= \int \left(\sum_{i=1}^m b_i \lambda_i^{1/2} z_i\right)^2 [1_{\mathcal{D}_r}(\mathbf{z}) - Q(m,r)]\phi_{I_m}(\mathbf{z}) d\mathbf{z}. \end{aligned} \tag{3.22}$$

By the symmetry of $\mathcal{D}_r$ and ϕ_{I_m}, the "covariance"

$$\int z_i z_j [1_{\mathcal{D}_r}(\mathbf{z}) - Q(m,r)]\phi_{I_m}(\mathbf{z}) d\mathbf{z} = 0, \qquad \text{if } i \neq j.$$

Hence

$$F(\mathbf{a}(\mathbf{b}); \mathcal{E}_r) = \int \sum_{i=1}^{m} b_i^2 \lambda_i z_i^2 [1_{\mathcal{D}_r}(\mathbf{z}) - Q(m,r)] \phi_{I_m}(\mathbf{z}) d\mathbf{z}$$

$$= \left(\int z_1^2 [1_{\mathcal{D}_r}(\mathbf{z}) - Q(m,r)] \phi_{I_m}(\mathbf{z}) d\mathbf{z} \right) \left(\sum_{i=1}^{m} b_i^2 \lambda_i \right).$$

Note that

$$\min_{\|\mathbf{b}\|=1} \left(\sum_{i=1}^{m} b_i^2 \lambda_i \right) = \lambda_{min},$$

which is positive since Γ is non-singular by assumption. On the other hand, we have

$$\int_{\mathcal{D}_r} z_1^2 \phi_{I_m}(\mathbf{z}) d\mathbf{z} = \frac{1}{m} \int_{\mathcal{D}_r} \|\mathbf{z}\|^2$$

$$\phi_{I_m}(\mathbf{z}) d\mathbf{z} = \frac{1}{m} \mathrm{E}[\chi_m^2 1_{\{\chi_m^2 > r^2\}}] = T(m,r).$$

and

$$\int z_1^2 \phi_{I_m}(\mathbf{z}) d\mathbf{z} = 1.$$

Hence

$$\min_{\|\mathbf{b}\|=1} F(\mathbf{a}(\mathbf{b}); \mathcal{E}_r) = \lambda_{\min} \left[T(m,r) - Q(m,r) \right].$$

□

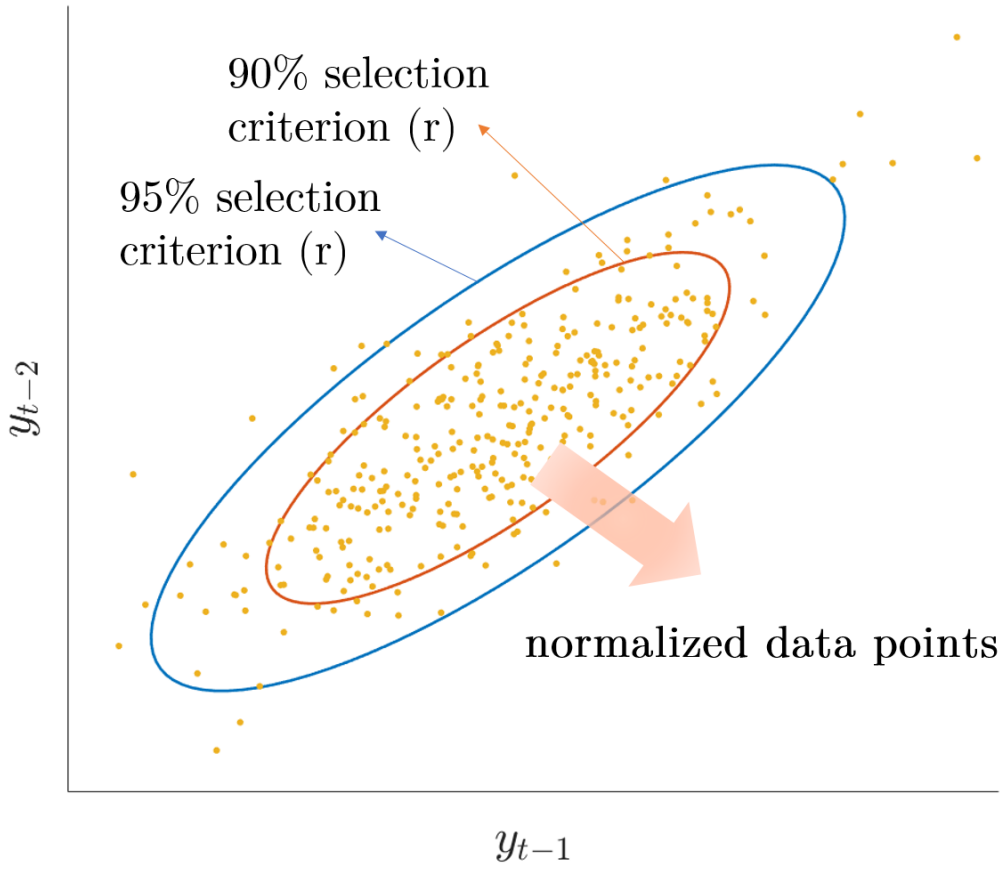

Figure 3.1: Illustration of sampling criterion: One-dimensional AR(2) time series $\{y_t\}_{t\in\beta Z}$ are plotted with axes lag-2 values y_{t-2} vs. lag-1 values y_{t-1}. Sampling criterion r is the quantile of a desirable chi-squared sampling probability distribution. The normalized data points outside the ellipses (orange: 90-th percentile; blue: 95-th percentile) will be selected by the LSS.

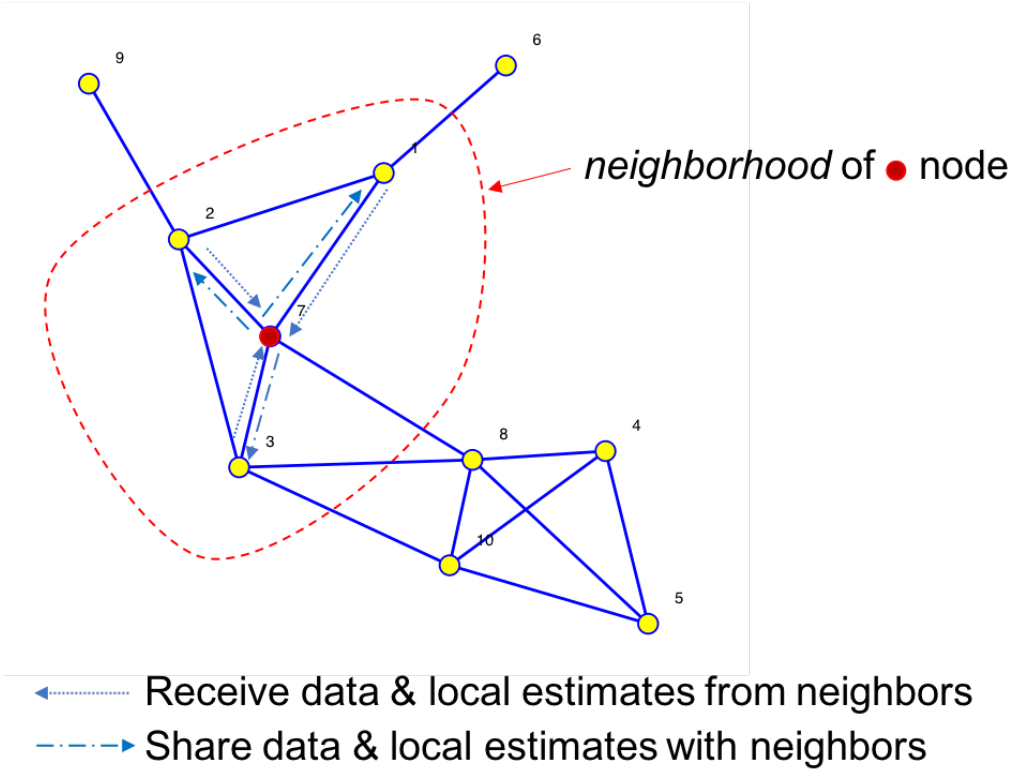

Figure 3.2: Diffusion strategy of the decentralized network. At every time t, node j collects a measurement $y_t^{(j)}$ and neighborhood data.

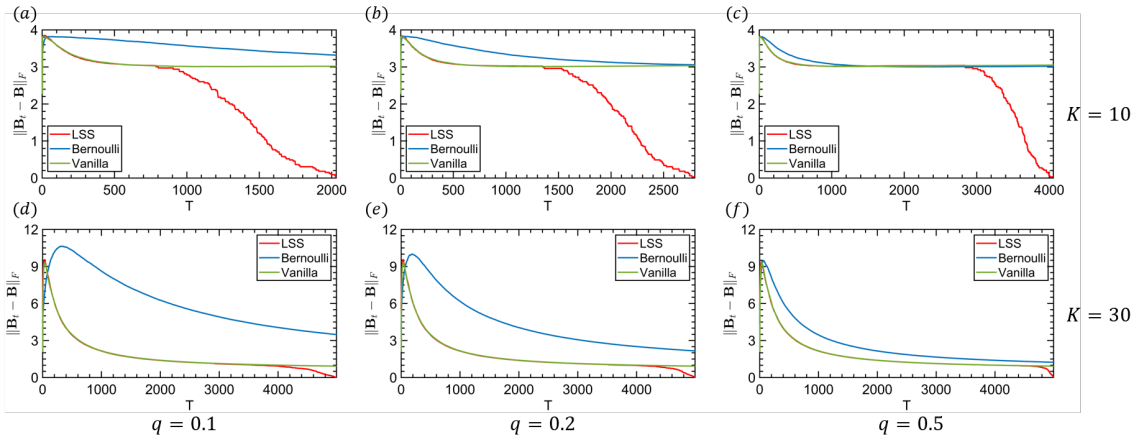

Figure 3.3: Each column shows the comparison of estimation error with different sampling rate (a): $q = 0.1$, (b): $q = 0.2$, and (c): $q = 0.5$. Fig.(a)-(c) show the results with a 10-dimensional stationary VAR(3) process and Fig.(d)-(f) show the results with a 30-dimensional stationary VAR(1) process. The estimation error, $||\mathbf{B}_t - \mathbf{B}||_F$ of LSS (red), Bernoulli (blue) and Vanilla (green) methods are plotted against time T with total time steps 5000.

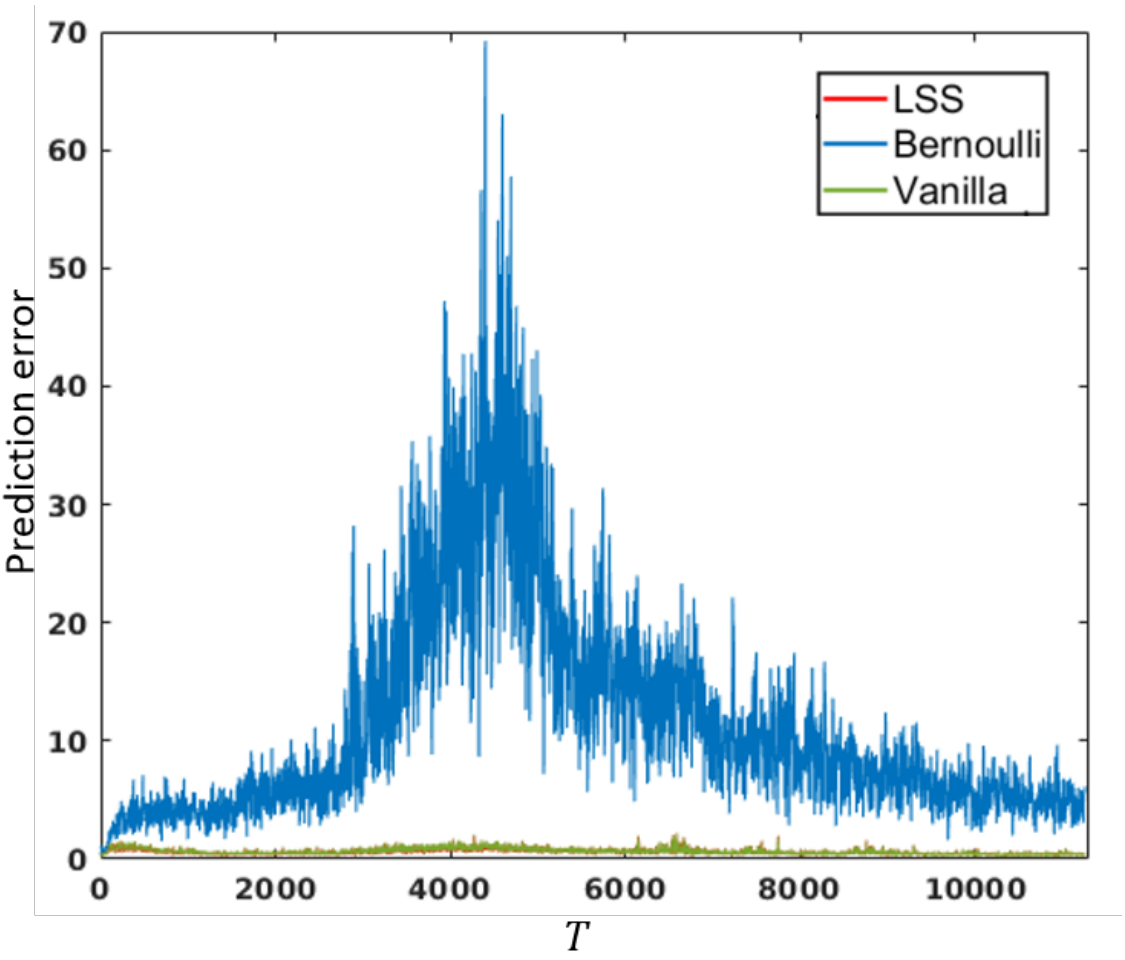

Figure 3.4: Prediction error from seismic data. The LSS (red) and Vanilla (green) error are tangled together in bottom of the plot.

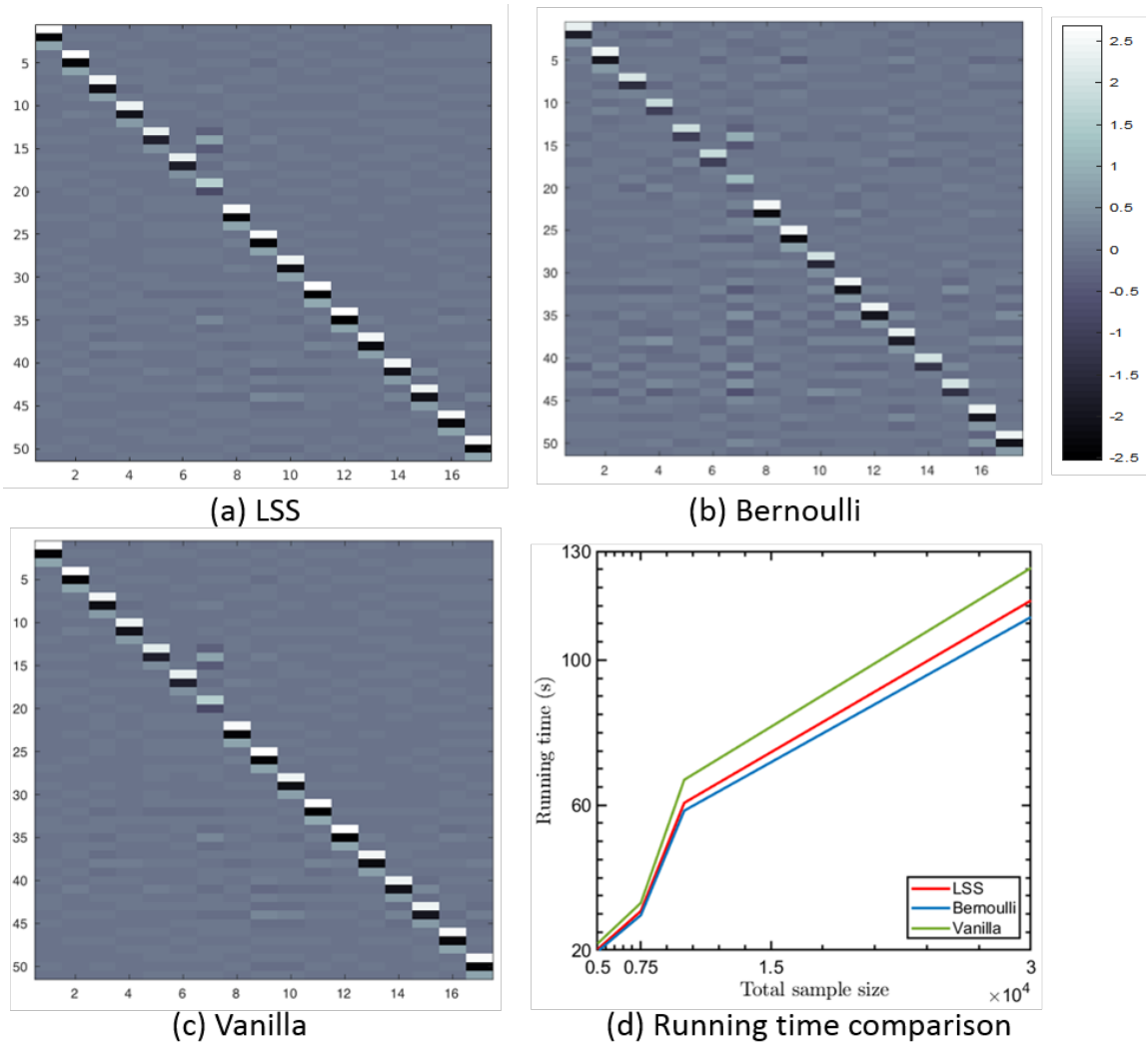

Figure 3.5: Seismic Data: Fig.(a)-(c) show first-order estimated parameter matrices $\mathbf{\Phi}_1$ at time $t = 8500$. Fig.(d) is the average elapsed time (seconds) of LSS(red), Bernoulli(blue) and Vanilla(green) methods over 100 replicates.

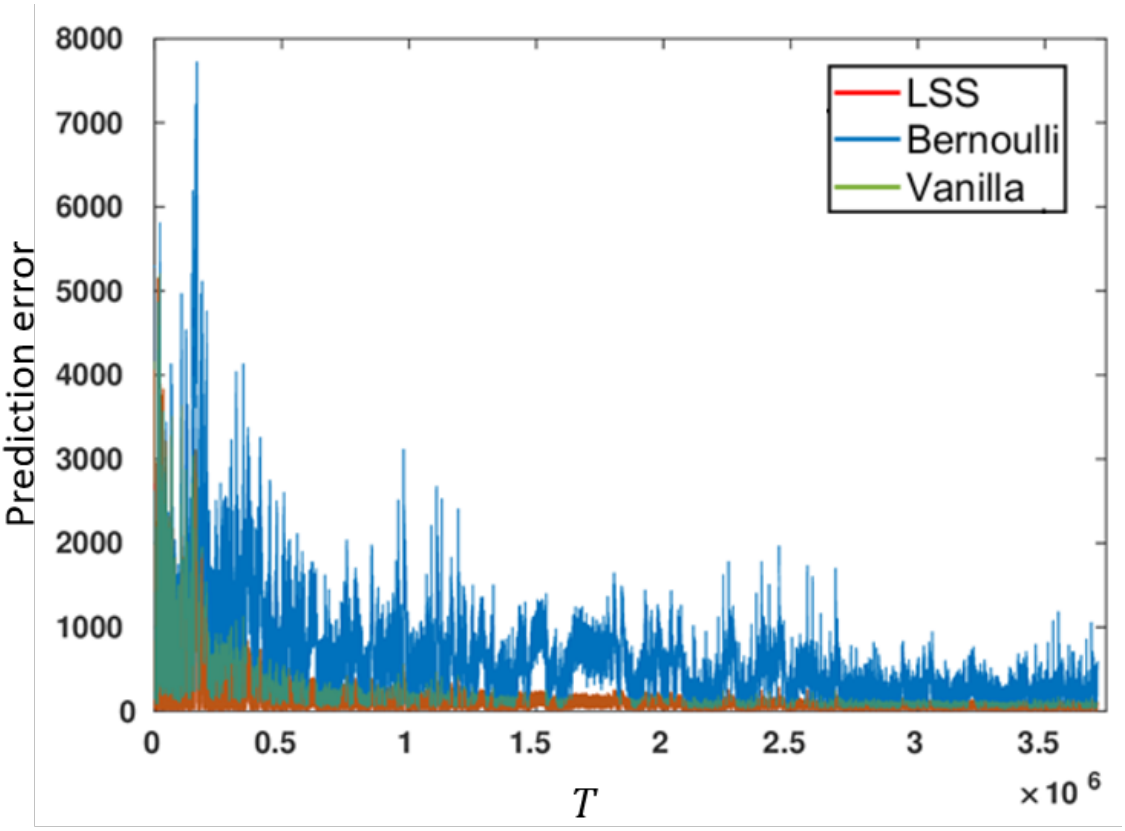

Figure 3.6: Prediction error from gas sensor data.

Chapter 4

Large Scale Randomized Learning Guided by Physical Laws with Applications in Full Waveform Inversion[1]

The rapid convergence rate, high fidelity learning outcome and low computational cost are key targets in solving the learning problem of the complex physical system. Guided by physical laws of wave propagation, in full waveform inversion (FWI), we learn the subsurface images through optimizing the media velocity model in a large scale non-linear problem. In this paper, we combine randomized subsampling

[1]R. Xie, F. Li, Z. Wang and W. Song, "LARGE SCALE RANDOMIZED LEARNING GUIDED BY PHYSICAL LAWS WITH APPLICATIONS IN FULL WAVEFORM INVERSION," 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Anaheim, CA, USA, 2018, pp. 66-70. doi: 10.1109/GlobalSIP.2018.8646507

techniques with a second-order optimization algorithm to propose the Sub-Sampled Newton (SSN) method for learning velocity model of FWI. By incorporating the curvature information, SSN preserves comparable convergence rate to Newton's method and significantly reduces the iteration cost by approximating the Hessian matrix through a non-uniform subsampling scheme. The numerical experiments demonstrate that the proposed SSN method has a faster convergence rate, and achieves a more accurate velocity model in terms of mean squared error than commonly used methods.

4.1 Introduction

The full waveform inversion (FWI) is a state-of-the-art method in subsurface imaging for providing the high-resolution estimate of the complex subsurface structure Operto et al. (2013); Virieux and Operto (2009). FWI uses measured wave-field data to learn the media velocity model that wave propagated through to invert the subsurface image. The learning procedure is through a nonlinear minimization of a penalized error function, which is the normed discrepancy between observed wave-field data and calculated data with the constraint of wave equation Tarantola (1984); Rao and Wang (2017). The calculated data is formulated by the most appropriate physical model for the system of wave propagation, which is mathematically described by a certain type of partial differential equation (PDE), such as wave equation, with unknown model coefficients, such as the velocity model in FWI.

Due to the complexity and ultra large data size of tracking the wave propagation,

the iterative optimization method is used to achieve the minimization of the error function so that we can find the optimal velocity model for subsurface imaging Pratt et al. (1998). The iterative optimization can be roughly divided into three part, forward modeling (solving PDE given the velocity model), error calculation (including gradient and Hessian) and velocity model update (first or second order iterative optimization) Tarantola (1984); Virieux and Operto (2009). Even though the relatively early introduction of the FWI technique, the intensive computation cost of these three components retard the development of FWI until the recent advances in high performance computing facilities along with the acquisition systems. As an example, for frequency domain FWI, the discretization of the Helmholtz equation, solving the forward modeling, calculating the error, and effectively updating the velocity, all involve the huge amount of data calculation and storage. The high computational burden calls for an efficient algorithm to provide a fast and accurate solution to the velocity model learning of FWI.

With the current development of FWI, the multiple-frequency or hierarchical data acquisition design promote FWI to more and more multi-scale and various field real data applications Plessix et al. (2010); Ernst et al. (2007). The development of data quality also requires a more realistic forward modeling and a more accurate velocity model learning method. Among the traditional methods for FWI, the first order methods such as gradient based methods Pratt (1999) and quasi-Newton methods such as the l-BFGS method Plessix and Mulder (2004); Pan et al. (2015) preserve the stable numerical performance and low computational costs, but their convergence rate are not satisfied Hu et al. (2011). On the other hand, the family of second

order method, the Newton-type methods Stakgold and Holst (2011); Métivier et al. (2014), has the advantages of fast convergence, exact update to second order Taylor approximation system, less geometric spreading and scattering artifacts Liu et al. (2014). However, in term of explicit calculation, the naive Newton-type methods of a large scale problem is impractical due to huge memory and storage cost and heavy computational burden. In recent years, developing fast and nearly scalable second order optimization methods is drawing more and more attentions on the optimizing and learning of complex optimization system Erdogdu and Montanari (2015); Xu et al. (2016); Pilanci and Wainwright (2017).

To conquer the high iteration computational cost of Newton-type method, extract the useful information from the highly correlated data and preserve the fast convergence rate of the second order method, we propose a novel Sub-Sampled Newton (SSN) method with non-uniform/important sampling scheme to deliver the fast and accurate subsurface imaging through FWI.

4.2 Algorithm Design

An accurate and fast optimization tool that can solve the complex subsurface imaging problem through FWI is crucially needed. Heavy computational burden has always been the bottleneck of the development in subsurface imaging, especially imaging through FWI. To deliver a high-resolution learning outcome of velocity model, we have to rely on the whole wave field information and accurate realization of the physical properties of wave propagation. The process that making efforts to get an

accurate subsurface imaging comes along with huge quantity of data for processing and storing. The efficient use of these information will accelerate the learning procedure of FWI. Relative to first-order methods, second-order methods enjoy plenty of advantages in the nonlinear optimization problem, such as the superior convergence, robustness to ill-conditioned problem, and global convergence guarantee under mild assumptions Sra et al. (2012).

Subsurface imaging through FWI, from computation perspective, is a constraint nonlinear learning problem. We consider the wave propagation as a multi-dimensional dynamic process $u(\boldsymbol{z}, \omega)$, where $\boldsymbol{z} = (z_1, \ldots, z_p)^T \in \mathbb{R}^p$ is a multi-dimensional covariate and ω is the additional covariate, if exists, that is associated with the varying model coefficients. We assume that the dynamic process $u(\boldsymbol{z}, \omega)$ follows an varying coefficients PDE,

$$\mathcal{F}\Big\{\boldsymbol{z}, \frac{\partial u(\boldsymbol{z},\omega)}{\partial z_1}, \ldots, \frac{\partial u(\boldsymbol{z},\omega)}{\partial z_p}, \frac{\partial^2 u(\boldsymbol{z},\omega)}{\partial z_1^2}, \frac{\partial^2 u(\boldsymbol{z},\omega)}{\partial z_1 \partial z_2}, \ldots, \boldsymbol{\theta}(\boldsymbol{z},\omega), f(\boldsymbol{z})\Big\} = 0, \tag{4.1}$$

where $\boldsymbol{\theta}(\boldsymbol{x}, \omega) = (\theta_1(\boldsymbol{x}, \omega), \ldots, \theta_m(\boldsymbol{x}, \omega))^T$ is the varying coefficient vector depending on $\boldsymbol{x}$ and ω, and $f(\boldsymbol{x})$ is a known "forcing term" or "source". In the application of FWI, the varying coefficients PDE is specified as the wave equation in the time domain, or equivalently, the Helmholtz equation in the frequency domain. The PDE in (4.1) then becomes the Helmholtz equation,

$$\left(\boldsymbol{m}(\boldsymbol{x}, \omega) + \nabla^2\right) u_\omega(\boldsymbol{x}) = -f_\omega(\boldsymbol{x}_s), \tag{4.2}$$

where $\boldsymbol{m}(\boldsymbol{x},\omega) = \frac{\omega^2}{v^2(\boldsymbol{x})}$, $u_\omega(\boldsymbol{x}) = u(\boldsymbol{x},t)e^{i\omega t}$, and $f_\omega(\boldsymbol{x}_s) = f(\boldsymbol{x}_s,t)e^{i\omega t}$ correspondingly with $\omega \in \Omega$ and $\boldsymbol{x},\boldsymbol{x}_s \in \mathcal{X} \subset \mathbb{R}^d$ given the frequency domain Ω and spatial domain $\mathcal{X}$. In the frequency domain Helmholtz equation, the varying coefficient $\boldsymbol{m}(\boldsymbol{x},\omega)$ depends not only on the spatial covariate $\boldsymbol{x}$, but also an additional covariate ω that is free from the derivatives in PDE.

In practice, we do not observe the dynamic process $u_\omega(\boldsymbol{x})$ and source term $f_\omega(\boldsymbol{x})$ on the whole domain. For the frequency domain FWI, instead we observe a surrogate $Y(\boldsymbol{x},\omega)$ at the locations where the sources and sensors are placed (usually on the surface of the ground) and within a certain frequency range. We assume that sources are located at source positions $\boldsymbol{x}_s$ with $s = 1,\ldots,n_s$, sensors are located at receiver positions $\boldsymbol{x}_r$ with $r = 1,\ldots,n_r$ and the frequency is observed at ω_w with $w = 1,\ldots,n_w$. The wave-field data we observed are denoted as

$$d_{obs,srw} = u_\omega(\boldsymbol{x}_r,\boldsymbol{x}_s,\omega_w,\boldsymbol{m}) + \epsilon_{srw}, \tag{4.3}$$

where ϵ_{srw}'s are assumed to be independent and identically distributed errors with zero mean and finite variance. Our goals are to estimate the varying model coefficient surface from the observed noisy data with the constraint of the PDE and to establish the statistical inference of the estimates Tarantola (1984).

The recorded data are acquired from an array of seismic receivers and denoted as $\mathbf{d}_{obs} = \{d_{obs,srw}\}_{srw}$. We track the wave propagation through PDE (4.2) given the velocity $\mathbf{m}$ and solve it numerically using finite-difference method Virieux (1986), where the wave fields are projected at the receiver positions $\boldsymbol{x}_r$. The velocity model $\mathbf{m} \triangleq [m(\mathbf{x}_1),...,m(\mathbf{x}_{N_zN_x})]$, $m(\mathbf{x}_i)$ indicates the squared-slowness value at the 2D

coordinate $\mathbf{x}_i, i = 1, ..., N_z N_x$, where N_z and N_x are the vertical and lateral grid number, respectively.

In velocity model learning, we estimate the varying model coefficients $\mathbf{m}$ by minimizing the penalized squared errors of the estimated PDE solution $\mathbf{d}_{cal}$ to the noisy data in (4.3),

$$\begin{aligned} E(\mathbf{m}) &= \frac{1}{2}||\mathbf{d}_{obs} - \mathbf{d}_{cal}||_2^2 + \lambda \mathcal{J}(\mathbf{m}) \\ &= \frac{1}{2}||\mathbf{d}_{obs} - u_\omega(\boldsymbol{x}, \mathbf{m})||_2^2 + \lambda \int \mathcal{F}\Big\{\boldsymbol{x}, \frac{\partial u(\boldsymbol{x},\omega)}{\partial x_1}, \ldots, \\ &\frac{\partial^2 u(\boldsymbol{x},\omega)}{\partial x_1^2}, \ldots, \frac{\partial^2 u(\boldsymbol{x},\omega)}{\partial x_1 \partial x_2}, \ldots, \boldsymbol{m}(\boldsymbol{x},\omega), f(\boldsymbol{x})\Big\} d\boldsymbol{x}, \end{aligned} \tag{4.4}$$

where the first term measures the goodness-of-fit of (4.3), and the second term measures the fidelity of (4.3) to the physical system, i.e. PDE model, defined in (4.1). The tuning parameter λ controls the trade-off between fitting to the observed data and fidelity to the physical system.

4.2.1 Method for Learning Velocity Model

We learn the varying coeffcient vector $\boldsymbol{\theta}(\boldsymbol{x}, \omega)$ in PDE model (4.1) in an alternating way with two nested levels of optimization. On one hand, we solve the PDE (4.1) given the current estimation of coefficient vector $\boldsymbol{\theta}(\boldsymbol{x}, \omega)$ through finite-difference method to generate the calculated data $\mathbf{d}_{cal}$, which is usually called forward modeling. On the other hand, we update the estimation of the coefficient vector $\boldsymbol{\theta}(\boldsymbol{x}, \omega)$ by minimizing the penalized squared errors of (4.4). This learning procedure is usually called inverse problem.

More specifically, in the application of FWI, to minimize $E(\mathbf{m})$, we search in an iterative manner, $\mathbf{m}_{k+1} = \mathbf{m}_k + \delta\mathbf{m}_k$, with $\delta\mathbf{m}_k$ being the optimal model perturbation at the k-th iteration step that minimizes $E(\mathbf{m})$. The optimal model perturbation comes from the expansion of $E(\mathbf{m}_k)$ in a small vicinity $\delta\mathbf{m}$ of $\mathbf{m}_k$ with a Taylor polynomial of degree two:

$$E(\mathbf{m}) = E(\mathbf{m}_k) + \delta\mathbf{m}^T\mathbf{g}_k + \frac{1}{2}\delta\mathbf{m}^T\mathbf{H}_k\delta\mathbf{m} + o(\|\delta\mathbf{m}\|^3),$$

where $\mathbf{g}_k \triangleq \partial E(\mathbf{m}_k)/\partial\mathbf{m}$ is the gradient of the error function $E(\mathbf{m})$ at $\mathbf{m}_k$ and $\mathbf{H}_k \triangleq \partial^2 E(\mathbf{m}_k)/\partial\mathbf{m}^2$ denotes the corresponding Hessian matrix. The Newton-type method is used to get the optimal model perturbation $\delta\mathbf{m}_k$ through the normal equation:

$$\mathbf{H}_k\delta\boldsymbol{m}_k = -\mathbf{g}_k, \tag{4.5}$$

and then update the velocity model according to

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \alpha_k[\mathbf{H}(\mathbf{m}_k)]^{-1}\mathbf{g}_k, \tag{4.6}$$

where α_k is the learning rate.

4.2.2 Gradient and Hessian

The conventional computation of gradient $\mathbf{g}_k$ and Hessian $\mathbf{H}_k$ can be specified as

$$\begin{aligned}\mathbf{g}_k &= \frac{\partial E(\mathbf{m}_k)}{\partial \mathbf{m}} = -\Re\left\{\left[\frac{\partial \mathcal{F}(\mathbf{m}_k)}{\partial \mathbf{m}}\right]^{\dagger}(\mathbf{d}_{obs} - \mathcal{F}(\mathbf{m}_k))\right\} \\ &= \Re\left\{\mathbf{J}_k^{\dagger}\delta\mathbf{d}_k\right\},\end{aligned} \tag{4.7}$$

where $\delta\mathbf{d}_k = \mathbf{d}_{obs} - \mathcal{F}(\mathbf{m}_k)$, $(\cdot)^{\dagger}$ denotes the adjoint of operator $(\cdot)$, $\Re\{z\}$ denotes the real part of complex number z, and $\mathbf{J}_k = -\partial\mathcal{F}(\mathbf{m}_k)/\partial\mathbf{m}$ is the Jacobian of $-\mathcal{F}(\cdot)$. By taking the derivative of $\mathbf{g}_k$, we get Hessian of $E(\cdot)$

$$\mathbf{H}_k = \frac{\partial^2 E(\mathbf{m}_k)}{\partial \mathbf{m}^2} = \Re\left\{\mathbf{J}_k^{\dagger}\mathbf{J}_k + \frac{\partial \mathbf{J}_k^{\dagger}}{\partial \boldsymbol{m}^T}[\delta\mathbf{d}_1^* \ \cdots \ \delta\mathbf{d}_k^*]\right\}, \tag{4.8}$$

where $(\cdot)^*$ denotes the conjugate of a complex number. Note that the first part of the Hessian matrix, $\Re\left\{\mathbf{J}_k^{\dagger}\mathbf{J}_k\right\}$, contains the most useful information of the curvature, and the second-order part of the Hessian matrix is easy to be contaminated be the noise since it usually presents numerous small or negative eigenvalues during the evaluation Métivier et al. (2013).

4.2.3 Sub-Sampled Newton (SSN) Method

To update the velocity model, we need to solve the normal equation (4.5). The computational bottleneck for solving equation (4.5) is the inverse of the Hessian matrix $\mathbf{H}_k$, which takes $O((N_z N_x)^3)$ flops and $O((N_z N_x)^2)$ memories. To preserve the quadratic convergence rate of the second order method, extract the useful information

Algorithm 4: Sub-sampled Newton method with Non-uniform Sampling

Input: Initial velocity $\boldsymbol{m}_0$, frequency ω, number of iteration K, sampling scheme $\mathcal{S}$ and solver $\mathcal{A}$.

Output: $\mathbf{m}_K$

for $k = 0, \ldots, K-1$ **do**

Construct the sampling distribution $\{\pi_i\}_{i=1}^N$ according to sampling scheme $\mathcal{S}$: leveraging score of $\mathbf{A}(\mathbf{m}) = [\mathbf{A}_1^T \cdots \mathbf{A}_N^T]^T$, say, $\pi_i = \frac{\|\mathbf{A}_i\|_F^2}{\|\mathbf{A}\|_F^2}$ Ma et al. (2015), or the block partial leverage score Xu et al. (2016).

Draw the sample, i.e. $S_{\mathbf{A}}^T\mathbf{A}$, according to an importance sampling distribution $\{\pi_i\}_{i=1}^N$, where $S_{\mathbf{A}}^T$ is a random sampling matrix.

Calculate randomized Hessian sketch $\widetilde{\mathbf{H}}(\mathbf{m}_k) = \sum_{l\in\mathcal{I}} \mathbf{A}_l^T(\mathbf{m}_k)\mathbf{A}_l(\mathbf{m}_k)/\pi_l + \mathbf{Q}(\boldsymbol{m}_k)$, where $\mathcal{I}$ is the subsampled indices set with size s.

Calculate $\mathbf{g}(\boldsymbol{m}_k)$, and learning rate α_k using line search.

Update velocity $\mathbf{m}_{k+1} = \mathbf{m}_k - \alpha_k[\widetilde{\mathbf{H}}(\mathbf{m}_k)]^{-1}\mathbf{g}_k$, using solver $\mathcal{A}$ to inverse Hessian $\widetilde{\mathbf{H}}(\mathbf{m}_k)$.

end

return $\mathbf{m}_K$

from the highly correlated data, and conquer such high per-iteration computational cost in forming and inverting the Hessian matrix, along with the line of Erdogdu and Montanari (2015); Xu et al. (2016); Pilanci and Wainwright (2017), we propose a randomized second order learning method called Sub-Sampled Newton (SSN) for FWI inverse problem.

We note that on realization of the Hessian of $E(\cdot)$ at certain receivers and frequencies,

$$\mathbf{H}(\mathbf{m}_k) = \sum_{s=1}^{N} \mathbf{A}_s^T(\mathbf{m}_k)\mathbf{A}_s(\mathbf{m}_k) + \mathbf{Q}(\mathbf{m}_k), \tag{4.9}$$

where $\sum_{s=1}^{N} \mathbf{A}_s^T(\mathbf{m}_k)\mathbf{A}_s(\mathbf{m}_k) = \Re\left\{\mathbf{J}_k^\dagger \mathbf{J}_k\right\}$, $\mathbf{Q}(\mathbf{m}_k) = \Re\left\{\frac{\partial \mathbf{J}_k^\dagger}{\partial \mathbf{m}^T}[\delta\mathbf{d}_1^* \ \cdots \ \delta\mathbf{d}_k^*]\right\}$, and $N = N_s N_r N_\omega$ with N_s, N_r, and N_ω are the source, receiver, and frequency numbers respectively.

As second order methods have been demonstrated to be effective in finding high precision minimizer, we propose a randomized second order learning method, Sub-Sample Newton, that exploit *non-uniform* sub-sampling of $\Re\{\mathbf{J}_k^\dagger \mathbf{J}_k\}$ to reduce the computational cost and achieve comparable convergence rate to Newton's method. We construct the non-uniform sub-sampling π_i over $\mathbf{A}_i(\boldsymbol{m}_k)$ with $i = 1, \ldots, N$ according to $\pi_i = \frac{\|\mathbf{A}_i\|_F^2}{\|\mathbf{A}\|_F^2}$ Ma et al. (2015); Zhang et al. (2018), or the block partial leverage score Xu et al. (2016) and take s sub-sample terms proportional to π_i. The details of the proposed method is summarized in Algorithm 4.

When the Hessian matrix dimension is large, the inexact solver, e.g. matrix free optimization Martens (2010) or conjugate gradient method Wright and Nocedal (1999), can be used to update the velocity model using a few iterations to produce a high-quality approximated solution to the normal equation.

4.3 Numerical Experiments

A 2D SEG/EAGE overthrust model (Fig. 4.1a) is used to test the proposed algorithm. The initial model is a smoothed version of the true model (Fig. 4.1b). The original model consists of 801×187 grid cells in a 2-D section with 25 m horizontal and vertical grid intervals. There are 100 sources and 100 receivers laid on the surface, which are spread out with 25 m spatial interval. A multi-scale inversion

approach is adopted in our numerical experiments in frequency bands $0.5 - 4$ Hz in every 0.5 Hz.

Fig. 4.1c – Fig. 4.1e demonstrate the learning results of (c) gradient decent, (d) l-BFGS and (e) Sub-Sampled Newton based on the data set of the lowest frequency band ($0.5-4$Hz). Fig. 4.2 provides the convergence rate comparison among the three methods, which shows the overall performances of them given the same number of forward modelling evaluations.

From both Fig. 4.1 and Fig. 4.2, we see that the proposed SSN recover the best velocity model among three methods. In Fig. 4.2, given the same numbers, 10 per 0.5 Hz, of forward modelling evaluations, SSN method converge faster than gradient decent and l-BFGS by achieving smaller mean squared error (MSE) across almost all frequencies. The gradient descent method has a large MSE, which implies that the numbers of forward modelling evaluations may not be sufficient for efficiently recover the velocity model or even obtaining a correct search direction. l-BFGS shows a better convergence performances as the frequency raise, but still significantly slower than SSN. Considering the expense of evaluating forward modeling, SSN outperforms the first order method, gradient decent and l-BFGS, and save the high per-iteration cost of vanilla Newton type method so that achieves the fast and accuracy learning results.

4.4 Conclusions

We present an efficient Sub-Sampled Newton (SSN) method to solve complex nonlinear system guided by physical laws with application to FWI problem. SSN significantly reduces the computational complexity while preserving a fast convergence property, by using the non-uniform subsampling techniques. SSN captures the important information in the second order term thus having a rapid rate of convergence. In numerical experiments of the Overthrust velocity model, we demonstrate that SSN significantly outperformed gradient descent and l-BFGS, resulting in high-quality inverted velocity model.

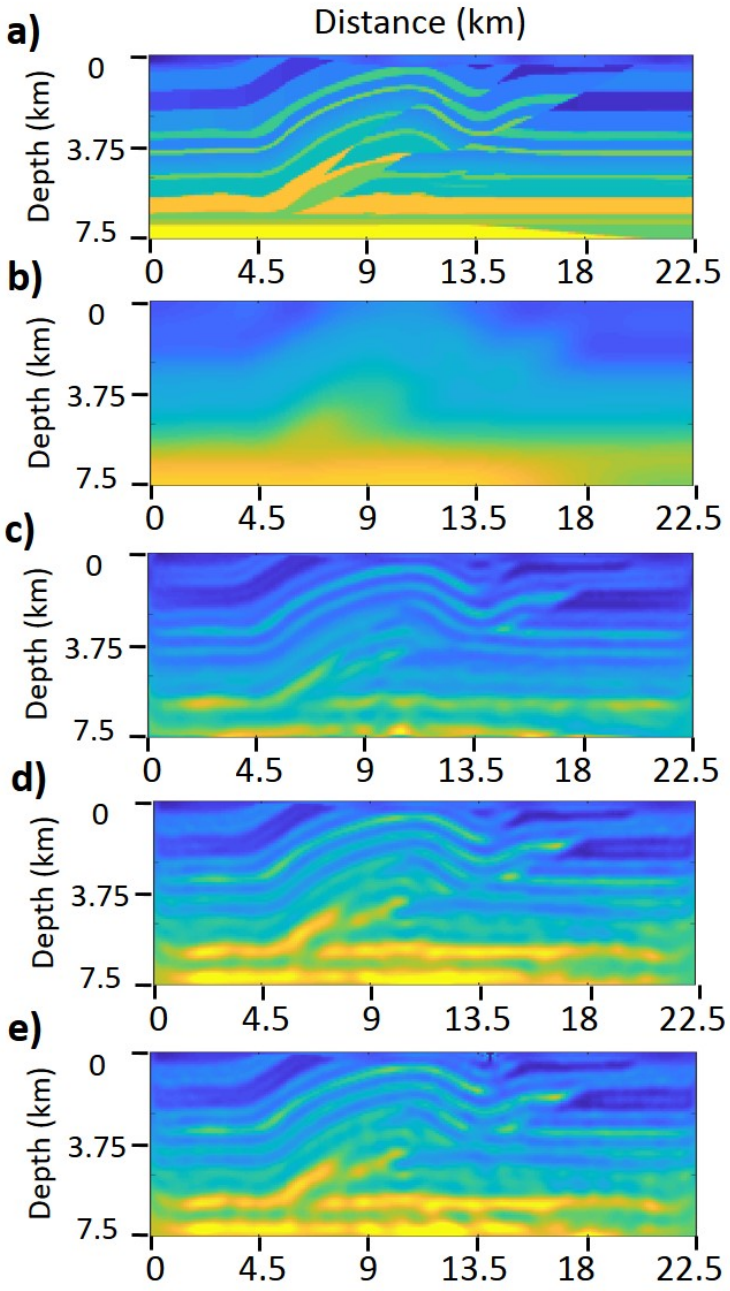

Figure 4.1: (a) Overthrust model, (b) Initial velocity model, (c-e) The learning results of (c) gradient decent, (d) l-BFGS and (e) Sub-Sampled Newton using the data set of the lowest frequency band ($0.5 - 4$Hz).

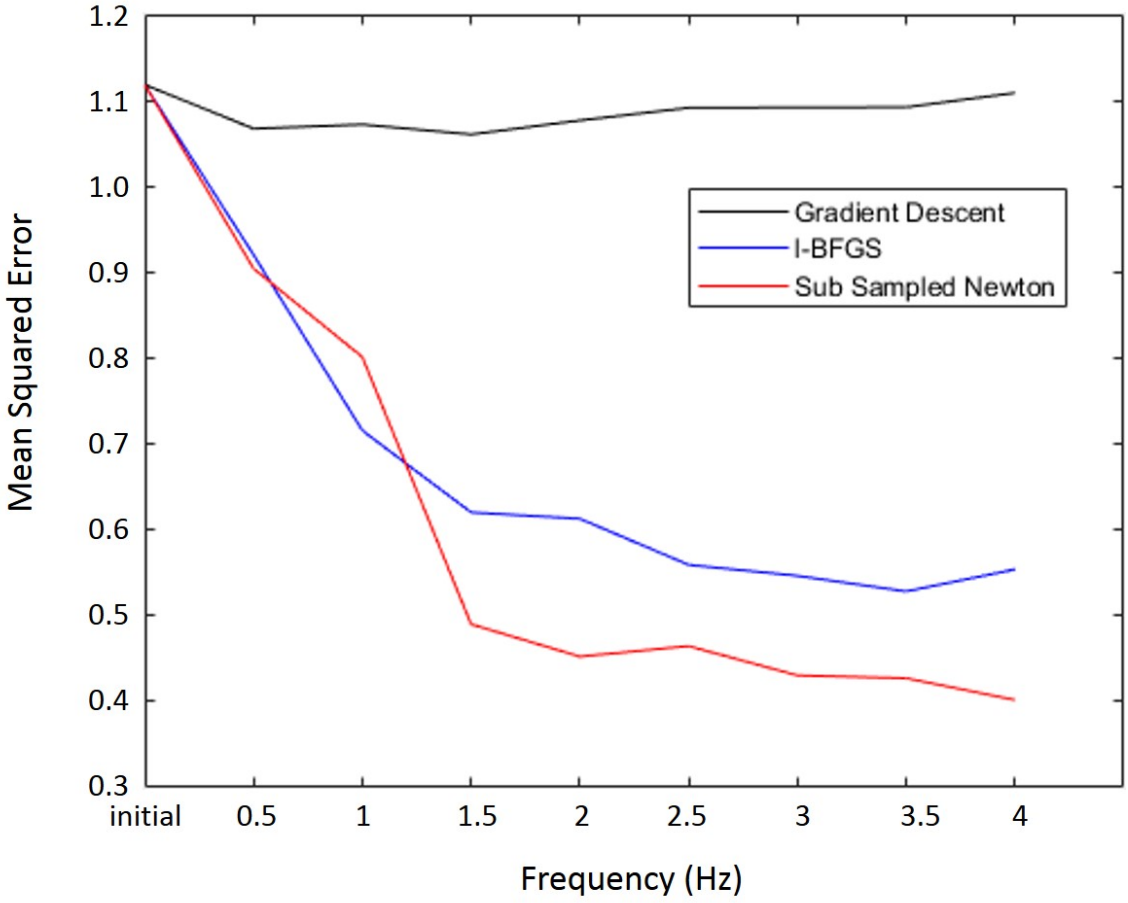

Figure 4.2: Convergence comparison of different methods. The mean squared error (MSE) of velocity model is plotted every 0.5 Hz with 10 forward modelling are evaluated at each of the 0.5 Hz frequencies.

Bibliography

Achlioptas, D., F. McSherry, and B. Schölkopf (2002). Sampling techniques for kernel methods. In *Advances in neural information processing systems*, pp. 335–342.

Ahmed, N. K., J. Neville, and R. Kompella (2014). Network sampling: From static to streaming graphs. *ACM Transactions on Knowledge Discovery from Data (TKDD) 8*(2), 7.

Ailon, N. and B. Chazelle (2010). Faster dimension reduction. *Communications of the ACM 53*(2), 97–104.

Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. In *Springer Series in Statistics*, pp. 199–213. Springer New York.

Alaoui, A. and M. W. Mahoney (2015). Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems*, pp. 775–783.

Anderson, T. and J. B. Taylor (1979). Strong consistency of least squares estimates in dynamic models. *The Annals of Statistics*, 484–489.

Anderson, T. W. (1959). On asymptotic distributions of estimates of parameters of stochastic difference equations. *Annals of Mathematical Statistics 30*, 676–687.

Anderson, T. W. (2011). *The Statistical Analysis of Time Series*, Volume 19. John Wiley & Sons.

Anderson, T. W. and J. B. Taylor (1976). Strong consistency of least squares estimates in normal linear regression. *Annals of Statistics 4*(4), 788–790.

Avron, H., P. Maymounkov, and S. Toledo (2010). Blendenpik: Supercharging LAPACK's least-squares solver. *SIAM Journal on Scientific Computing 32*, 1217–1236.

Babu, S. and J. Widom (2001). Continuous queries over data streams. *ACM Sigmod Record 30*(3), 109–120.

Barndorff-Nielsen, O. E. and D. R. Cox (1984). The effect of sampling rules on likelihood statistics. *International Statistical Review/Revue Internationale de Statistique*, 309–326.

Ben-Israel, A. and T. N. Greville (2003). *Generalized Inverses: Theory and Applications*, Volume 15. Springer Science & Business Media.

Bierens, H. J. (2001). Complex unit roots and business cycles: Are they real? *Econometric Theory 17*(5), 962–983.

Bifet, A., G. Holmes, R. Kirkby, and B. Pfahringer (2010). Moa: Massive online analysis. *Journal of Machine Learning Research 11*(May), 1601–1604.

Botta, A., W. De Donato, V. Persico, and A. Pescapé (2016). Integration of cloud computing and internet of things: a survey. *Future Generation Computer Systems 56*, 684–700.

Boutsidis, C., P. Drineas, and M. Magdon-Ismail (2014). Near-optimal column-based matrix reconstruction. *SIAM Journal on Computing 43*(2), 687–717.

Box, G. E., G. M. Jenkins, and G. C. Reinsel (2011). *Time Series Analysis: Forecasting and Control*, Volume 734. John Wiley & Sons.

Box, G. E., G. M. Jenkins, G. C. Reinsel, and G. M. Ljung (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Brockwell, P. J. and R. A. Davis (2013). *Time Series: Theory and Methods*. Springer.

Cai, T. T., Z. Ren, H. H. Zhou, et al. (2016). Estimating structured high-dimensional covariance and precision matrices: Optimal rates and adaptive estimation. *Electronic Journal of Statistics 10*(1), 1–59.

Calandriello, D., A. Lazaric, and M. Valko (2017a). Efficient second-order online kernel learning with adaptive embedding. In *Advances in Neural Information Processing Systems*, pp. 6140–6150.

Calandriello, D., A. Lazaric, and M. Valko (2017b). Second-order kernel online convex optimization with adaptive sketching. In *International Conference on Machine Learning*.

Cao, J., W. S. Cleveland, D. Lin, and D. X. Sun (2001). On the nonstationarity of internet traffic. In *ACM SIGMETRICS Performance Evaluation Review*, Volume 29, pp. 102–112. ACM.

Cattivelli, F. S. and A. H. Sayed (2010). Diffusion strategies for distributed kalman filtering and smoothing. *IEEE Transactions on Automatic Control 55*(9), 2069–2084.

Chan, N. H. and C. Z. Wei (1988). Limiting distributions of least squares estimates of unstable autoregressive processes. *Annals of Statistics 16*, 367–401.

Chen, X., R. E. Abercrombie, C. Pennington, X. Meng, and Z. Peng (2016). Source parameter validations using multiple-scale approaches for earthquake sequences in Oklahoma: implications for earthquake triggering processes. In *2016 AGU Fall Meeting, San Francisco, California*.

Chen, X., M. Xu, W. B. Wu, et al. (2013). Covariance and precision matrix estimation for high-dimensional time series. *The Annals of Statistics 41*(6), 2994–3021.

Chen, Y., G. Dong, J. Han, B. W. Wah, and J. Wang (2002). Multi-dimensional regression analysis of time-series data streams. In *Proceedings of the 28th international conference on Very Large Data Bases*, pp. 323–334. VLDB Endowment.

Chitnis, R., G. Cormode, H. Esfandiari, M. Hajiaghayi, A. McGregor, M. Monemizadeh, and S. Vorotnikova (2016). Kernelization via sampling with applications to finding matchings and related problems in dynamic graph streams. In *Proceedings*

of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms, pp. 1326–1344. Society for Industrial and Applied Mathematics.

Chow, Y. S., H. Robbins, and D. Siegmund (1971). *Great Expectations: The Theory of Optimal Stopping*. Houghton Mifflin Boston.

Christensen, K., M. Podolskij, N. Thamrongrat, and B. Veliyev (2017). Inference from high-frequency data: A subsampling approach. *Journal of Econometrics 197*(2), 245–272.

Clarkson, K. L., P. Drineas, M. Magdon-Ismail, M. W. Mahoney, X. Meng, and D. P. Woodruff (2013). The Fast Cauchy Transform and faster robust linear regression. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 466–477.

Clarkson, K. L. and D. P. Woodruff (2013). Low rank approximation and regression in input sparsity time. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pp. 81–90.

Cohen, M. B., Y. T. Lee, C. Musco, C. Musco, R. Peng, and A. Sidford (2015). Uniform sampling for matrix approximation. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pp. 181–190. ACM.

Cohen, M. B., C. Musco, and C. Musco (2017). Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1758–1777. SIAM.

Cohen, M. B., C. Musco, and J. Pachocki (2016). Online row sampling. *arXiv preprint arXiv:1604.05448*.

Cormode, G., S. Muthukrishnan, K. Yi, and Q. Zhang (2012). Continuous sampling from distributed streams. *Journal of the ACM (JACM) 59*(2), 10.

Cortez, P., M. Rio, M. Rocha, and P. Sousa (2012). Multi-scale internet traffic forecasting using neural networks and time series methods. *Expert Systems 29*(2), 143–155.

Dahlhaus, R. et al. (1997). Fitting time series models to nonstationary processes. *The annals of Statistics 25*(1), 1–37.

Dahlhaus, R. and L. Giraitis (1998). On the optimal segment length for parameter estimates for locally stationary time series. *Journal of Time Series Analysis 19*(6), 629–655.

Davis, J. L. and A. P. Annan (1989). Ground-penetrating radar for high-resolution mapping of soil and rock stratigraphy. *Geophysical prospecting 37*(5), 531–551.

Derezinski, M. and M. K. Warmuth (2017). Unbiased estimates for linear regression via volume sampling. In *Advances in Neural Information Processing Systems*, pp. 3087–3096.

Dette, H., A. Pepelyshev, and A. Zhigljavsky (2013). Optimal design for linear models with correlated observations. *The Annals of Statistics 41*(1), 143–176.

Dickey, D. A. and W. A. Fuller (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association 74*(366a), 427–431.

Dickey, D. A. and W. A. Fuller (1981). Likelihood ratio statistics for autoregressive time series with a unit root. *Econometrica: Journal of the Econometric Society*, 1057–1072.

Drineas, P., R. Kannan, and M. W. Mahoney (2006). Fast monte carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM Journal on Computing 36*(1), 132–157.

Drineas, P., M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff (2012a). Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research 13*, 3475–3506.

Drineas, P., M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff (2012b). Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research 13*(Dec), 3475–3506.

Drineas, P., M. W. Mahoney, and S. Muthukrishnan (2006a). Sampling algorithms for ℓ_2 regression and applications. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1127–1136.

Drineas, P., M. W. Mahoney, and S. Muthukrishnan (2006b). Subspace sampling and relative-error matrix approximation: Column-based methods. In *Approximation,*

Randomization, and Combinatorial Optimization. Algorithms and Techniques, pp. 316–326. Springer.

Drineas, P., M. W. Mahoney, and S. Muthukrishnan (2008). Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications 30*(2), 844–881.

Drineas, P., M. W. Mahoney, S. Muthukrishnan, and T. Sarlós (2010). Faster least squares approximation. *Numerische Mathematik 117*(2), 219–249.

Efraimidis, P. S. (2015). Weighted random sampling over data streams. In *Algorithms, Probability, Networks, and Games*, pp. 183–195. Springer.

Elhamifar, E. and M. Kaluza (2017a). Online summarization via submodular and convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Elhamifar, E. and M. C. D. P. Kaluza (2017b). Subset selection and summarization in sequential data. In *Advances in Neural Information Processing Systems*, pp. 1036–1045.

Erdogdu, M. A. and A. Montanari (2015). Convergence rates of sub-sampled newton methods. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pp. 3052–3060. MIT Press.

Ernst, J. R., A. G. Green, H. Maurer, and K. Holliger (2007). Application of a new 2d time-domain full-waveform inversion scheme to crosshole radar data. *Geophysics 72*(5), J53–J64.

Fonollosa, J. and R. Huerta (2015). Gas sensor array under dynamic gas mixtures data set.

Fonollosa, J., S. Sheik, R. Huerta, and S. Marco (2015). Reservoir computing compensates slow response of chemosensor arrays exposed to fast varying gas concentrations in continuous monitoring. *Sensors and Actuators B: Chemical 215*, 618–629.

Freedman, D. (1971). *Brownian Motion and Diffusion*. San Francisco, Holden-Day.

Fu, T.-C. (2011). A review on time series data mining. *Engineering Applications of Artificial Intelligence 24*(1), 164–181.

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

Galtchouk, L. and V. Konev (2011). On asymptotic normality of sequential LS-estimates of unstable autoregressive processes. *Sequential Analysis 30*(2), 117–144.

Gama, F., A. G. Marques, G. Mateos, and A. Ribeiro (2016). Rethinking sketching as sampling: Linear transforms of graph signals. *2016 50th Asilomar Conference on Signals, Systems and Computers*, 522–526.

Garofalakis, M., J. Gehrke, and R. Rastogi (2016). *Data Stream Management: Processing High-Speed Data Streams*. Springer.

Gephart, J. W. and D. W. Forsyth (1984). An improved method for determining the regional stress tensor using earthquake focal mechanism data: application to

the San Fernando earthquake sequence. *Journal of Geophysical Research: Solid Earth 89*(B11), 9305–9320.

Gilbert, A. C., M. J. Strauss, J. A. Tropp, and R. Vershynin (2007). One sketch for all: fast algorithms for compressed sensing. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pp. 237–246. ACM.

Golub, G. and C. Van Loan (1996). *Matrix Computations*. Baltimore: Johns Hopkins University Press.

Gong, B., W.-L. Chao, K. Grauman, and F. Sha (2014). Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems*, pp. 2069–2077.

Grambsch, P. (1983). Sequential sampling based on the observed Fisher information to guarantee the accuracy of the maximum likelihood estimator. *The Annals of Statistics*, 68–77.

Grewal, M. S. (2011). Kalman filtering. In *International Encyclopedia of Statistical Science*, pp. 705–708. Springer.

Guo, L. (1994). Further results on least squares based adaptive minimum variance control. *SIAM journal on control and optimization 32*(1), 187–212.

Hall, P., J. L. Horowitz, and B.-Y. Jing (1995). On blocking rules for the bootstrap with dependent data. *Biometrika 82*(3), 561–574.

Hallin, M. (1978). Mixed autoregressive-moving average multivariate processes with time-dependent coefficients. *Journal of Multivariate Analysis 8*(4), 567–572.

Hamilton, J. (1994). *Time Series Analysis*. Princeton University Press.

Harvey, A. C. (1990). *Forecasting, structural time series models and the Kalman filter*. Cambridge university press.

Hau, M. C. and H. Tong (1989). A practical method for outlier detection in autoregressive time series modelling. *Stochastic Hydrology and Hydraulics 3*, 241–260.

Haykin, S. et al. (1996). Adaptive filtering theory. *Englewood Cliffs, NJ: Prentice-Hall*.

Himberg, J., K. Korpiaho, H. Mannila, J. Tikanmaki, and H. T. Toivonen (2001). Time series segmentation for context recognition in mobile devices. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pp. 203–210. IEEE.

Hoffman, M., F. R. Bach, and D. M. Blei (2010). Online learning for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pp. 856–864.

Hu, B., Y. Chen, and E. Keogh (2013). Time series classification under more realistic assumptions. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 578–586. SIAM.

Hu, W., A. Abubakar, T. Habashy, and J. Liu (2011). Preconditioned non-linear conjugate gradient method for frequency domain full-waveform seismic inversion. *Geophysical Prospecting 59*(3), 477–491.

Huang, H. and S. P. Kasiviswanathan (2015). Streaming anomaly detection using randomized matrix sketching. *Proceedings of the VLDB Endowment 9*(3), 192–203.

Hubbard, S., K. Grote, and Y. Rubin (2002). Mapping the volumetric soil water content of a California vineyard using high-frequency GPR ground wave data. *The Leading Edge 21*(6), 552–559.

Isacks, B. and P. Molnar (1971). Distribution of stresses in the descending lithosphere from a global survey of focal-mechanism solutions of mantle earthquakes. *Reviews of Geophysics 9*(1), 103–174.

Jeganathan, P. (1988). On the strong approximation of the distribution of estimators in linear stochastic models, I and II: Stationary and explosive AR models. *Annals of Statistics 16*, 1283–1314.

Jörnsten, R. and B. Yu (2003). Simultaneous gene clustering and subset selection for sample classification via MDL. *Bioinformatics 19*(9), 1100–1109.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering 82*(1), 35–45.

Kapralov, M., Y. T. Lee, C. Musco, C. Musco, and A. Sidford (2017). Single pass spectral sparsification in dynamic streams. *SIAM Journal on Computing 46*(1), 456–477.

Karagiannis, T., M. Molle, M. Faloutsos, and A. Broido (2004). A nonstationary Poisson view of Internet traffic. In *INFOCOM 2004. Twenty-third AnnualJoint*

Conference of the IEEE Computer and Communications Societies, Volume 3, pp. 1558–1569. IEEE.

Keogh, E., S. Chu, D. Hart, and M. Pazzani (2001). An online algorithm for segmenting time series. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pp. 289–296. IEEE.

Kossmann, D., F. Ramsak, and S. Rost (2002). Shooting stars in the sky: An online algorithm for skyline queries. In *Proceedings of the 28th International Conference on Very Large Data Bases*, pp. 275–286. VLDB Endowment.

Koutis, I., G. L. Miller, and R. Peng (2010). Approaching optimality for solving SDD linear systems. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pp. 235–244. IEEE.

Kulesza, A., B. Taskar, et al. (2012). Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning 5*(2–3), 123–286.

Lahiri, S. N. (2003). *Resampling Methods for Dependent Data*. Springer New York.

Lai, T. and D. Siegmund (1983). Fixed accuracy estimation of an autoregressive parameter. *The Annals of Statistics*, 478–485.

Lai, T. and C. Wei (1983). Asymptotic properties of general autoregressive models and strong consistency of least-squares estimates of their parameters. *Journal of Multivariate Analysis 13*(1), 1–23.

Lai, T. L. (2001). Sequential analysis: some classical problems and new challenges. *Statistica Sinica*, 303–351.

Lai, T. L. and C. Z. Wei (1982). Least squares estimates in stochastic regression models with applications to identification and control of dynamic systems. *The Annals of Statistics*, 154–166.

Leland, W. E., M. S. Taqqu, W. Willinger, and D. V. Wilson (1993). On the self-similar nature of ethernet traffic. In *ACM SIGCOMM Computer Communication Review*, Volume 23, pp. 183–193. ACM.

Liang, G., N. Taft, and B. Yu (2006). A fast lightweight approach to origin-destination IP traffic estimation using partial measurements. *IEEE Transactions on Information Theory 52*(6), 2634–2648.

Lin, H. and J. Bilmes (2012). Learning mixtures of submodular shells with application to document summarization. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pp. 479–490. AUAI Press.

Liu, C., F. Gao, X. Feng, Y. Liu, and Q. Ren (2014). Memoryless quasi-newton (mlqn) method for 2d acoustic full waveform inversion. *Exploration Geophysics 46*(2), 168–177.

Ljung, L. and T. Söderström (1983). *Theory and practice of recursive identification*. MIT press.

Luts, J., T. Broderick, and M. P. Wand (2014). Real-time semiparametric regression. *Journal of Computational and Graphical Statistics 23*(3), 589–615.

Ma, P., M. Mahoney, and B. Yu (2014). A statistical perspective on algorithmic leveraging. *JMLR: Workshop and Conference Proceedings 32*, 91–99.

Ma, P., M. Mahoney, and B. Yu (2015). A statistical perspective on algorithmic leveraging. *Journal of Machine Learning Research 16*, 861–911.

Ma, P., M. W. Mahoney, and B. Yu (2014). A statistical perspective on algorithmic leveraging. In *Proceedings of the 31st International Conference on Machine Learning*.

Ma, P., M. W. Mahoney, and B. Yu (2015a). A statistical perspective on algorithmic leveraging. *Journal of Machine Learning Research 16*, 861–911.

Ma, P., M. W. Mahoney, and B. Yu (2015b). A statistical perspective on algorithmic leveraging. *The Journal of Machine Learning Research 16*(1), 861–911.

Mahoney, M. W. (2011). Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning 3*(2), 123–224.

Mahoney, M. W. et al. (2011). Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning 3*(2), 123–224.

Mahoney, M. W. and P. Drineas (2009a). CUR matrix decompositions for improved data analysis. *Proc. Natl. Acad. Sci. USA 106*, 697–702.

Mahoney, M. W. and P. Drineas (2009b). Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences 106*(3), 697–702.

Mann, H. B. and A. Wald (1943). On the statistical treatment of linear stochastic difference equations. *Econometrica 11*, 173–220.

Martens, J. (2010). Deep learning via hessian-free optimization. In *ICML*, Volume 27, pp. 735–742.

Mathioudakis, M. and N. Koudas (2010). Twittermonitor: trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, pp. 1155–1158. ACM.

Meng, X. and M. W. Mahoney (2013). Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pp. 91–100.

Meng, X., M. A. Saunders, and M. W. Mahoney (2014). LSRN: A parallel iterative solver for strongly over- or under-determined systems. *SIAM Journal on Scientific Computing 36*(2), C95–C118.

Métivier, L., F. Bretaudeau, R. Brossier, S. Operto, and J. Virieux (2014). Full waveform inversion and the truncated newton method: quantitative imaging of complex subsurface structures. *Geophysical Prospecting 62*(6), 1353–1375.

Métivier, L., R. Brossier, J. Virieux, and S. Operto (2013). Full waveform inversion and the truncated newton method. *SIAM Journal on Scientific Computing 35*(2), B401–B437.

Michalak, S., A. DuBois, D. DuBois, S. V. Wiel, and J. Hogden (2012). Developing systems for real-time streaming analysis. *Journal of Computational and Graphical Statistics 21*(3), 561–580.

Moreira-Matias, L., J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas (2013). Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems 14*(3), 1393–1402.

Musco, C. and C. Musco (2017). Recursive sampling for the nystrom method. In *Advances in Neural Information Processing Systems*, pp. 3836–3848.

Näther, W. (1985). Exact designs for regression models with correlated errors. *Statistics 16*(4), 479–484.

Olfati-Saber, R. (2005). Distributed kalman filter with embedded consensus filters. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pp. 8179–8184. IEEE.

Olivier, M., R. Eeles, M. Hollstein, M. A. Khan, C. C. Harris, and P. Hainaut (2002). The IARC TP53 database: new online mutation analysis and recommendations to users. *Human Mutation 19*(6), 607–614.

Operto, S., Y. Gholami, V. Prieux, A. Ribodetti, R. Brossier, L. Metivier, and J. Virieux (2013). A guided tour of multiparameter full-waveform inversion with multicomponent data: From theory to practice. *The Leading Edge 32*(9), 1040–1054.

Pan, W., K. A. Innanen, G. F. Margrave, and D. Cao (2015). Efficient pseudo-gauss-newton full-waveform inversion in the τ-p domain. *Geophysics 80*(5), R225–R14.

Papadimitriou, S., J. Sun, and C. Faloutsos (2005). Streaming pattern discovery in

multiple time-series. In *Proceedings of the 31st International Conference on Very Large Data Bases*, pp. 697–708. VLDB Endowment.

Papailiopoulos, D., A. Kyrillidis, and C. Boutsidis (2014). Provable deterministic leverage score sampling. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 997–1006. ACM.

Pilanci, M. and M. J. Wainwright (2017). Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. *SIAM Journal on Optimization 27*(1), 205–245.

Plessix, R.-E., G. Baeten, J. W. de Maag, M. Klaassen, Z. Rujie, and T. Zhifei (2010). Application of acoustic full waveform inversion to a low-frequency large-offset land data set. In *SEG Technical Program Expanded Abstracts 2010*, pp. 930–934. Society of Exploration Geophysicists.

Plessix, R.-E. and W. Mulder (2004). Frequency-domain finite-difference amplitude-preserving migration. *Geophysical Journal International 157*(3), 975–987.

Pole, A., M. West, and J. Harrison (1994). *Applied Bayesian forecasting and time series analysis*. Chapman and Hall/CRC.

Politis, D., J. Romano, and M. Wolf (1999). *Subsampling*. Springer.

Pratt, R. G. (1999). Seismic waveform inversion in the frequency domain, part 1: Theory and verification in a physical scale model. *Geophysics 64*(3), 888–901.

Pratt, R. G., C. Shin, and G. Hick (1998). Gauss–newton and full newton methods in frequency–space seismic waveform inversion. *Geophysical Journal International 133*(2), 341–362.

Rao, T. S. (1970). The fitting of non-stationary time-series models with time-dependent parameters. *Journal of the Royal Statistical Society. Series B (Methodological)*, 312–322.

Rao, Y. and Y. Wang (2017). Seismic waveform tomography with shot-encoding using a restarted l-bfgs algorithm. *Scientific Reports 7*(1), 8494.

Raskutti, G. and M. Mahoney (2015). Statistical and algorithmic perspectives on randomized sketching for ordinary least-squares. In *International Conference on Machine Learning*, pp. 617–625.

Raskutti, G. and M. W. Mahoney (2016). A statistical perspective on randomized sketching for ordinary least-squares. *The Journal of Machine Learning Research 17*(1), 7508–7538.

Rissanen, J. (2000). MDL denoising. *IEEE Transactions on Information Theory 46*(7), 2537–2543.

Sayed, A. H. et al. (2014). Adaptation, learning, and optimization over networks. *Foundations and Trends® in Machine Learning 7*(4-5), 311–801.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The Annals of statistics 6*(2), 461–464.

Seber, G. A. (2008). *A matrix handbook for statisticians*, Volume 15. John Wiley & Sons.

Shi, W., Q. Ling, K. Yuan, G. Wu, and W. Yin. On the linear convergence of the admm in decentralized consensus optimization.

Simon, I., N. Snavely, and S. M. Seitz (2007). Scene summarization for online image collections. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8. IEEE.

Solo, V. (1981). Strong consistency of least squares estimators in regression with correlated disturbances. *The Annals of Statistics*, 689–693.

Song, Z., L. F. Yang, and P. Zhong (2018). Sensitivity sampling over dynamic geometric data streams with applications to k-clustering. *arXiv preprint arXiv:1802.00459*.

Spielman, D. A. and N. Srivastava (2011). Graph sparsification by effective resistances. *SIAM Journal on Computing 40*(6), 1913–1926.

Sra, S., S. Nowozin, and S. J. Wright (2012). *Optimization for machine learning*. Mit Press.

Stakgold, I. and M. J. Holst (2011). *Green's functions and boundary value problems*, Volume 99. John Wiley & Sons.

Tarantola, A. (1984). Inversion of seismic reflection data in the acoustic approximation. *Geophysics 49*(8), 1259–1266.

The MathWorks, I. (2018). *System Identification Toolbox: User's Guide (r2018a)*.

Tiao, G. C. and R. S. Tsay (1989). Model specification in multivariate time series. *Journal of the Royal Statistical Society. Series B (Methodological)*, 157–213.

Tsay, R. S. (2013). *Multivariate Time Series Analysis: With R and Financial Applications*. John Wiley & Sons.

Virieux, J. (1986). P-sv wave propagation in heterogeneous media: Velocity-stress finite-difference method. *Geophysics 51*(4), 889–901.

Virieux, J. and S. Operto (2009). An overview of full-waveform inversion in exploration geophysics. *Geophysics 74*(6), WCC1–WCC26.

Wang, H., R. Zhu, and P. Ma (2017). Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association* (just-accepted).

West, M., P. J. Harrison, and H. S. Migon (1985). Dynamic generalized linear models and bayesian forecasting. *Journal of the American Statistical Association 80*(389), 73–83.

Wiggins, S. M. and J. A. Hildebrand (2007). High-frequency acoustic recording package (harp) for broad-band, long-term marine mammal monitoring. In *Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies, 2007. Symposium on*, pp. 551–557. IEEE.

Woodruff, D. P. (2014). Data streams and applications in computer science. *Bulletin of EATCS 3*(114).

Woodruff, D. P. et al. (2014). Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science 10*(1–2), 1–157.

Wright, S. and J. Nocedal (1999). Numerical optimization. *Springer Science 35*(67-68), 7.

Wu, J. and I. Stojmenovic (2004). Ad hoc networks. *Computer 37*(2), 29–31.

Wu, P.-Y. and M. D. Wang (2016). The selection of quantification pipelines for illumina rna-seq data using a subsampling approach. In *Biomedical and Health Informatics (BHI), 2016 IEEE-EMBS International Conference on*, pp. 78–81. IEEE.

Wu, T., K. Yuan, Q. Ling, W. Yin, and A. H. Sayed (2018). Decentralized consensus optimization with asynchrony and delays. *IEEE Transactions on Signal and Information Processing over Networks 4*(2), 293–307.

Xiao, F. and L. Wang (2008). Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays. *IEEE Transactions on Automatic Control 53*(8), 1804–1816.

Xu, P., J. Yang, and F. N. Roosta-Khorasani (2016). Sub-sampled newton methods with non-uniform sampling. In *Advances In Neural Information Processing Systems*, pp. 2530–2538.

Yilmaz, Ö. (2001). *Seismic Data Analysis: Processing, Inversion, and Interpretation of Seismic Data*. Society of exploration geophysicists.

Young, P. C. (2012). *Recursive estimation and time-series analysis: an introduction*. Springer Science & Business Media.

Yuan, K., Q. Ling, and W. Yin (2016). On the convergence of decentralized gradient descent. *SIAM Journal on Optimization 26*(3), 1835–1854.

Yun, S.-H., G. J. Tearney, J. F. de Boer, N. Iftimia, and B. E. Bouma (2003). High-speed optical frequency-domain imaging. *Optics express 11*(22), 2953–2963.

Zhang, K., C. Liu, J. Zhang, H. Xiong, E. Xing, and J. Ye (2017). Randomization or condensation?: Linear-cost matrix sketching via cascaded compression sampling. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 615–623. ACM.

Zhang, T., H.-C. Ho, M. Wendler, and W. B. Wu (2013). Block sampling under strong dependence. *Stochastic Processes and their Applications 123*(6), 2323–2339.

Zhang, X., R. Xie, and P. Ma (2018). *Statistical Leveraging Methods in Big Data*, pp. 51–74. Cham: Springer International Publishing.

Zhu, Y. and D. Shasha (2002). Statstream: Statistical monitoring of thousands of data streams in real time. In *Proceedings of the 28th International Conference on Very Large Data Bases*, pp. 358–369. VLDB Endowment.