5-10-2016

Online Updating Methods for Big Data Streams

Chun Wang
chun.wang@uconn.edu

Online Updating Methods for Big Data Streams

Chun Wang, Ph.D.
University of Connecticut, 2016

ABSTRACT

Big data are data on a massive scale in terms of volume, intensity, and complexity that exceed the capacity of standard analytic tools. They present opportunities as well as challenges to statisticians. This dissertation summarizes recent methodological and software developments in statistics that address the big data challenges at first and then presents statistical methods for big data arising from online analytical processing, where large amounts of data arrive in streams and require fast analysis without storage/access to the historical data, which is called online updating methods. In particular, iterative estimating algorithms and statistical inferences are developed for linear models and estimating equations that update as new data arrive. These algorithms are computationally efficient, minimally storage-intensive, and allow for possible rank deficiencies in the subset design matrices due to rare-event covariates. Goodness-of-fit tests, model diagnostics, and variable selection criteria are also developed under the same framework. When new variables become available, a method that utilizes the information from earlier data in the online updating algorithm with some corrections to reduce bias and improve efficiency is presented.

Online Updating Methods for Big Data Streams

Chun Wang

B.S., Mathematics, Nanjing University, Nanjing, China, 2011

M.S., Statistics, University of Connecticut, CT, USA, 2015

A Dissertation
Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy
at the
University of Connecticut

2016

APPROVAL PAGE

Doctor of Philosophy Dissertation

Online Updating Methods for Big Data Streams

Presented by

Chun Wang, B.S. Mathematics, M.S. Statistics

Co-Major Advisor _____

Jun Yan

Co-Major Advisor _____

Elizabeth D. Schifano

Associate Advisor _____

Ming-Hui Chen

University of Connecticut

2016

To my parents Yujun, Qi, and my wife Qianzhu.

Acknowledgements

Though only my name appears on the cover of this dissertation, a great many people have contributed to its production. I owe my gratitude to all people who have made this dissertation possible and because of whom my graduate experience has been one that I will cherish forever.

My deepest gratitude is to my two co-major advisors Dr. Jun Yan and Dr. Elizabeth Schifano for the continuous support of my Ph.D. study and research, for their patience, motivation, enthusiasm, and immense knowledge. Besides the research on dissertation, Dr. Yan brought me a chance to involve in a project with an insurance company which directly helped me get the first job offer and will have big influence in my career. Dr. Schifano is sweet-tempered and always nice to everyone. As a non-native speaker, I have got tremendous amount of help and suggestions on writing from her which not only makes the dissertation easy to read, but will continue benefiting me in the future.

I would like to thank Dr. Ming-Hui Chen for being my associate advisor in my dissertation committee, for initializing the Big Data research group where most of my research were accomplished, and for all the great advice that enriched and improved my dissertation. I am also thankful to the other two members in the Big Data research group, Jing Wu and Dr. Yuping Zhang, for the many valuable discussions that helped me understand my research area better. I'm also grateful to Dr. Joseph Glaz, Dr. Zhiyi

Chi, and all other professors, staff, and friends in the department who helped me stay sane and happy through the last five years. Without you, my journey here will be much tougher.

I am also indebted to Xiuwen and Xiuhong who have supported, encouraged, and guided me in the last ten years. I always consider myself extremely lucky to have them not only being mentors, but also as friends in life. Their broad view and positive attitude to the world will continue keeping me being self-motivated and full of passion in the future.

Last but not the least, none of this would have been possible without the love and patience of my family. My parents Yujun, Qi, and my wife Qianzhu whom this dissertation is dedicated to, have been a constant source of love, concern, support and strength all these years. I love you all!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Big Data in Statistics

A 2011 McKinsey report predicted shortage of talent necessary for organizations to take advantage of big data (Manyika et al., 2011). Data now stream from daily life thanks to technological advances, and big data has indeed become a big deal (e.g., Shaw, 2014). In the President's Corner of the June 2013 issue of AMStat News, the three presidents (elect, current, and past) of the American Statistical Association (ASA) wrote an article titled "The ASA and Big Data" (Schenker et al., 2013). In the followup July 2013 column, president Marie Davidian further raised the issues of statistics not being recognized as data science and mainstream academic statisticians being left behind by the rise of big data (Davidian, 2013). A white paper prepared by a working group of the ASA called for more ambitious efforts from statisticians to work together with researchers in other fields on national research priorities in order to achieve better science more quickly (Rudin et al., 2014). The same concern was expressed in a 2014 president's address of the Institute of Mathematical Statistics (IMS) (Yu, 2014). President Bin Yu of the IMS called for statisticians to own Data Science by working on real problems such as those

from genomics, neuroscience, astronomy, nanoscience, computational social science, personalized medicine/healthcare, finance, and government; relevant methodology/theory will follow naturally.

Big data in the media or the business world may mean differently than what are familiar to academic statisticians (Jordan and Lin, 2014). Big data are data on a massive scale in terms of volume, intensity, and complexity that exceed the ability of standard software tools to manage and analyze (e.g., Snijders et al., 2012). The origin of the term "big data" as it is understood today has been traced back in a recent study (Diebold, 2012) to lunch-table conversations at Silicon Graphics in the mid-1990s, in which John Mashey figured prominently (Mashey, 1998). Big data are generated by countless online interactions among people, transactions between people and systems, and sensor-enabled machinery. Internet search engines (e.g., Google and YouTube) and social network tools (e.g., Facebook and Twitter) generate billions of activity data per day. Rather than Gigabytes and Terabytes, nowadays, the data produced are estimated by zettabytes, and are growing 40% every day (Fan and Bifet, 2013). In the big data analytics world, a 3V definition by Laney (2001) is widely accepted: volume (amount of data), velocity (speed of data in and out), and variety (range of data types and sources). High variety brings nontraditional or even unstructured data types, such as social network sentiments and internet map usage, which calls for new, creative ways to understand the structure of data and even to ask intelligent research questions (e.g., Jordan and Lin, 2014). High volume and high velocity may bring noise accumulation, spurious correlation and incidental

homogeneity, creating issues in computational feasibility and algorithmic stability (Fan et al., 2014).

Notwithstanding that new statistical thinking and methods are needed for the high variety aspect of big data, our focus is on fitting standard statistical models to big data whose size exceeds the capacity of a single computer from its high volume and high velocity. There are two computational barriers for big data analysis: 1) the data can be too big to hold in a computer's memory; and 2) the computing task can take too long to wait for the results (Wang et al., 2016). These barriers can be approached with newly developed statistical methodologies and/or computational methodologies. Despite the impression that statisticians are left behind in media discussions or governmental summits on big data, some statisticians have made important contributions and are pushing the frontier. Sound statistical procedures that are scalable computationally to massive datasets have been proposed (Jordan, 2013). Examples are subsampling-based approaches (Kleiner et al., 2014; Ma et al., 2013; Liang et al., 2013; Maclaurin and Adams, 2014), divide and conquer approaches (Lin and Xi, 2011; Chen and Xie, 2014; Song and Liang, 2014; Neiswanger et al., 2013), and online updating approaches (Schifano et al., 2016). From a computational perspective, much effort has been put into the most active, open source statistical environment, R (R Core Team, 2014a). Statistician R developers are relentless in their drive to extend the reach of R into big data (Rickert, 2013). Recent UseR! conferences had many presentations that directly addressed big data, including a 2014 keynote lecture by John Chambers, the inventor

of the S language (Chambers, 2014). Most cutting edge methods are first and easily implemented in R. Given the open source nature of R and the active recent development, our focus on software for big data will be on R and R packages. Revolution R Enterprise (RRE) is a commercialized version of R, but it offers free academic use, so it is also included in our case study and benchmarked. Other commercial software such as SAS, SPSS, and MATLAB will be briefly touched upon for completeness.

1.2 Methods on Big Data Streams

The divide-and-conquer approach mentioned above is appealing because the data are first divided into subsets and then numeric and visualization methods are applied to each of the subsets separately. The approach culminates by aggregating the results from each subset to produce a final solution. Nevertheless, In some applications, data arrives in streams or in large chunks, and an online, sequentially updated analysis is desirable without storage requirements. Besides, most of the focus to date in the final aggregation step is in estimating the unknown quantity of interest, with little to no attention devoted to standard error estimation and inference. We firstly examined inference in the online-updating setting in a recently published paper Schifano et al. (2016). Even with big data, inference remains an important issue for statisticians, particularly in the presence of rare-event covariates. In this dissertation, standard error formulae

for divide-and-conquer estimators is provided in the linear model (LM) and estimating equation (EE) framework. We further develop iterative estimating algorithms and statistical inferences for the LM and EE frameworks for online-updating, which update as new data arrive. These algorithms are computationally efficient, minimally storage-intensive, and allow for possible rank deficiencies in the subset design matrices due to rare-event covariates. Within the online-updating setting for linear models, we propose tests for outlier detection based on predictive residuals and derive the exact distribution and the asymptotic distribution of the test statistics for the normal and non-normal cases, respectively. In addition, within the online-updating setting for estimating equations, we propose a new estimator and show that it is asymptotically consistent. We further establish new uniqueness results for the resulting cumulative EE estimators in the presence of rank-deficient subset design matrices. Our simulation study and real data analysis demonstrate that the proposed estimator outperforms other divide-and-conquer or online-updated estimators in terms of bias and mean squared error.

The naive online updating method works well when the variables of interest do not change over time. Nevertheless, in a typical regression setting, emergence of new variables is common, due to, for example, negligence in data collection in the past, change of protocol, or advances in information technology. In medical studies, new medical devices make possible monitoring measurements that have been unseen before (e.g., Hood et al., 2004). In financial analyses at the company level, new companies become public every month, releasing unprecedented information at the initial public

offering (e.g., Certo, 2003). In auto insurance, new devices gather data on drivers' behaviors, which could be exploited in pricing (e.g., Desyllas and Sako, 2013). In the airline data analysis, domestic carriers are required to report causes of flight delays to the Bureau of Transportation Statistics (BTS) since June 2003 (e.g., Rupp, 2007). In regression modeling for such situations with newly available covariates, the naive online updating analyses would start from the scratch, discarding the possibly useful information contained in the existing data and, hence, losing efficiency in statistical inferences.

In this dissertation, we also propose a novel extension of the online updating algorithm for the added variable situation. When new covariates become available, under that assumption that the true model contains the new variables, the previous cumulated estimates for the coefficients of existing variables are biased. To improve efficiency with the existing information, we correct the bias in the cumulative coefficient estimates and their variance estimates of the existing covariates; the online updating process resumes from the next block.

Because of no storage of previous data blocks, the bias correction is constructed using the block where the new variables emerge. The correction is the difference in the estimates of the coefficients of the existing covariates with and without the new covariates. This problem has been studied as comparing regression coefficients between nested models (Clogg et al., 1995; Allison, 1995; Yan et al., 2013), which is an important subject of mediation analysis with wide applications in social sciences. The recent asymptotic

validation in Yan et al. (2013) provides a unified, intuitive way to make inferences about the difference in the LM setting with clustered data. We extend this method to the GLM setting, and in the LM case, our derivation matches the closed-form expressions in Clogg et al. (1995) and Allison (1995). With this bias correction, we develop all the quantities that are needed in adjusting the cumulative coefficient estimates and variance estimates from previous data. We show that in a three-variable regression setting where y is the response variable, x is the existing variable, and z is the newly added variable, the variance of the bias-corrected online updating estimator is always smaller than the variance of the naive estimator from the changing block data only as long as the squared correlation coefficient between x and z is smaller than $1/2$. The efficiency gain remains under a mild condition when the squared correlation coefficient between x and z is greater than $1/2$. Further, more accumulated data in the past yields more efficiency gain from the bias-corrected online updating estimator over the naive estimator from the changing block data only.

The rest of the dissertation is organized as follows: Chapter 2 summarizes recent methodological and software developments in statistics that address the big data challenges. In Chapter 3, the online updating method for big data streams is presented. Iterative estimating algorithms and statistical inferences are developed for linear models and estimating equations, as well as the goodness-of-fit tests, model diagnostics, and variable selection criteria. Chapter 4 addresses a condition where online updating method no longer suffices that new variables become available. A brief summary of

results and proposal of future works for the dissertation is discussed in Chapter 5.

Chapter 2

Review on Methods and Software

2.1 Statistical Methods

The recent methodologies for big data can be loosely grouped into three categories: resampling-based, divide and conquer, and online updating. To put the different methods in a context, consider a dataset with n independent and identically distributed observations, where n is too big for standard statistical routines such as logistic regression.

2.1.1 Subsampling-Based Methods

Bag of Little Bootstraps

Kleiner et al. (2014) proposed the bag of little bootstraps (BLB) approach that provides both point estimates and quality measures such as variance or confidence intervals. It is a combination of subsampling (Politis et al., 1999), the m-out-of-n bootstrap (Bickel et al., 1997), and the bootstrap (Efron, 1979) to achieve computational efficiency. BLB consists of the following steps. First, draw s subsamples of size m from the original data

of size n. For each of the s subsets, draw r bootstrap samples of size n instead of m, and obtain the point estimates and their quality measures (e.g., confidence interval) from the r bootstrap samples. Then, the s bootstrap point estimates and quality measures are combined (e.g., by average) to yield the overall point estimates and quality measures. In summary, BLB has two nested procedures: the inner procedure applies the bootstrap to a subsample, and the outer procedure combines these multiple bootstrap estimates. The subsample size m was suggested to be n^γ with $\gamma \in [0.5, 1]$ (Kleiner et al., 2014), a much smaller number than n. Although the inner bootstrap procedure conceptually generates multiple resampled data of size n, what is really needed in the storage and computation is a sample of size m with a weight vector. In contrast to subsampling and the m-out-of-n bootstrap, there is no need for an analytic correction (e.g., $\sqrt{m/n}$) to rescale the confidence intervals from the final result. The BLB procedure facilitates distributed computing by letting each subsample of size m be processed by a separate processor. Kleiner et al. (2014) proved the consistency of BLB and provided high order correctness. Their simulation study showed good accuracy, convergence rate and remarkable computational efficiency.

Leveraging

Ma and Sun (2014) proposed to use leveraging to facilitate scientific discoveries from big data using limited computing resources. In a leveraging method, one samples a small proportion of the data with certain weights (subsample) from the full sample, and then

performs intended computations for the full sample using the small subsample as a surrogate. The key to success of the leveraging methods is to construct the weights, the nonuniform sampling probabilities, so that influential data points are sampled with high probabilities (Ma et al., 2013). Leveraging methods are different from the traditional subsampling or m-out-of-n bootstrap in that 1) they are used to achieve feasible computation even if the simple analytic results are available; 2) they enable visualization of the data when visualization of the full sample is impossible; and 3) they usually use unequal sampling probabilities for subsampling data. This approach is quite unique in allowing pervasive access to extract information from big data without resorting to high performance computing.

Mean Log-likelihood

Liang et al. (2013) proposed a resampling-based stochastic approximation approach with an application to big geostatistical data. The method uses Monte Carlo averages calculated from subsamples to approximate the quantities needed for the full data. Motivated from minimizing the Kullback–Leibler (KL) divergence, they approximate the KL divergence by averages calculated from subsamples. This leads to a maximum mean log-likelihood estimation method. The solution to the mean score equation is obtained from a stochastic approximation procedure, where at each iteration, the current estimate is updated based on a subsample of size m drawn from the full data. As m is much smaller than n, the method is scalable to big data. Liang et al. (2013) established the

consistency and asymptotic normality of the resulting estimator under mild conditions. In a simulation study, the convergence rate of the method was almost independent of n, the sample size of the full data.

Subsampling-Based MCMC

As a popular general purpose tool for Bayesian inference, Markov chain Monte Carlo (MCMC) for big data is challenging because of the prohibitive cost of likelihood evaluation of every datum at every iteration. Liang and Kim (2013) extended the mean log-likelihood method to a bootstrap Metropolis–Hastings (MH) algorithm in MCMC. The likelihood ratio of the proposal and current estimate in the MH ratio is replaced with an approximation from the mean log-likelihood based on k bootstrap samples of size m. The algorithm can be implemented by exploiting the embarrassingly parallel structure and avoids repeated scans of the full dataset in iterations. Maclaurin and Adams (2014) proposed an auxiliary variable MCMC algorithm called Firefly Monte Carlo (FlyMC) that only queries the likelihoods of a potentially small subset of the data at each iteration yet simulates from the exact posterior distribution. For each data point, a binary auxiliary variable and a strictly positive lower bound of the likelihood contribution are introduced. The binary variable for each datum effectively turn on and off data points in the posterior, hence the "firefly" name. The probability of turning on each datum depends on the ratio of its likelihood contribution and the introduced lower bound. The computational gain depends on that the lower bound is tight enough

and that simulation of the auxiliary variables is cheap enough. Because of the need to hold the whole data in computer memory, the size of the data this method can handle is limited.

The pseudo-marginal Metropolis–Hasting algorithm replaces the intractable target (posterior) density in the MH algorithm with an unbiased estimator (Andrieu and Roberts, 2009). The log-likelihood is estimated by an unbiased subsampled version, and an unbiased estimator of the likelihood is obtained by correcting the bias of the exponentiation of this estimator. Quiroz et al. (2014) proposed subsampling the data using probability proportional-to-size (PPS) sampling to obtain an approximately unbiased estimate of the likelihood which is used in the MH acceptance step. The subsampling approach was further improved in Quiroz et al. (2015) using the efficient and robust difference estimator from the survey sampling literature.

2.1.2 Divide and Conquer

A divide and conquer algorithm (which may appear under other names such as divide and recombine, split and conquer, or split and merge) generally has three steps: 1) partitions a big dataset into K blocks; 2) processes each block separately (possibly in parallel); and 3) aggregates the solutions from each block to form a final solution to the full data.

Aggregated Estimating Equations

For a linear regression model, the least squares estimator for the regression coefficient β for the full data can be expressed as a weighted average of the least squares estimator for each block with weight being the inverse of the estimated variance matrix. The success of this method for linear regression depends on the linearity of the estimating equations in β and that the estimating equation for the full data is a simple summation of that for all the blocks. For general nonlinear estimating equations, Lin and Xi (2011) proposed a linear approximation of the estimating equations with the Taylor expansion at the solution in each block, and, hence, reduce the nonlinear estimating equation to the linear case so that the solutions to all the blocks are combined by a weighted average. The weight of each block is the slope matrix of the estimating function at the solution in that block, which is the Fisher information or inverse of the variance matrix if the equations are score equations. Lin and Xi (2011) showed that, under certain technical conditions including $K = O(n^{\gamma})$ for some $\gamma \in (0, 1)$, the aggregated estimator has the same limit as the estimator from the full data.

Majority Voting

Chen and Xie (2014) consider a divide and conquer approach for generalized linear models (GLM) where both the sample size n and the number of covariates p are large, by incorporating variable selection via penalized regression into a subset processing step. More specifically, for p bounded or increasing to infinity slowly, (p_n not faster than $o(e^{n_k})$,

while model size may increase at a rate of $o(n_k)$), they propose to first randomly split the data of size n into K blocks (size $n_k = O(n/K)$). In step 2, penalized regression is applied to each block separately with a sparsity-inducing penalty function satisfying certain regularity conditions. This approach can lead to different variable selection among the blocks, as different blocks of data may result in penalized estimates with different non-zero regression coefficients. Thus, in step 3, the results from the K blocks are combined by majority vote to create a combined estimator. The implicit assumption is that real effects should be found persistently and therefore should be present even under perturbation by subsampling (e.g. Meinshausen and Buhlmann, 2010). The derivation of the combined estimator in step 3 stems from ideas for combining confidence distributions in meta-analysis (Singh et al., 2005; Xie et al., 2011), where one can think of the K blocks as K independent and separate analyses to be combined in a meta-analysis. The authors show under certain regularity conditions that their combined estimator in step 3 is model selection consistent, asymptotically equivalent to the penalized estimator that would result from using all of the data simultaneously, and achieves the oracle property when it is attainable for the penalized estimator from each block (see e.g., Fan and Lv, 2011). They additionally establish an upper bound for the expected number of incorrectly selected variables and a lower bound for the expected number of correctly selected variables.

Screening with Ultrahigh Dimension

Instead of dividing the data into blocks of observations in step 1, Song and Liang (2014) proposed a split-and-merge (SAM) method that divides the data into subsets of covariates for variable selection in ultrahigh dimensional regression from the Bayesian perspective. This method is particularly suited for big data where the number of covariates P_n is much larger than the sample size n, $P_n \gg n$, and possibly increasing with n. In step 2, Bayesian variable selection is separately performed on each lower dimensional subset, which facilitates parallel processing. In step 3, the selected variables from each subset are aggregated, and Bayesian variable selection is applied on the aggregated data. The embarrassingly parallel structure in step 2 makes the SAM method applicable to big data problems with millions or more predictors. Posterior consistency is established for correctly specified models and for misspecified models, the latter of which is necessary because it is quite likely that some true predictors are missing. With correct model specification, true covariates will be identified as the sample size becomes large; under misspecified models, all predictors correlated with the response variable will be identified. Compared with the sure independence screening (SIS) approach (Fan and Lv, 2008), the method uses the joint information of multiple predictors in predictor screening while SIS only uses the marginal information of each predictor. Their numerical results show that the SAM approach outperforms competing methods for ultrahigh dimensional regression.

Parallel MCMC

In the Bayesian framework, it is natural to partition the data into K subsets and run parallel MCMC on each one of them. The prior distribution for each subset is often obtained by taking a power $1/K$ of the prior distribution for whole data in order to preserve the total amount of prior information (which may change the impropriety of the prior). MCMC is run independently on each subset with no communications between subsets (and, thus, embarrassingly parallel), and the resulting samples are combined to approximate samples from the full data posterior distribution. Neiswanger et al. (2013) proposed to use kernel density estimators of the posterior density for each data subset, and estimate the full data posterior by multiplying the subset posterior densities together. This method is asymptotically exact in the sense of being converging in the number of MCMC iterations. Wang et al. (2015) replaced the kernel estimator of Neiswanger et al. (2013) with a random partition tree histogram, which uses the same block partition across all terms in the product representation of the posterior to control the number of terms in the approximation such that it does not explode with m. Scott et al. (2013) proposed a consensus Monte Carlo algorithm, which produces the approximated full data posterior using weighted averages over the subset MCMC samples. The weight used (for Gaussian models) for each subset is the inverse of the variance-covariance matrix of the MCMC samples. The method is effective when the posterior is close to Gaussian but may cause bias when the distribution is skewed or has multi-modes. The consensus Monte Carlo principal is approached from a variational

perspective by Rabinovich et al. (2015). The embarrassingly parallel feature of these methods facilitates their implementation in the MapReduce framework that exploits the division and recombination strategy (Dean and Ghemawat, 2008). The final recombination step is implemented in R package `parallelMCMCcombine` (Miroshnikov and Conlon, 2014).

Going beyond embarrassingly parallel MCMC remains challenging because of storage issues and communication overheads. General strategies for parallel MCMC such as multiple-proposal MH algorithm (Calderhead, 2014) and population MCMC (Song et al., 2014) mostly require full data at each node.

2.1.3 Online Updating for Stream Data

In some applications, data come in streams or large chunks, and a sequentially updated analysis is desirable without storing the data. Motivated from a Bayesian inference perspective, Schifano et al. (2016) extends the work of Lin and Xi (2011) in a few important ways. First, they introduce divide-and-conquer-type variance estimates of regression parameters in the linear model and estimating equation settings. These estimates of variability allow for users to make inferences about the true regression parameters based upon the previously developed divide-and-conquer point estimates of the regression parameters. Second, they develop iterative estimating algorithms and statistical inferences for linear models and estimating equations that update as new data arrive. Thus, while

the divide-and-conquer setting is quite amenable to parallel processing for each subset, the online-updating approach for data streams is inherently sequential in nature. Their algorithms were designed to be computationally efficient and minimally storage-intensive, as they assume no access/storage of the historical data. Third, the authors address the issue of possible rank deficiencies when dealing with blocks of data, and the uniqueness properties of the combined and cumulative estimators when using a generalized inverse. The authors also provide methods for assessing goodness of fit in the linear model setting, as standard residual-based diagnostics cannot be performed with the cumulative data without access to historical data. Instead, they propose outlier tests relying on predictive residuals, which are based on the predictive values computed from the cumulative estimate of the regression coefficients attained at the previous accumulation point. Additionally, they introduce a new online-updated estimator of the regression coefficients and corresponding estimator of the standard error in the estimating equation setting that takes advantage of information from the previous data. They show theoretically that this new estimator, the cumulative updated estimating equation (CUEE) estimator, is asymptotically consistent, and show empirically that the CUEE estimator is less biased in their finite sample simulations than the cumulatively estimated version of the estimator of Lin and Xi (2011).

2.2 Open Source R and R Packages

Handling big data is one of the topics of high performance computing. As the most popular open source statistical software, R and its add-on packages provide a wide range of high performance computing; see Comprehensive R Archive Network (CRAN) task view on "High-Performance and Parallel Computing with R" (Eddelbuettel, 2014). The focus of this section is on how to break the computer memory barrier and the computing power barrier in the context of big data.

2.2.1 Breaking the Memory Barrier

The size of big data is relative to the available computing resources. The theoretical limit of random access memory (RAM) is determined by the width of memory addresses: 4 gigabyte (GB) (2^{32} bytes) for a 32-bit computer and 16.8 million terabyte (2^{64} bytes) for a 64-bit computer. In practice, however, the latter is limited by the physical space of a computer case, the operating system, and specific software. Individual objects in R have limits in size too; an R user can hardly work with any object of size close to that limit. Emerson and Kane (2012) suggested that a data set would be considered *large* if it exceeds 20% of RAM on a given machine and *massive* if it exceeds 50%, in which case, even the simplest calculation would consume all the remaining RAM.

Memory boundary can be broken with an external memory algorithms (EMA) (e.g., Vitter, 2001), which conceptually works by storing the data on a disk storage (which

has a much greater limit than RAM), and processing one chunk of it at a time in RAM (e.g., Lumley, 2013). The results from each chunk will be saved or updated and the process continues until the entire dataset is exhausted; then, if needed as in an iterative algorithm, the process is reset from the beginning of the data. To implement an EMA for each statistical function, one need to address 1) data management and 2) numerical calculation.

Data Management

Earlier solutions to oversize data resorted to relational databases. This method depends on an external database management system (DBMS) such as MySQL, PostgreSQL, SQLite, H2, ODBC, Oracle, and others. Interfaces to R are provided through many R packages such as **sqldf** (Grothendieck, 2014), **DBI** (R Special Interest Group on Databases, 2014), **RSQLite** (Wickham et al., 2014), and others. The database approach requires a DBMS to be installed and maintained, and knowledge of structured query language (SQL); an exception for simpler applications is package **filehash** (Peng, 2006), which comes with a simple key-value database implementation itself. The numerical functionality of SQL is quite limited, and calculations for most statistical analyses require copying subsets of the data into objects in R facilitated by the interfaces. Extracting chunks from an external DBMS is computationally much less efficient than the more recent approaches discussed below (Kane et al., 2013).

Two R packages, **bigmemory** (Kane et al., 2013) and **ff** (Adler et al., 2014) provide

data structures for massive data while retaining a look and feel of R objects. Package **bigmemory** defines a data structure `big.matrix` for numeric matrices which uses memory-mapped files to allow matrices to exceed the RAM size on computers with 64-bit operating systems. The underling technology is memory mapping on modern operating systems that associates a segment of virtual memory in a one-to-one correspondence with contents of a file. These files are accessed at a much faster speed than in the database approaches because operations are handled at the operating-system level. The `big.matrix` structure has several advantages such as support of shared memory for efficiency in parallel computing, reference behavior that avoids unnecessary temporary copies of massive objects, and column-major format that is compatible with legacy linear algebra packages (e.g., BLAS, LAPACK) (Kane et al., 2013). The package provides utility to read in a csv file to form a `big.matrix` object, but it only allows one type of data, numeric; it is a numeric matrix after all.

Package **ff** provides data structures that are stored in binary flat files but behave (almost) as if they were in RAM by transparently mapping only a section (pagesize) of meta data in main memory. Unlike **bigmemory**, it supports R's standard atomic data types (e.g., double or logical) as well as nonstandard, storage efficient atomic types (e.g., the 2-bit unsigned `quad` type allows efficient storage of genomic data as a factor with levels A, T, G, and C). It also provides class `ffdf` which is like `data.frame` in R, and import/export filters for csv files. A binary flat file can be shared by multiple `ff` objects in the same or multiple R processes for parallel access. Utility functions allow interactive

process of selections of big data.

Numerical Calculation

The data management systems in packages **bigmemory** or **ff** do not mean that one can apply existing R functions yet. Even a simple statistical analysis such as linear model or survival analysis will need to be implemented for the new data structures. Chunks of big data will be processed in RAM one at a time, and often, the process needs to be iterated over the whole data. A special case is the linear model fitting, where one pass of the data is sufficient and no resetting from the beginning is needed. Consider a regression model $E[Y] = X\beta$ with $n \times 1$ response Y, $n \times p$ model matrix X and $p \times 1$ coefficient β. The base R implementation `lm.fit` takes $O(np + p^2)$ memory, which can be reduced dramatically by processing in chunks. The first option is to compute $X'X$ and $X'y$ in increments, and get the least squares estimate of β, $\hat{\beta} = (X'X)^{-1}X'Y$. This method is adopted in package **speedglm** (Enea, 2014). A slower but more accurate option is to compute the incremental QR decomposition (Miller, 1992) of $X = QR$ to get R and $Q'Y$, and then solve β from $R\beta = Q'Y$. This option is implemented in package **biglm** (Lumley, 2013). Function `biglm` uses only p^2 memory of p variables and the fitted object can be updated with more data using `update`. The package also provides an incremental computation of sandwich variance estimator by accumulating a $(p+1)^2 \times (p+1)^2$ matrix of products of X and Y without a second pass of the data.

In general, a numerical calculation needs an iterative algorithm in computation and,

hence, multiple passes of the data are necessary. For example, a GLM fitting is often obtained through the iterated reweighted least squares (IRLS) algorithm. The `bigglm` function in package **biglm** implements the generic IRLS algorithm that can be applied to any specific data management system such as DBMS, **bigmemory**, or **ff**, provided that a function `data(reset = FALSE)` supplies the next chunk of data or zero-row data if there is no more, and `data(reset = TRUE)` resets to the beginning of the data for the next iteration. Specific implementation of the `data` function for object of class `big.matrix` and `ffdf` are provided in package **biganalytics** (Emerson and Kane, 2013a) and **ffbase** (Jonge et al., 2014), respectively.

For any statistical analysis on big data making use of the data management system, one would need to implement the necessary numerical calculations like what package **biglm** does for GLM. The family of **bigmemory** provides a collection of functions for `big.matrix` objects: **biganalytics** for basic analytic and statistical functions, **bigtabulate** for tabulation operations (Emerson and Kane, 2013b), and **bigalgebra** for matrix operation with the BLAS and LAPACK libraries (Kane et al., 2014). Some additional functions for `big.matrix` objects are available from other contributed packages, such as **bigpca** for principal component analysis and single-value decomposition (Cooper, 2014), and **bigrf** for random forest (Lim et al., 2014). For `ff` objects, package **ffbase** provides basic statistical functions (Jonge et al., 2014). Additional functions for `ff` objects are provided in other packages, with examples including **biglars** for least angle regression and LASSO (Seligman et al., 2011) and **PopGenome** for population genetic and genomic

analysis (Pfeifer et al., 2014).

If some statistical analysis, such as generalized estimating equations or Cox proportional hazards model, has not been implemented for big data, then one will need to modify the existing algorithm to implement it. As pointed out by Kane et al. (2013, p.5), this would open Pandora's box of recoding which is not a long-term solution for scalable statistical analyses; this calls for redesign of the next-generation statistical programming environment which could provide seamless scalability through file-backed memory-mapping for big data, help avoid the need for specialized tools for big data management, and allow statisticians and developers to focus on new methods and algorithms.

2.2.2 Breaking the Computing Power Barrier

Speeding Up

As a high level interpreted language, for which most of instructions are executed directly, R is infamously slow with loops. Some loops can be avoided by taking advantage of the vectorized functions in R or by clever vectorizing with some effort. When vectorization is not straightforward or loops are unavoidable, as in the case of MCMC, acceleration is much desired, especially for big data. The least expensive tool in a programmer's effort to speed up R code is to compile them to byte code with the **compiler** package, which was developed by Luke Tierney and is now part of base R. The byte code compiler translates the high-level R into a very simple language that can be interpreted by a

very fast byte code interpreter, or virtual machine. Starting with R 2.14.0 in 2011, the base and recommended packages were pre-compiled into byte-code by default. Users' functions, expressions, scripts, and packages can be compiled for an immediate boost in speed by a factor of 2 to 5.

Computing bottlenecks can be implemented in a compiled language such as C/C++ or FORTRAN and interfaced to R through R's foreign language interfaces (R Core Team, 2014b, ch.5). Typical bottlenecks are loops, recursions, and complex data structures. Recent developments have made the interfacing with C++ much easier than it used to be (Eddelbuettel, 2013). Package **inline** (Sklyar et al., 2013) provides functions that wrap C/C++ (or FORTRAN) code as strings in R and takes care of compiling, linking, and loading by placing the resulting dynamically-loadable object code in the per-session temporary directory used by R. For more general usage, package **Rcpp** (Eddelbuettel et al., 2011) provides C++ classes for many basic R data types, which allow straightforward passing of data in both directions. Package **RcpEigen** (Bates et al., 2014) provides access to the high-performance linear algebra library Eigen for a wide variety of matrix methods, various decompositions and support of sparse matrices. Package **RcppArmadillo** (Eddelbuettel and Sanderson, 2014) connects R with Armadillo, a powerful templated linear algebra library which provides a good balance between speed and ease of use. Package **RInside** (Eddelbuettel and Francois, 2014) gives easy access of R objects from C++ by wrapping the existing R embedding application programming interface (API) in C++ classes. The **Rcpp** project has revolutionized the integration of R with C++; it

is now used by hundreds of R packages.

Diagnostic tools can help identify the bottlenecks in R code. Package **microbenchmark** (Mersmann, 2014) provides very precise timings for small pieces of source code, making it possible to compare operations that only take a tiny amount of time. For a collection of code, run-time of each individual operation can be measured with realistic inputs; the process is known as profiling. Function `Rprof` in R does the profiling, but the outputs are not intuitive to understand for many users. Packages **proftools** (Tierney and Jarjour, 2013) and **aprof** (Visser, 2014) provide tools to analyze profiling outputs. Packages **profr** (Wickham, 2014b), **lineprof** (Wickham, 2014c), and **GUIProfiler** (de Villar and Rubio, 2014) provide visualization of profiling results.

Scaling Up

The R package system has long embraced integration of parallel computing of various technologies to address the big data challenges. For embarrassingly parallelizable jobs such as bootstrap or simulation, where there is no dependency or communication between parallel tasks, many options are available with computer clusters or multicores. Schmidberger et al. (2009) reviewed the then state-of-the-art parallel computing with R, highlighting two packages for cluster use: **Rmpi** (Yu, 2002) which provides an R interface to the Message Passing Interface (MPI) in parallel computing; **snow** (Rossini et al., 2007) which provides an abstract layer with the communication details hidden from the end users. Since then, some packages have been developed and some discontinued. Packages

snowFT (Sevcikova and Rossini, 2012a) and **snowfall** (Knaus, 2013) extend **snow** with fault tolerance and wrappers for easier development of parallel R programs. Package **multicore** (Urbanek, 2014) provides parallel processing of R code on machines with multiple cores or CPUs. Its work and some of **snow** have been incorporated into the base R package **parallel**, which was first included in R 2.14.0 in 2011. Package **foreach** (Revolution Analytics and Weston, 2014) allows general iteration over elements in a collection without any explicit loop counter. Using `foreach` loop without side effects facilitates executing the loop in parallel with different parallel mechanisms, including those provided by **parallel**, **Rmpi**, and **snow**. For massive data that exceed the computer memory, a combination of **foreach** and **bigmemory**, with shared-memory data structure referenced by multiple processes, provides a framework with ease of development and efficiency of execution (both in speed and memory) as illustrated by Kane et al. (2013). Package **Rdsm** provides facilities for distributed shared memory parallelism at the R level, and combined with **bigmemory**, it enables parallel processing on massive, out-of-core matrices.

The "Programming with Big Data in R" project (pbdR) enables high-level distributed data parallelism in R with easy utilization of large clusters with thousands of cores (Ostrouchov et al., 2012). Big data are interpreted quite literally to mean that a dataset requires parallel processing either because it does not fit in the memory of a single machine or because its processing time needs to be made tolerable. The project focuses on

distributed memory systems where data are distributed across processors and communications between processors are based on MPI. It consists of a collection of R packages in a hierarchy. Package **pbdMPI** provides `S4` classes to directly interface with MPI to support the Single Program Multiple Data (SPMD) parallelism. Package **pbdSLAP** serves as a mechanism to utilize a subset of functions of scalable dense linear algebra in ScaLAPACK (Blackford et al., 1997), a subset of LAPACK routines redesigned with the SPMD style. Package **pbdBASE** contains a set of wrappers of low level functions in ScaLAPACK, upon which package **pbdMAT** builds to provide distributed dense matrix computing while preserving the friendly and familiar R syntax for these computations. Demonstrations on how to use these and other packages from the pbdR are available in package **pbdDEMO**.

A recent, widely adopted open source framework for massive data storage and distributed computing is Hadoop (The Apache Software Foundation, 2014b). Its heart is an implementation of the MapReduce programming model first developed at Google (Dean and Ghemawat, 2008), which divides the data to distributed systems and computes for each group (the map step), and then recombines the results (the reduce step). It provides fault tolerant and scalable storage of massive datasets across machines in a cluster (White, 2011). The model suits perfectly the embarrassingly parallelizable jobs and the distributed file system helps break the memory boundary. McCallum and Weston (2011, ch.5–8) demonstrated three ways to combine Hadoop and R. The first is to submit R scripts directly to a Hadoop cluster, which gives the user the most control and

the most power, but comes at the cost of a Hadoop learning curve. The second is a pure R solution via package **Rhipe**, which hides the communications to Hadoop from R users. The package (not on CRAN) is from the RHIPE project, which stands for R and Hadoop Integrated Programming Environment (Guha et al., 2012). With **Rhipe**, data analysts only need to write R code for the map step and the reduce step (Guha et al., 2012), and get the power of Hadoop without leaving R. The third approach targets specifically the Elastic MapReduce (EMR) at Amazon by a CRAN package **segue** (Long, 2012), which makes EMR as easy to use as a parallel backend for `lapply`-style operations. An alternative open source project that connects R and Hadoop is the RHadoop project, which is actively being developed by Revolution Analytics (Revolution Analytics, 2014). This project is a collection of R packages that allow users to manage and analyze data with Hadoop: **rhbase** provides functions for database management for the HBase distributed database, **rhdfs** provides functions for Hadoop distributed file system (HDFS), **rmr** provides functions to Hadoop MapReduce functionality, **plymr** provides higher level data processing for structured data, and the most recent addition **ravro** provides reading and writing functions for files in `avro` format, an efficient data serialization system developed at Apache (The Apache Software Foundation, 2014a).

Spark is a more recent, cousin project of Hadoop that supports tools for big data related tasks (The Apache Software Foundation, 2014c). The functions of Spark and Hadoop are neither exactly the same nor mutually exclusive, and they often work together. Hadoop has its own distributed storage system, which is fundamental for any big

data computing framework, allowing vast datasets to be stored across the hard drives of a scalable computer cluster rather than on a huge costly hold-it-all device. It persists back to the disk after a map or reduce action. In contrast, Spark does not have its own distributed file system, and it processes data in-memory (Zaharia et al., 2010). The biggest difference is disk-based computing versus memory-based computing. This is why Spark could work 100 times faster than Hadoop for some applications when the data fit in the memory. Some applications such as machine learning or stream processing where data are repeatedly queried makes Spark an ideal framework. For big data that does not fit in memory, Spark's operators spill data to disk, allowing it to run well on any sized data. For this purpose, it can be installed on top of Hadoop to take advantage of Hadoop's HDFS. An R frontend to Spark is provided in R package **SparkR** (Venkataraman, 2013), which has become part of Apache Spark recently. By using Spark's distributed computation engine, the package allows users to run large scale data analysis such as selection, filtering, aggregation from R. Karau et al. (2015) provides a summary of the state-of-the-art on using Spark.

As multicores have become the standard setup for computers today, it is desirable to automatically make use of the cores in implicit parallelism without any explicit requests from the user. The experimental packages **pnmath** and **pnmath0** by Luke Tierney replace a number of internal vector operations in R with alternatives that can take advantage of multicores (Tierney, 2009). For a serial algorithm such as MCMC, it is desirable to parallelize the computation bottleneck if possible, but this generally involves learning

new computing tools and the debugging can be challenging. For instance, Yan et al. (2007) used the parallel linear algebra package (PLAPACK) (van de Geijn, 1997) for the matrix operations (especially the Cholesky decomposition) in a MCMC algorithm for Bayesian spatiotemporal geostatistical models, but the scalability was only moderate.

When random numbers are involved as in the case of simulation, extra care is needed to make sure the parallelized jobs run independent (and preferably reproducible) random-number streams. Package **rsprng** (Li, 2010) provides an interface to the Scalable Parallel Random Number Generators (SPRNG) (Mascagni and Srinivasan, 2000). Package **rlecuyer** (Sevcikova and Rossini, 2012b) provides an interface to the random number generator with multiple independent streams developed by L'Ecuyer et al. (2002), the ideas of which are also implemented in the base package **parallel**: make independent streams by separating a single stream with a sufficiently large number of steps apart. Package **doRNG** (Gaujoux, 2014) provides functions to perform reproducible parallel `foreach` loops, independent of the parallel environment and associated `foreach` backend.

From a hardware perspective, many computers have mini clusters of graphics processing units (GPUs) that can help with bottlenecks. GPUs are dedicated numerical processors that were originally designed for rendering three dimensional computer graphics. A GPU has hundreds of processor cores on a single chip and can be programmed to apply the same numerical operations on large data array. Suchard et al. (2010) investigated the use of GPUs in massively parallel massive mixture modeling, and showed

better performance of GPUs than multicore CPUs, especially for larger samples. To reap the advantage, however, one needs to learn the related tools such as Compute Unified Device Architecture (CUDA), Open Computing Language (OpenCL), and so on, which may be prohibitive. An R package **gputools** (Buckner et al., 2013) provides an interface to NVidia CUDA toolkit and others.

If one is willing to step out of the comfort zone of R and take full control/responsibility of parallel computing, one may program with open source MPI or Open Multi-Processing (OpenMP). MPI is a language-independent communication system designed for programming on parallel computers, targeting high performance, scalability and portability (Pacheco, 1997). Most MPI implementations are available as libraries from C/C++, FORTRAN, and any language that can interface with such libraries, including C#, Java or Python. The interface from R can be accessed with package **Rmpi** (Yu, 2002) as mentioned earlier. Freely available implementations include OpenMPI (not OpenMP) and MPICH, while others come with license such as Intel MPI. OpenMP is an API that supports multi-platform shared memory multiprocessing programming in C/C++ and FORTRAN on most processor architectures and operating systems (Chapman et al., 2008). It is an add on to compilers (e.g., gcc, intel compiler) to take advantage of of shared memory systems such as multicore computers where processors share the main memory. MPI targets both distributed as well as shared memory systems while OpenMP targets only shared memory systems. MPI provides both process and thread based approach while OpenMP provides only thread based parallelism. OpenMP uses a portable,

scalable model that gives programmers a simple and flexible interface for writing multi-threaded programs in C/C++ and FORTRAN (Dagum and Enon, 1998). Debugging parallel programs can be very challenging.

2.3 Commercial Statistical Software

RRE is the core product of Revolution Analytics (formerly Revolution Computing), a company that provides R tools, support, and training. RRE focuses on big data, large scale multiprocessor (or high performance) computing, and multicore functionality. Massive datasets are handled via EMA and parallel EMA (PEMA) when multiprocessors or multicores are available. The commercial package **RevoScaleR** (Revolution Analytics, 2013) breaks the memory boundary by a special `XDF` data format that allows efficient storage and retrieval of data. Functions in the package (e.g., `rxGlm` for GLM fitting) know to work on a massive dataset one chunk at a time. The computing power boundary is also addressed — functions in the package can exploit multicores or computer clusters. Packages from the aforementioned open source project RHadoop developed by the company provide support for Hadoop. Other components in RRE allow high speed connection for various types of data sources and threading and inter-process communication for parallel and distributed computing. The same code works on small and big data, and on workstations, servers, clusters, Hadoop, or in the cloud. The single workstation version of RRE is free for academic use currently, and was used in the case

study in Section 2.4.

SAS, one of the most widely used commercial software for statistical analysis, provides big data support through SAS High Performance Analytics. Massive datasets are approached by grid computing, in-database processing, in-memory analytics and connection to Hadoop. The SAS High Performance Analytics Products include statistics, econometrics, optimization, forecasting, data mining, and text mining, which, respectively, correspond to SAS products STAS, ETS, OR, high-performance forecasting, enterprise miner, and text miner (Cohen and Rodriguez, 2013).

IBM SPSS, the Statistical Product and Services Solution, provides big data analytics through SPSS Modeler, SPSS Analytic Server, SPSS Collaboration and Deployment Services, and SPSS Analytic Catalyst (IBM, 2014). SPSS Analytic Server is the foundation and it focuses on high performance analytics for data stored in Hadoop-based distributed systems.

SPSS modeler is the high-performance data mining workbench, utilizing SPSS Analytic Server to leverage big data in Hadoop environments. Analysts can define analysis in a familiar and accessible workbench to conduct analysis modeling and scoring over high volumes of varied data. SPSS Collaboration and Deployment Services helps manage analytical assets, automate processes and efficiently share results widely and securely. SPSS Analytic Catalyst is the automation of analysis that makes analytics and data more accessible to users.

MATLAB provides a number of tools to tackle the challenges of big data analytics

(The MathWorks, Inc., 2014). Memory mapped variables map a file or a proportion of a file to a variable in RAM; disk variables direct access to variables from files on disk; datastore allows access to data that do not fit into RAM. Their combination addresses the memory boundary. The computation power boundary is broken by intrinsic multicore math, GPU computing, parallel computing, cloud computing, and Hadoop support.

2.4 A Case Study

The airline on-time performance data from the 2009 ASA Data Expo (`http://www.jstatsoft.org/index.php/jss/article/downloadSuppFile/v055i14/Airline.tar.bz2`) is used as a case study to demonstrate a logistic model fitting with a massive dataset that exceeds the RAM of a single computer. The data is publicly available and has been used for demonstration with big data by Kane et al. (2013) and others. It consists of flight arrival and departure details for all commercial flights within the USA, from October 1987 to April 2008. About 120 million flights were recorded with 29 variables. A compressed version of the pre-processed data set from the bigmemory project (`http://data.jstatsoft.org/v55/i14/Airline.tar.bz2`) is approximately 1.7GB, and it takes 12GB when uncompressed.

The response of the logistic regression is late arrival which was set to 1 if a flight was late by more than 15 minutes and 0 otherwise. Two binary covariates were created from the departure time: night (1 if departure occurred between 8pm and 5am and 0

Table 1: Timing results (in seconds) for reading in the whole 12GB data, transforming to create new variables, and fitting the logistic regression with three methods: **bigmemory**, **ff**, and RRE.

	Reading	Transforming	Fitting
bigmemory	968.6	105.5	1501.7
ff	1111.3	528.4	1988.0
RRE	851.7	107.5	189.4

otherwise) and weekend (1 if departure occurred on weekends and 0 otherwise). Two continuous covariates were included: departure hour (DepHour, range 0 to 24) and distance from origin to destination (in 1000 miles). In the raw data, the departure time was an integer of the HHmm format. It was converted to minutes first to prepare for DepHour. Three methods are considered in the case study: 1) combination of `bigglm` with package **bigmemory**; 2) combination of `bigglm` with package **ff**; and 3) the academic, single workstation version of RRE. The default settings of **ff** were used. Before fitting the logistic regression, the 12GB raw data needs to be read in from the csv format, and new variables need to be generated. This leads to a total of $120,748,239$ observations with no missing data. The R scripts for the three methods are in the supplementary materials for interested readers.

The R scripts were executed in batch mode on an 8-core machine running CenOS (a free Linux distribution functionally compatible with Red Hat Enterprise Linux which is officially supported by RRE), with Intel Core i7 2.93GHz CPU, and 16GB memory. Table 1 summarizes the timing results of reading in the whole 12GB data, transforming to create new variables, and fitting the logistic regression with the three methods. The

Table 2: Logistic regression results for late arrival.

	Estimate	Std. Error ($\times 10^4$)
(Intercept)	−2.985	9.470
DepHour	0.104	0.601
Distance	0.235	4.032
Night	−0.448	8.173
Weekend	−0.177	5.412

chunk sizes were set to be 500,000 observations for all three methods. For RRE, this was set when reading in the data to the `XDF` format; for the other two methods, this was set at the fitting stage using function `bigglm`. Under the current settings, RRE has a clear advantage in fitting with only 8% of the time used by the other two approaches. This is a result of the joint force of its using all 8 cores implicitly and efficient storage and retrieval of the data; the `XDF` version of the data is about 1/10 of the size of the external files saved by **bigmemory** or **ff**. Using **bigmemory** and using **ff** in `bigglm` had very similar performance in fitting the logistic regression, but the former took less time in reading, and significantly less time (only about 1/5) in transforming variables of the latter. The **bigmemory** method was quite close to the RRE method in the reading and the transforming tasks. The **ff** method took longer in reading and transforming than the **bigmemory** method, possibly because it used much less memory.

The results of the logistic regression are identical from all methods, and are summarized in Table 2. Flights with later departure hour or longer distance are more likely to be delayed. Night flights or weekend flights are less likely to be delayed. Given the huge sample size, all coefficients were highly significant. It is possible, however, that

Table 3: Time results (in seconds) for parallel computing quantiles of departure delay for each day of the week with 1 to 8 cores using **foreach**.

	1	2	3	4	5	6	7	8
bigmemory	22.1	11.2	7.8	6.9	6.2	6.3	6.4	6.8
ff	21.4	11.0	7.1	6.7	5.8	5.9	6.1	6.8

p-values can still be useful. A binary covariate with very low rate of event may still have an estimated coefficient with a not-so-low p-value (Schifano et al., 2016), an effect only estimable with big data.

As an illustration of **foreach** for embarrassingly parallel computing, the example in Kane et al. (2013) is expanded to include both **bigmemory** and **ff**. The task is to find three quantiles (0.5, 0.9, and 0.99) of departure delays for each day of the week; that is, 7 independent jobs can run on 7 cores separately. To make the task bigger, each job was set to run twice. The resulting 14 jobs were parallelized with `foreach` on the same Linux machine using 1 to 8 cores for the sake of illustration. The R script is included in the supplementary materials. The timing results are summarized in Table 3. There is little difference between the two implementations. When there is no communication overhead, with 14 jobs one would expect the run time to reduce to 1/2, 5/14, 4/14, 3/14, 3/14, 2/14, and 2/14, respectively, with 2, 3, 4, 5, 6, 7 and 8 cores. The impact of communication cost is obvious in Table 3. The time reduction is only closer to the expectation in the ideal case when the number of cores is smaller.

Chapter 3

Online Updating Algorithm

3.1 Introduction

As briefly introduced in Section 2.1.3, in certain applications, data come in streams or large chunks, and a sequentially updated analysis is desirable without storing the data. In this chapter, we present the online updating method for big data streams. When data arrives, only some small dimensional vectors and matrices need to be saved and updated. At any time point, coefficients and variance estimators can be constructed from these update-to-date vectors and matrices, as well as the statistical inferences, model diagnostics, and variable selection criteria. This feature makes the online updating method computationally efficient and minimum storage-intensive. Nevertheless, it is possible that some variables are not observable or homogeneous in certain blocks which makes the design matrix not of full rank and causes the rank-deficiency problem in the updating. This is more common when rare-event covariates are involved which are often of more interest than normal covariates in the big data setting. Online updating method allows rank deficiencies in the subset design matrices by utilizing generalized inverse method.

In this chapter, Section 3.2 focuses on the normal linear regression where notation and preliminaries are introduced in Section 3.2.1, the full online updating algorithm and inferences are presented in Section 3.2.2, and predictive residual diagnostic tests are developed in Section 3.2.3. Section 3.3 extends the online updating algorithm and estimators to the estimating equations. Rank deficiency problems for both linear models and estimating equations are discussed in Section 3.4 and variable section criteria is given in Section 3.5. Section 3.6 contains the numerical simulation results for both linear model and estimating equation settings, and Section 3.7 contains results from the analysis of real data regarding airline on-time statistics.

3.2 Normal Linear Regression

3.2.1 Notation and Preliminaries

Since the research is motivated by the work from Lin and Xi (2011), we present the notations for a fixed amount of data first to make connections with the aggregated estimating equation method and then extend it to the data streams.

Suppose there are N independent observations $\{(y_i, \mathbf{x}_i),\ i = 1, 2, \ldots, N\}$ of interest and we wish to fit a normal linear regression model

$$y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \epsilon_i, \tag{3.1}$$

where $\epsilon_i \sim N(0, \sigma^2)$ independently for $i = 1, 2, \ldots, N$, and $\boldsymbol{\beta}$ is a p-dimensional vector of regression coefficients corresponding to covariates $\mathbf{x}_i$ $(p \times 1)$. Write $\mathbf{y} = (y_1, y_2, \ldots, y_N)^\top$ and $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^\top$ where we assume the design matrix $\mathbf{X}$ is of full rank $p < N$. The least squares (LS) estimate of $\boldsymbol{\beta}$ and the corresponding residual mean square, or mean squared error (MSE), are given by $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$ and $\mathsf{MSE} = \frac{1}{N-p}\mathbf{y}^\top(\mathbf{I}_N - \mathbf{H})\mathbf{y}$, respectively, where $\mathbf{I}_N$ is the $N \times N$ identity matrix and $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$.

In the online-updating setting, we suppose that the N observations are not available all at once, but rather arrive in chunks from a large data stream. Suppose at each accumulation point k we observe $\mathbf{y}_k$ and $\mathbf{X}_k$, the n_k-dimensional vector of responses and the $n_k \times p$ matrix of covariates, respectively, for $k = 1, \ldots, K$ such that $\mathbf{y} = (\mathbf{y}_1^\top, \mathbf{y}_2^\top, \ldots, \mathbf{y}_K^\top)^\top$ and $\mathbf{X} = (\mathbf{X}_1^\top, \mathbf{X}_2^\top, \ldots, \mathbf{X}_K^\top)^\top$. Provided $\mathbf{X}_k$ is of full rank, the LS estimate of $\boldsymbol{\beta}$ based on the k^{th} subset is given by $\hat{\boldsymbol{\beta}}_{n_k,k} = (\mathbf{X}_k^\top\mathbf{X}_k)^{-1}\mathbf{X}_k^\top\mathbf{y}_k$ and the MSE is given by $\mathsf{MSE}_{n_k,k} = \frac{1}{n_k-p}\mathbf{y}_k^\top(\mathbf{I}_{n_k} - \mathbf{H}_k)\mathbf{y}_k$, where $\mathbf{H}_k = \mathbf{X}_k(\mathbf{X}_k^\top\mathbf{X}_k)^{-1}\mathbf{X}_k^\top$, for $k = 1, 2, \ldots, K$.

As in the divide-and-conquer approach (e.g., Lin and Xi, 2011), we can write $\hat{\boldsymbol{\beta}}$ as

$$\hat{\boldsymbol{\beta}} = \Big(\sum_{k=1}^{K} \mathbf{X}_k^\top\mathbf{X}_k\Big)^{-1} \sum_{k=1}^{K} \mathbf{X}_k^\top\mathbf{X}_k\hat{\boldsymbol{\beta}}_{n_k,k}. \tag{3.2}$$

We provide a similar divide-and-conquer expression for the residual sum of squares, or

sum of squared errors (SSE), given by

$$\mathsf{SSE} \;=\; \sum_{k=1}^{K} \mathbf{y}_k^\top \mathbf{y}_k - \Big(\sum_{k=1}^{K} \mathbf{X}_k^\top \mathbf{X}_k \hat{\boldsymbol{\beta}}_{n_k,k} \Big)^\top \Big(\sum_{k=1}^{K} \mathbf{X}_k^\top \mathbf{X}_k \Big)^{-1} \Big(\sum_{k=1}^{K} \mathbf{X}_k^\top \mathbf{X}_k \hat{\boldsymbol{\beta}}_{n_k,k} \Big), \quad (3.3)$$

and $\mathsf{MSE} = \mathsf{SSE}/(N-p)$. Expression (3.3) is quite useful if one is interested in performing inference in the divide-and-conquer setting, as $\mathsf{Var}(\hat{\boldsymbol{\beta}})$ may be estimated by $\mathsf{MSE}\Big(\sum_{k=1}^{K} \mathbf{X}_k^\top \mathbf{X}_k \Big)^{-1}$. We will see in Section 3.2.2 that both expressions (3.2) and (3.3) may be expressed in sequential form that is more advantageous from the online-updating perspective.

3.2.2 Online Updating

While equations (3.2) and (3.3) are quite amenable to parallel processing for each subset, the online-updating approach for data streams is inherently sequential in nature. Equations (3.2) and (3.3) can certainly be used for estimation and inference for regression coefficients resulting at some terminal point K from a data stream, provided quantities $(\mathbf{X}_k^\top \mathbf{X}_k, \hat{\boldsymbol{\beta}}_{n_k,k}, \mathbf{y}_k^\top \mathbf{y}_k)$ are available for all accumulation points $k = 1, \ldots, K$. However, such data storage may not always be possible or desirable. Furthermore, it may also be of interest to perform inference at a given accumulation step k, using the k subsets of data observed to that point. Thus, our objective is to formulate a computationally efficient and minimally storage-intensive procedure that will allow for online-updating of estimation and inference.

While our ultimate estimation and inferential procedures are frequentist in nature, a Bayesian perspective provides some insight into how we may construct our online-updating estimators. Under a Bayesian framework, using the previous $k-1$ subsets of data to construct a prior distribution for the current data in subset k, we immediate identify the appropriate online updating formulae for estimating the regression coefficients $\boldsymbol{\beta}$ and the error variance σ^2 with each new incoming dataset $(\mathbf{y}_k, \mathbf{X}_k)$. The Bayesian paradigm and accompanying formulae are provided in the Supplementary Material.

Let $\hat{\boldsymbol{\beta}}_k$ and MSE_k denote the LS estimate of $\boldsymbol{\beta}$ and the corresponding MSE based on the cumulative data $D_k = \{(\mathbf{y}_\ell, \mathbf{X}_\ell),\ \ell = 1, 2, \ldots, k\}$. The online-updated estimator of $\boldsymbol{\beta}$ based on cumulative data D_k is given by

$$\hat{\boldsymbol{\beta}}_k = (\mathbf{X}_k^\top \mathbf{X}_k + \mathbf{V}_{k-1})^{-1}(\mathbf{X}_k^\top \mathbf{X}_k \hat{\boldsymbol{\beta}}_{n_k k} + \mathbf{V}_{k-1}\hat{\boldsymbol{\beta}}_{k-1}), \tag{3.4}$$

where $\hat{\boldsymbol{\beta}}_0 = \mathbf{0}$, $\mathbf{V}_k = \sum_{\ell=1}^{k} \mathbf{X}_\ell^\top \mathbf{X}_\ell$ for $k = 1, 2, \ldots,$ and $\mathbf{V}_0 = \mathbf{0}_p$ is a $p \times p$ matrix of zeros. Although motivated through Bayesian arguments, (3.4) may also be found in a (non-Bayesian) recursive linear model framework (e.g., Stengel, 2012, p. 313).

The online-updated estimator of the SSE based on cumulative data D_k is given by

$$\mathsf{SSE}_k \;=\; \mathsf{SSE}_{k-1} + \mathsf{SSE}_{n_k,k} + \hat{\boldsymbol{\beta}}_{k-1}^\top \mathbf{V}_{k-1}\hat{\boldsymbol{\beta}}_{k-1} + \hat{\boldsymbol{\beta}}_{n_k,k}^\top \mathbf{X}_k^\top \mathbf{X}_k \hat{\boldsymbol{\beta}}_{n_k,k} - \hat{\boldsymbol{\beta}}_k^\top \mathbf{V}_k \hat{\boldsymbol{\beta}}_k \tag{3.5}$$

where $\mathsf{SSE}_{n_k,k}$ is the residual sum of squares from the k^{th} dataset, with corresponding

residual mean square $\mathsf{MSE}_{n_k,k} = \mathsf{SSE}_{n_k,k}/(n_k - p)$. The MSE based on the data D_k is then $\mathsf{MSE}_k = \mathsf{SSE}_k/(N_k - p)$ where $N_k = \sum_{\ell=1}^{k} n_\ell$ $(= n_k + N_{k-1})$ for $k = 1, 2, \ldots$. Note that for $k = K$, equations (3.4) and (3.5) are identical to those in (3.2) and (3.3), respectively.

Notice that, in addition to quantities only involving the current data $(\mathbf{y}_k, \mathbf{X}_k)$ (i.e., $\hat{\boldsymbol{\beta}}_{n_k,k}$, $SSE_{n_k,k}$, $\mathbf{X}_k^\top \mathbf{X}_k$, and n_k), we only used quantities $(\hat{\boldsymbol{\beta}}_{k-1}, \mathsf{SSE}_{k-1}, \mathbf{V}_{k-1}, N_{k-1})$ from the previous accumulation point to compute $\hat{\boldsymbol{\beta}}_k$ and MSE_k. Based on these online-updated estimates, one can easily obtain online-updated t-tests for the regression parameter estimates. Online-updated ANOVA tables require storage of two additional scalar quantities from the previous accumulation point; details are provided in the Supplementary Material.

3.2.3 Model Fit Diagnostics

While the advantages of saving only lower-dimensional summaries are clear, a potential disadvantage arises in terms of difficulty performing classical residual-based model diagnostics. Since we have not saved the individual observations from the previous $(k-1)$ datasets, we can only compute residuals based upon the current observations $(\mathbf{y}_k, \mathbf{X}_k)$. For example, one may compute the residuals $e_{ki} = y_{ki} - \hat{y}_{ki}$, where $i = 1, \ldots, n_k$ and $\hat{y}_{ki} = \mathbf{x}_{ki}^\top \hat{\boldsymbol{\beta}}_{n_k,k}$, or even the externally studentized residuals given by

$$t_{ki} = \frac{e_{ki}}{\sqrt{\mathsf{MSE}_{n_k,k(i)}(1 - h_{k,ii})}} = e_{ki}\Big[\frac{n_k - p - 1}{\mathsf{SSE}_{n_k,k}(1 - h_{k,ii}) - e_{ki}^2}\Big]^{1/2}, \tag{3.6}$$

where $h_{k,ii} = \text{Diag}(\mathbf{H}_k)_i = \text{Diag}(\mathbf{X}_k(\mathbf{X}_k^\top\mathbf{X}_k)\mathbf{X}_k^\top)_i$ and $\mathsf{MSE}_{n_k,k(i)}$ is the MSE computed from the k^{th} subset with the i^{th} observation removed, $i = 1, \ldots, n_k$.

However, for model fit diagnostics in the online-update setting, it would arguably be more useful to consider the *predictive residuals*, based on $\hat{\boldsymbol{\beta}}_{k-1}$ from data D_{k-1} with predicted values $\check{\mathbf{y}}_k = (\check{y}_{k1}, \ldots, \check{y}_{kn_k})^\top = \mathbf{X}_k\hat{\boldsymbol{\beta}}_{k-1}$, as $\check{e}_{ki} = y_{ki} - \check{y}_{ki}, \quad i = 1, \ldots, n_k$. Define the standardized predictive residuals as $\check{t}_{ki} = \check{e}_{ki}/\sqrt{\hat{\mathsf{Var}}(\check{e}_{ki})}$, $i = 1, \ldots, n_k$.

Distribution of standardized predictive residuals

To derive the distribution of $\check{t}_{ki}$, we introduce new notation. Denote $\dagger_{k-1} = (\mathbf{y}_1^\top, \ldots, \mathbf{y}_{k-1}^\top)^\top$, and $\boldsymbol{\mathcal{X}}_{k-1}$ and $\boldsymbol{\varepsilon}_{k-1}$ the corresponding $N_{k-1} \times p$ design matrix of stacked $\mathbf{X}_\ell$, $\ell = 1, \ldots, k-1$, and $N_{k-1} \times 1$ random errors, respectively. For new observations $\mathbf{y}_k, \mathbf{X}_k$, we assume $\mathbf{y}_k = \mathbf{X}_k\boldsymbol{\beta} + \boldsymbol{\epsilon}_k$, where the elements of $\boldsymbol{\epsilon}_k$ are independent with mean 0 and variance σ^2 independently of the elements of $\boldsymbol{\varepsilon}_{k-1}$ which also have mean 0 and variance σ^2. Thus, $E(\check{e}_{ki}) = 0$, $\mathsf{Var}(\check{e}_{ki}) = \sigma^2(1 + x_{ki}^\top(\boldsymbol{\mathcal{X}}_{k-1}^\top\boldsymbol{\mathcal{X}}_{k-1})^{-1}x_{ki})$ for $i = 1, \ldots, n_k$, and $\mathsf{Var}(\check{\mathbf{e}}_k) = \sigma^2(\mathbf{I}_{n_k} + \mathbf{X}_k(\boldsymbol{\mathcal{X}}_{k-1}^\top\boldsymbol{\mathcal{X}}_{k-1})^{-1}\mathbf{X}_k^\top)$ where $\check{\mathbf{e}}_k = (\check{e}_{k1}, \ldots, \check{e}_{kn_k})^\top$.

If we assume that both $\boldsymbol{\epsilon}_k$ and $\boldsymbol{\varepsilon}_{k-1}$ are normally distributed, then it is easy to show that $\check{\mathbf{e}}_k^\top\mathsf{Var}(\check{\mathbf{e}}_k)^{-1}\check{\mathbf{e}}_k \sim \chi^2_{n_k}$. Thus, estimating σ^2 with MSE_{k-1} and noting that $\frac{N_{k-1}-p}{\sigma^2}\mathsf{MSE}_{k-1} \sim \chi^2_{N_{k-1}-p}$ independently of $\check{\mathbf{e}}_k^\top\mathsf{Var}(\check{\mathbf{e}}_k)^{-1}\check{\mathbf{e}}_k$, we find that $\check{t}_{ki} \sim t_{N_{k-1}-p}$ and

$$\check{F}_k := \frac{\check{\mathbf{e}}_k^\top(\mathbf{I}_{n_k} + \mathbf{X}_k(\boldsymbol{\mathcal{X}}_{k-1}^\top\boldsymbol{\mathcal{X}}_{k-1})^{-1}\mathbf{X}_k^\top)^{-1}\check{\mathbf{e}}_k}{n_k\mathsf{MSE}_{k-1}} \sim F_{n_k,N_{k-1}-p}. \tag{3.7}$$

If we are not willing to assume normality of the errors, we introduce the following

proposition. The proof of the proposition is given in the Supplementary Material.

Proposition 3.1. *Assume that (i)* ϵ_i, $i = 1, \ldots, n_k$, *are independent and identically distributed with* $E(\epsilon_i) = 0$ *and* $E(\epsilon_i^2) = \sigma^2$*; (ii) the elements of the design matrix* $\boldsymbol{\mathcal{X}}_k$ *are uniformly bounded, i.e.,* $|X_{ij}| < C$, $\forall\ i, j$, *where* $C < \infty$ *is constant; (iii)* $\lim_{N_{k-1}\to\infty} \frac{\boldsymbol{\mathcal{X}}_{k-1}^\top \boldsymbol{\mathcal{X}}_{k-1}}{N_{k-1}} = \boldsymbol{Q}$, *where* $\boldsymbol{Q}$ *is a positive definite matrix. Let* $\check{\boldsymbol{e}}_k^* = \boldsymbol{\Gamma}^{-1}\check{\boldsymbol{e}}_k$, *where* $\boldsymbol{\Gamma}\boldsymbol{\Gamma}^\top \triangleq \boldsymbol{I}_{n_k} + \boldsymbol{X}_k(\boldsymbol{\mathcal{X}}_{k-1}^\top \boldsymbol{\mathcal{X}}_{k-1})^{-1}\boldsymbol{X}_k^\top$. *Write* $\check{\boldsymbol{e}}_k^{*\top} = (\check{\boldsymbol{e}}_{k_1}^{*\top}, \ldots, \check{\boldsymbol{e}}_{k_m}^{*\top})$, *where* $\check{\boldsymbol{e}}_{k_i}^*$ *is an* $n_{k_i} \times 1$ *vector consisting of the* $(\sum_{\ell=1}^{i-1} n_{k_\ell} + 1)$*th component through the* $(\sum_{\ell=1}^{i} n_{k_\ell})$*th component of* $\check{\boldsymbol{e}}_k^*$, *and* $\sum_{i=1}^m n_{k_i} = n_k$. *We further assume that (iv)* $\lim_{n_k\to\infty} \frac{n_{k_i}}{n_k} = C_i$, *where* $0 < C_i < \infty$ *is constant for* $i = 1, \ldots, m$. *Letting* $\boldsymbol{1}_{k_i}$ *be an* $n_{k_i} \times 1$ *vector of all ones, then at accumulation point* k, *we have*

$$\frac{\sum_{i=1}^m \frac{1}{n_{k_i}} (\boldsymbol{1}_{k_i}^\top \check{\boldsymbol{e}}_{k_i}^*)^2}{\mathsf{MSE}_{k-1}} \xrightarrow{d} \chi_m^2, \qquad \text{as } n_k, N_{k-1} \to \infty. \tag{3.8}$$

Tests for Outliers

Under normality of the random errors, we may use the standardized predictive residuals $\check{t}_{ki}$ and $\check{F}_k$ in (3.7) to test individually or globally if there are any outliers in the k^{th} dataset. Notice that $\check{t}_{ki}$ and $\check{F}_k$ can be re-expressed equivalently as

$$\check{t}_{ki} = \check{e}_{ki} / \sqrt{\mathsf{MSE}_{k-1}(1 + x_{ki}^\top (\mathbf{V}_{k-1})^{-1} x_{ki})} \quad \text{and} \quad \check{F}_k = \frac{\check{\mathbf{e}}_k^\top (\mathbf{I}_{n_k} + \mathbf{X}_k (\mathbf{V}_{k-1})^{-1} \mathbf{X}_k^\top)^{-1} \check{\mathbf{e}}_k}{n_k \mathsf{MSE}_{k-1}}, \tag{3.9}$$

respectively, and thus can both be computed with the lower-dimensional stored summary statistics from the previous accumulation point.

We may identify as outlying y_{ki} observations those cases whose standardized predicted $\check{t}_{ki}$ are large in magnitude. If the regression model is appropriate, so that no case is outlying because of a change in the model, then each $\check{t}_{ki}$ will follow the t distribution with $N_{k-1} - p$ degrees of freedom. Let $p_{ki} = P(|t_{N_{k-1}-p}| > |\check{t}_{ki}|)$ be the unadjusted p-value and let $\tilde{p}_{ki}$ be the corresponding *adjusted* p-value for multiple testing (e.g., Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001). We will declare y_{ki} an outlier if $\tilde{p}_{ki} < \alpha$ for a prespecified α level. Note that while the Benjamini-Hochberg (BH) procedure assumes the multiple tests to be independent or positively correlated, the predictive residuals will be approximately independent as the sample size increases. Thus, we would expect the false discovery rate to be controlled with the BH p-value adjustment for large N_{k-1}.

To test if there is at least one outlying value based upon null hypothesis $H_0 : E(\check{\mathbf{e}}_k) = \mathbf{0}$, we will use statistic $\check{F}_k$. Values of the test statistic larger than $F(1-\alpha, n_k, N_{k-1}-p)$ would indicate at least one outlying y_{ki} exists among $i = 1, \ldots, n_k$ at the corresponding α level.

If we are unwilling to assume normality of the random errors, we may still perform a global outlier test under the assumptions of Proposition 3.1. Using Proposition 3.1 and following the calibration proposed in Muirhead (2009) (Muirhead, 2009, page 218), we

obtain an asymptotic F statistic

$$\check{F}_k^a := \frac{\sum_{i=1}^m \frac{1}{n_{k_i}} (\mathbf{1}_{k_i}^\top \check{\mathbf{e}}_{k_i}^*)^2}{\mathsf{MSE}_{k-1}} \frac{N_{k-1} - m + 1}{N_{k-1} \cdot m} \xrightarrow{d} F(m, N_{k-1} - m + 1), \quad \text{as } n_k, N_{k-1} \to \infty. \tag{3.10}$$

Values of the test statistic $\check{F}_k^a$ larger than $F(1-\alpha, m, N_{k-1} - m + 1)$ would indicate at least one outlying observation exists among $\mathbf{y}_k$ at the corresponding α level.

Remark 3.2. *Recall that* $\mathsf{Var}(\check{\boldsymbol{e}}_k) = (\boldsymbol{I}_{n_k} + \boldsymbol{X}_k(\boldsymbol{\mathcal{X}}_{k-1}^\top \boldsymbol{\mathcal{X}}_{k-1})^{-1}\boldsymbol{X}_k^\top)\sigma^2 \triangleq \boldsymbol{\Gamma}\boldsymbol{\Gamma}^\top\sigma^2$, *where* $\boldsymbol{\Gamma}$ *is an* $n_k \times n_k$ *invertible matrix. For large* n_k*, it may be challenging to compute the Cholesky decomposition of* $\mathsf{Var}(\check{\boldsymbol{e}}_k)$*. One possible solution that avoids the large* n_k *issue is given in the Supplementary Material.*

3.3 Estimating Equations

A nice property in the normal linear regression model setting is that regardless of whether one "divides and conquers" or performs online updating, the final solution $\hat{\boldsymbol{\beta}}_K$ will be the same as it would have been if one could fit all of the data simultaneously and obtained $\hat{\boldsymbol{\beta}}$ directly. However, with generalized linear models and estimating equations, this is typically not the case, as the score or estimating functions are often nonlinear in $\boldsymbol{\beta}$. Consequently, divide and conquer strategies in these settings often rely on some form of linear approximation to attempt to convert the estimating equation problem into a least square-type problem. For example, following Lin and Xi (2011), suppose N independent

observations $\{\mathbf{w}_i,\ i = 1, 2, \ldots, N\}$. For generalized linear models, $\mathbf{w}_i$ will be $(y_i, \mathbf{x}_i)$ pairs, $i = 1, \ldots, N$ with $E(y_i) = g(\mathbf{x}_i^\top \boldsymbol{\beta})$ for some known function g. Suppose there exists $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ such that $\sum_{i=1}^N E[\psi(\mathbf{w}_i, \boldsymbol{\beta}_0)] = 0$ for some score or estimating function ψ. Let $\hat{\boldsymbol{\beta}}_N$ denote the solution to the estimating equation (EE) $M(\boldsymbol{\beta}) = \sum_{i=1}^N \psi(\mathbf{w}_i, \boldsymbol{\beta}) = \mathbf{0}$ and let $\hat{\mathbf{V}}_N$ be its corresponding estimate of covariance, often of sandwich form.

Let $\{\mathbf{w}_{ki}, i = 1, \ldots, n_k\}$ be the observations in the kth subset. The estimating function for subset k is $M_{n_k,k}(\boldsymbol{\beta}) = \sum_{i=1}^{n_k} \psi(\mathbf{w}_{ki}, \boldsymbol{\beta})$. Denote the solution to $M_{n_k,k}(\boldsymbol{\beta}) = \mathbf{0}$ as $\hat{\boldsymbol{\beta}}_{n_k,k}$. If we define

$$\mathbf{A}_{n_k,k} = -\sum_{i=1}^{n_k} \frac{\partial \psi(\mathbf{w}_{ki}, \hat{\boldsymbol{\beta}}_{n_k,k})}{\partial \boldsymbol{\beta}}, \tag{3.11}$$

a Taylor Expansion of $-M_{n_k,k}(\boldsymbol{\beta})$ at $\hat{\boldsymbol{\beta}}_{n_k,k}$ is given by $-M_{n_k,k}(\boldsymbol{\beta}) = \mathbf{A}_{n_k,k}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_{n_k,k}) + \mathbf{R}_{n_k,k}$ as $M_{n_k,k}(\hat{\boldsymbol{\beta}}_{n_k,k}) = \mathbf{0}$ and $\mathbf{R}_{n_k,k}$ is the remainder term. As in the linear model case, we do not require $\mathbf{A}_{n_k,k}$ to be invertible for each subset k, but do require that $\sum_{\ell=1}^k \mathbf{A}_{n_\ell,\ell}$ is invertible. Note that for the asymptotic theory in Section 3.3.3, we assume that $\mathbf{A}_{n_k,k}$ is invertible for large n_k. For ease of notation, we will assume for now that each $\mathbf{A}_{n_k,k}$ is invertible, and we will address rank deficient $\mathbf{A}_{n_k,k}$ in Section 3.4 below.

The aggregated estimating equation (AEE) estimator of Lin and Xi (2011) combines the subset estimators through

$$\hat{\boldsymbol{\beta}}_{NK} = \left(\sum_{k=1}^K \mathbf{A}_{n_k,k}\right)^{-1} \sum_{k=1}^K \mathbf{A}_{n_k,k} \hat{\boldsymbol{\beta}}_{n_k,k} \tag{3.12}$$

which is the solution to $\sum_{k=1}^K \mathbf{A}_{n_k,k}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_{n_k,k}) = \mathbf{0}$. Lin and Xi (2011) did not discuss

a variance formula, but a natural variance estimator is given by

$$\hat{\mathbf{V}}_{NK} = \left(\sum_{k=1}^{K} \mathbf{A}_{n_k,k}\right)^{-1} \sum_{k=1}^{K} \mathbf{A}_{n_k,k} \hat{\mathbf{V}}_{n_k,k} \mathbf{A}_{n_k,k}^{\top} \left[\left(\sum_{k=1}^{K} \mathbf{A}_{n_k,k}\right)^{-1}\right]^{\top}, \tag{3.13}$$

where $\hat{\mathbf{V}}_{n_k,k}$ is the variance estimator of $\hat{\boldsymbol{\beta}}_{n_k,k}$ from the subset k. If $\hat{\mathbf{V}}_{n_k,k}$ is of sandwich form, it can be expressed as $\mathbf{A}_{n_k,k}^{-1} \hat{\mathbf{Q}}_{n_k,k} \mathbf{A}_{n_k,k}^{-1}$, where $\hat{\mathbf{Q}}_{n_k,k}$ is an estimate of $\mathbf{Q}_{n_k,k} = \mathsf{Var}(M_{n_k,k}(\boldsymbol{\beta}))$. Then, the variance estimator is still of sandwich form as

$$\hat{\mathbf{V}}_{NK} = \left(\sum_{k=1}^{K} \mathbf{A}_{n_k,k}\right)^{-1} \sum_{k=1}^{K} \hat{\mathbf{Q}}_{n_k,k} \left[\left(\sum_{k=1}^{K} \mathbf{A}_{n_k,k}\right)^{-1}\right]^{\top}. \tag{3.14}$$

3.3.1 Online Updating

Now consider the online-updating perspective in which we would like to update the estimates of $\boldsymbol{\beta}$ and its variance as new data arrives. For this purpose, we introduce the cumulative estimating equation (CEE) estimator for the regression coefficient vector at accumulation point k as

$$\hat{\boldsymbol{\beta}}_k = (\mathbf{A}_{k-1} + \mathbf{A}_{n_k,k})^{-1} (\mathbf{A}_{k-1} \hat{\boldsymbol{\beta}}_{k-1} + \mathbf{A}_{n_k,k} \hat{\boldsymbol{\beta}}_{n_k,k}). \tag{3.15}$$

for $k = 1, 2, \ldots$ where $\hat{\boldsymbol{\beta}}_0 = \mathbf{0}$, $\mathbf{A}_0 = \mathbf{0}_p$, and $\mathbf{A}_k = \sum_{\ell=1}^{k} \mathbf{A}_{n_\ell,\ell} = \mathbf{A}_{k-1} + \mathbf{A}_{n_k,k}$. With $\hat{\mathbf{V}}_0 = \mathbf{0}_p$ and $\mathbf{A}_0 = \mathbf{0}_p$, the variance estimator at the k^{th} update is given by

$$\hat{\mathbf{V}}_k = (\mathbf{A}_{k-1} + \mathbf{A}_{n_k,k})^{-1}(\mathbf{A}_{k-1}\hat{\mathbf{V}}_{k-1}\mathbf{A}_{k-1}^{\top} + \mathbf{A}_{n_k,k}\hat{\mathbf{V}}_{n_k,k}\mathbf{A}_{n_k,k}^{\top})[(\mathbf{A}_{k-1} + \mathbf{A}_{n_k,k})^{-1}]^{\top}. \quad (3.16)$$

By induction, it can be shown that (4.10) is equivalent to the AEE combination (3.12) when $k = K$, and likewise (4.11) is equivalent to (3.14) (i.e., AEE=CEE). However, the AEE estimators, and consequently the CEE estimators, are not identical to the EE estimators $\hat{\boldsymbol{\beta}}_N$ and $\hat{\mathbf{V}}_N$ based on all N observations. It should be noted, however, that Lin and Xi (2011) did prove asymptotic consistency of AEE estimator $\hat{\boldsymbol{\beta}}_{NK}$ under certain regularity conditions. Since the CEE estimators are not identical to the EE estimators in finite sample sizes, there is room for improvement.

Towards this end, consider the Taylor expansion of $-M_{n_k,k}(\boldsymbol{\beta})$ around some vector $\check{\boldsymbol{\beta}}_{n_k,k}$, to be defined later. Then

$$-M_{n_k,k}(\boldsymbol{\beta}) \;=\; -M_{n_k,k}(\check{\boldsymbol{\beta}}_{n_k,k}) + [\mathbf{A}_{n_k,k}(\check{\boldsymbol{\beta}}_{n_k,k})](\boldsymbol{\beta} - \check{\boldsymbol{\beta}}_{n_k,k}) + \check{\mathbf{R}}_{n_k,k}$$

with $\check{\mathbf{R}}_{n_k,k}$ denoting the remainder. Denote $\tilde{\boldsymbol{\beta}}_K$ as the solution of

$$\sum_{k=1}^{K} -M_{n_k,k}(\check{\boldsymbol{\beta}}_{n_k,k}) + \sum_{k=1}^{K}[\mathbf{A}_{n_k,k}(\check{\boldsymbol{\beta}}_{n_k,k})](\boldsymbol{\beta} - \check{\boldsymbol{\beta}}_{n_k,k}) = \mathbf{0}. \quad (3.17)$$

Define $\tilde{\mathbf{A}}_{n_k,k} = [\mathbf{A}_{n_k,k}(\check{\boldsymbol{\beta}}_{n_k,k})]$ and assume $\mathbf{A}_{n_k,k}$ refers to $\mathbf{A}_{n_k,k}(\hat{\boldsymbol{\beta}}_{n_k,k})$. Then we have

$$\tilde{\boldsymbol{\beta}}_K = \left\{ \sum_{k=1}^{K} \tilde{\mathbf{A}}_{n_k,k} \right\}^{-1} \left\{ \sum_{k=1}^{K} \tilde{\mathbf{A}}_{n_k,k} \check{\boldsymbol{\beta}}_{n_k,k} + \sum_{k=1}^{K} M_{n_k,k}(\check{\boldsymbol{\beta}}_{n_k,k}) \right\}. \tag{3.18}$$

If we choose $\check{\boldsymbol{\beta}}_{n_k,k} = \hat{\boldsymbol{\beta}}_{n_k,k}$, then $\tilde{\boldsymbol{\beta}}_K$ in (3.18) reduces to the AEE estimator of Lin and Xi (2011) in (3.12), as (3.17) reduces to $\sum_{k=1}^{K} \mathbf{A}_{n_k,k}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_{n_k,k}) = \mathbf{0}$ because $M_{n_k,k}(\hat{\boldsymbol{\beta}}_{n_k,k}) = \mathbf{0}$ for all $k = 1, \ldots, K$. However, one does not need to choose $\check{\boldsymbol{\beta}}_{n_k,k} = \hat{\boldsymbol{\beta}}_{n_k,k}$. In the online-updating setting, at each accumulation point k, we have access to the summaries from the previous accumulation point $k-1$, so we may use this information to our advantage when defining $\check{\boldsymbol{\beta}}_{n_k,k}$. Consider the intermediary estimator given by

$$\check{\boldsymbol{\beta}}_{n_k,k} = (\tilde{\mathbf{A}}_{k-1} + \mathbf{A}_{n_k,k})^{-1} \left(\sum_{\ell=1}^{k-1} \tilde{\mathbf{A}}_{n_\ell,\ell} \check{\boldsymbol{\beta}}_{n_\ell,\ell} + \mathbf{A}_{n_k,k} \hat{\boldsymbol{\beta}}_{n_k,k} \right) \tag{3.19}$$

for $k = 1, 2, \ldots,$ $\tilde{\mathbf{A}}_0 = \mathbf{0}_p$, $\check{\boldsymbol{\beta}}_{n_0,0} = \mathbf{0}$, and $\tilde{\mathbf{A}}_k = \sum_{\ell=1}^{k} \tilde{\mathbf{A}}_{n_\ell,\ell}$. Estimator (3.19) combines the previous intermediary estimators $\check{\boldsymbol{\beta}}_{n_\ell,\ell}$, $\ell = 1, \ldots, k-1$ and the current subset estimator $\hat{\boldsymbol{\beta}}_{n_k,k}$, and arises as the solution to the estimating equation $\sum_{\ell=1}^{k-1} \tilde{\mathbf{A}}_{n_\ell,\ell}(\boldsymbol{\beta} - \check{\boldsymbol{\beta}}_{n_\ell,\ell}) + \mathbf{A}_{n_k,k}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_{n_k,k}) = \mathbf{0}$, where $\mathbf{A}_{n_k,k}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_{n_k,k})$ serves as a bias correction term due to the omission of $-\sum_{\ell=1}^{k-1} M_{n_k,k}(\check{\boldsymbol{\beta}}_{n_k,k})$ from the equation.

With the choice of $\check{\boldsymbol{\beta}}_{n_k,k}$ as given in (3.19), we introduce the cumulatively updated estimating equation (CUEE) estimator $\tilde{\boldsymbol{\beta}}_k$ as

$$\tilde{\boldsymbol{\beta}}_k \;=\; (\tilde{\mathbf{A}}_{k-1} + \tilde{\mathbf{A}}_{n_k,k})^{-1} (\mathbf{a}_{k-1} + \tilde{\mathbf{A}}_{n_k,k} \check{\boldsymbol{\beta}}_{n_k,k} + \mathbf{b}_{k-1} + M_{n_k,k}(\check{\boldsymbol{\beta}}_{n_k,k})) \tag{3.20}$$

with $\mathbf{a}_k = \sum_{\ell=1}^{k} \tilde{\mathbf{A}}_{n_k,k}\check{\boldsymbol{\beta}}_{n_k,k} = \tilde{\mathbf{A}}_{n_k,k}\check{\boldsymbol{\beta}}_{n_k,k}+\mathbf{a}_{k-1}$ and $\mathbf{b}_k = \sum_{\ell=1}^{k} M_{n_k,k}(\check{\boldsymbol{\beta}}_{n_k,k}) = M_{n_k,k}(\check{\boldsymbol{\beta}}_{n_k,k})+$ $\mathbf{b}_{k-1}$ where $\mathbf{a}_0 = \mathbf{b}_0 = \mathbf{0}$, $\tilde{\mathbf{A}}_0 = \mathbf{0}_p$, and $k = 1, 2, \ldots$. Note that for a terminal $k = K$, (3.20) is equivalent to (3.18).

For the variance of $\tilde{\boldsymbol{\beta}}_k$, observe that $\mathbf{0} = -M_{n_k,k}(\hat{\boldsymbol{\beta}}_{n_k,k}) \approx -M_{n_k,k}(\check{\boldsymbol{\beta}}_{n_k,k})+\tilde{\mathbf{A}}_{n_k,k}(\hat{\boldsymbol{\beta}}_{n_k,k}-$ $\check{\boldsymbol{\beta}}_{n_k,k})$. Thus, we have $\tilde{\mathbf{A}}_{n_k,k}\check{\boldsymbol{\beta}}_{n_k,k} + M_{n_k,k}(\check{\boldsymbol{\beta}}_{n_k,k}) \approx \tilde{\mathbf{A}}_{n_k,k}\hat{\boldsymbol{\beta}}_{n_k,k}$. Using the above approximation, the variance formula is given by

$$\tilde{\mathbf{V}}_k = (\tilde{\mathbf{A}}_{k-1} + \tilde{\mathbf{A}}_{n_k,k})^{-1}(\tilde{\mathbf{A}}_{k-1}\tilde{\mathbf{V}}_{k-1}\tilde{\mathbf{A}}_{k-1}^{\top} + \tilde{\mathbf{A}}_{n_k,k}\hat{\mathbf{V}}_{n_k,k}\tilde{\mathbf{A}}_{n_k,k}^{\top})[(\tilde{\mathbf{A}}_{k-1} + \tilde{\mathbf{A}}_{n_k,k})^{-1}]^{\top} \quad (3.21)$$

for $k = 1, 2, \ldots$ and $\tilde{\mathbf{A}}_0 = \tilde{\mathbf{V}}_0 = \mathbf{0}_p$.

Remark 3.3. *Under the normal linear regression model, all of the estimating equation estimators become "exact", in the sense that* $\hat{\boldsymbol{\beta}}_N = (\boldsymbol{X}^{\top}\boldsymbol{X})^{-1}\boldsymbol{X}^{\top}\boldsymbol{y} = \hat{\boldsymbol{\beta}}_{NK} = \hat{\boldsymbol{\beta}}_K = \tilde{\boldsymbol{\beta}}_K$.

3.3.2 Online Updating for Wald Tests

Wald tests may be used to test individual coefficients or nested hypotheses based upon either the CEE or CUEE estimators from the cumulative data. Let $(\breve{\boldsymbol{\beta}}_k = (\breve{\beta}_{k,1}, \ldots, \breve{\beta}_{k,p})^{\top}, \breve{\mathbf{V}}_k)$ refer to either the CEE regression coefficient estimator and corresponding variance in equations (4.10) and (4.11), or the CUEE regression coefficient estimator and corresponding variance in equations (3.20) and (3.21).

To test $H_0 : \beta_j = 0$ at the k^{th} update ($j = 1, \ldots, p$), we may take the Wald statistic $z_{k,j}^{*2} = \breve{\beta}_{k,j}^2/\mathsf{Var}(\breve{\beta}_{k,j})$, or equivalently, $z_{k,j}^{*} = \breve{\beta}_{k,j}/se(\breve{\beta}_{k,j})$, where the standard error

$se(\breve{\beta}_{k,j}) = \sqrt{\mathsf{Var}(\breve{\beta}_{k,j})}$ and $\mathsf{Var}(\breve{\beta}_{k,j})$ is the j^{th} diagonal element of $\breve{\mathbf{V}}_k$. The corresponding p-value is $P(|Z| \geq |z^*_{k,j}|) = P(\chi^2_1 \geq z^{*2}_{k,j})$ where Z and χ^2_1 are standard normal and 1 degree-of-freedom chi-squared random variables, respectively.

The Wald test statistic may also be used for assessing the difference between a full model M1 relative to a nested submodel M2. If $\boldsymbol{\beta}$ is the parameter of model M1 and the nested submodel M2 is obtained from M1 by setting $\mathbf{C}\boldsymbol{\beta} = \mathbf{0}$, where $\mathbf{C}$ is a rank q contrast matrix and $\breve{\mathbf{V}}$ is a consistent estimate of the covariance matrix of estimator $\breve{\boldsymbol{\beta}}$, the test statistic is $\breve{\boldsymbol{\beta}}^\top \mathbf{C}^\top (\mathbf{C}\breve{\mathbf{V}}\mathbf{C}^\top)^{-1}\mathbf{C}\breve{\boldsymbol{\beta}}$, which is distributed as χ^2_q under the null hypothesis that $\mathbf{C}\boldsymbol{\beta} = \mathbf{0}$. As an example, if M1 represents the full model containing all p regression coefficients at the k^{th} update, where the first coefficient β_1 is an intercept, we may test the global null hypothesis $H_0 : \beta_2 = \ldots = \beta_p = 0$ with $w^*_k = \breve{\boldsymbol{\beta}}_k^\top \mathbf{C}^\top (\mathbf{C}\breve{\mathbf{V}}_k\mathbf{C}^\top)^{-1}\mathbf{C}\breve{\boldsymbol{\beta}}_k$, where $\mathbf{C}$ is $(p-1) \times p$ matrix $\mathbf{C} = [\mathbf{0}, \mathbf{I}_{p-1}]$ and the corresponding p-value is $P(\chi^2_{p-1} \geq w^*_k)$.

3.3.3 Asymptotic Results

In this section, we show consistency of the CUEE estimator. Specifically, Theorem 3.3.1 shows that, under regularity, if the EE estimator based on the all N observations $\hat{\boldsymbol{\beta}}_N$ is a consistent estimator and the partition number K goes to infinity, but not too fast, then the CUEE estimator $\tilde{\boldsymbol{\beta}}_K$ is also a consistent estimator. The technical regularity conditions are provided in the Supplementary Material. We use the same conditions, (C1)-(C6), as Lin and Xi (2011) with the exception of condition (C4). Instead, we use a slightly modified version which focuses on the behavior of $\mathbf{A}_{n,k}(\boldsymbol{\beta})$ for all $\boldsymbol{\beta}$ in the

neighborhood of $\boldsymbol{\beta}_0$ (as in (C5)), rather than just at the subset estimate $\hat{\boldsymbol{\beta}}_{n,k}$.

(C4') In a neighborhood of $\boldsymbol{\beta}_0$, there exists two positive definite matrices $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$ such that $\boldsymbol{\Lambda}_1 \leq n^{-1}\mathbf{A}_{n,k}(\boldsymbol{\beta}) \leq \boldsymbol{\Lambda}_2$ for all $\boldsymbol{\beta}$ in the neighborhood of $\boldsymbol{\beta}_0$ and for all $k = 1, ..., K$.

We assume for simplicity of notation that $n_k = n$ for all $k = 1, 2, \ldots, K$. The proof of the theorem can be found in the Supplementary Material.

Theorem 3.3.1. *Let $\hat{\boldsymbol{\beta}}_N$ be the EE estimator based on entire data. Then under (C1)-(C2), (C4')-(C6), if the partition number K satisfies $K = O(n^\gamma)$ for some $0 < \gamma < min\{1 - 2\alpha, 4\alpha - 1\}$, we have $P(\sqrt{N}\|\tilde{\boldsymbol{\beta}}_K - \hat{\boldsymbol{\beta}}_N\| > \delta) = o(1)$ for any $\delta > 0$.*

Remark 3.4. *If $n_k \neq n$ for all k, Theorem 3.3.1 will still hold, provided for each k, $\frac{n_{k-1}}{n_k}$ is bounded, where n_{k-1} and n_k are the respective sample sizes for subsets $k-1$ and k.*

Remark 3.5. *Suppose N independent observations $(y_i, \boldsymbol{x}_i)$, $i = 1, \ldots, N$, where y is a scalar response and $\boldsymbol{x}$ is a p-dimensional vector of predictor variables. Further suppose $E(y_i) = g(\boldsymbol{x}_i^\top \boldsymbol{\beta})$ for $i = 1, \ldots, N$ for g a continuously differentiable function. Under mild regularity conditions, Lin and Xi (2011) show in their Theorem 5.1 that condition (C6) is satisfied for a simplified version of the quasi-likelihood estimator of $\boldsymbol{\beta}$ (Chen et al., 1999), given as the solution to the estimating equation $Q(\boldsymbol{\beta}) = \sum_{i=1}^N [y_i - g(\boldsymbol{x}_i^\top \boldsymbol{\beta})]\boldsymbol{x}_i = \mathbf{0}$.*

3.4 Rank Deficiencies in the Design Matrix

When dealing with subsets of data, either in the divide-and-conquer or the online-updating setting, it is quite possible (e.g., in the presence of rare event covariates) that some of the design matrix subsets $\mathbf{X}_k$ will not be of full rank, even if the design matrix $\mathbf{X}$ for the entire dataset is of full rank. For a given subset k, note that if the columns of $\mathbf{X}_k$ are not linearly independent, but lie in a space of dimension $q_k < p$, the estimate

$$\hat{\boldsymbol{\beta}}_{n_k,k} = (\mathbf{X}_k^\top \mathbf{X}_k)^- \mathbf{X}_k^\top \mathbf{y}_k, \tag{3.22}$$

where $(\mathbf{X}_k^\top \mathbf{X}_k)^-$ is a generalized inverse of $(\mathbf{X}_k^\top \mathbf{X}_k)$ for subset k, will not be unique. However, both $\hat{\boldsymbol{\beta}}$ and MSE will be unique, which leads us to introduce the following proposition.

Proposition 3.6. *Suppose $\boldsymbol{X}$ is of full rank $p < N$. If the columns of $\boldsymbol{X}_k$ are not linearly independent, but lie in a space of dimension $q_k < p$ for any $k = 1, \ldots, K$, $\hat{\boldsymbol{\beta}}$ in* (3.2) *and SSE* (3.3) *using $\hat{\boldsymbol{\beta}}_{n_k,k}$ as in* (3.22) *will be invariant to the choice of generalized inverse $(\boldsymbol{X}_k^\top \boldsymbol{X}_k)^-$.*

To see this, recall that a generalized inverse of a matrix $\mathbf{B}$, denoted by $\mathbf{B}^-$, is a matrix such that $\mathbf{B}\mathbf{B}^-\mathbf{B} = \mathbf{B}$. Note that for $(\mathbf{X}_k^\top \mathbf{X}_k)^-$, a generalized inverse of $(\mathbf{X}_k^\top \mathbf{X}_k)$, $\hat{\boldsymbol{\beta}}_{n_k,k}$ given in (3.22) is a solution to the linear system $(\mathbf{X}_k^\top \mathbf{X}_k)\boldsymbol{\beta}_k = \mathbf{X}_k^\top \mathbf{y}_k$. It is well known that if $(\mathbf{X}_k^\top \mathbf{X}_k)^-$ is a generalized inverse of $(\mathbf{X}_k^\top \mathbf{X}_k)$, then $\mathbf{X}_k(\mathbf{X}_k^\top \mathbf{X}_k)^- \mathbf{X}_k^\top$ is invariant to the choice of $(\mathbf{X}_k^\top \mathbf{X}_k)^-$ (e.g., Searle, 1971, p20). Both (3.2) and (3.3) rely on $\hat{\boldsymbol{\beta}}_{n_k,k}$

only through product $\mathbf{X}_k^\top \mathbf{X}_k \hat{\boldsymbol{\beta}}_{n_k,k} = \mathbf{X}_k^\top \mathbf{X}_k (\mathbf{X}_k^\top \mathbf{X}_k)^- \mathbf{X}_k^\top \mathbf{y}_k = \mathbf{X}_k^\top \mathbf{y}_k$ which is invariant to the choice of $(\mathbf{X}_k^\top \mathbf{X}_k)^-$.

Remark 3.7. *The online-updating formulae* (3.4) *and* (3.5) *do not require* $\boldsymbol{X}_k^\top \boldsymbol{X}_k$ *for all* k *to be invertible. In particular, the online-updating scheme only requires* $\boldsymbol{V}_k = \sum_{\ell=1}^k \boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell$ *to be invertible. This fact can be made more explicit by rewriting* (3.4) *and* (3.5)*, respectively, as*

$$\hat{\boldsymbol{\beta}}_k = (\boldsymbol{X}_k^\top \boldsymbol{X}_k + \boldsymbol{V}_{k-1})^{-1}(\boldsymbol{X}_k^\top \boldsymbol{y}_k + \boldsymbol{W}_{k-1}) = \boldsymbol{V}_k^{-1}(\boldsymbol{X}_k^\top \boldsymbol{y}_k + \boldsymbol{W}_{k-1}) \tag{3.23}$$

$$\mathsf{SSE}_k = \mathsf{SSE}_{k-1} + \boldsymbol{y}_k^\top \boldsymbol{y}_k + \hat{\boldsymbol{\beta}}_{k-1}^\top \boldsymbol{V}_{k-1} \hat{\boldsymbol{\beta}}_{k-1} - \hat{\boldsymbol{\beta}}_k^\top \boldsymbol{V}_k \hat{\boldsymbol{\beta}}_k \tag{3.24}$$

where $\boldsymbol{W}_0 = \mathbf{0}$ *and* $\boldsymbol{W}_k = \sum_{\ell=1}^k \boldsymbol{X}_\ell^\top \boldsymbol{y}_\ell$ *for* $k = 1, 2, \ldots$.

Remark 3.8. *Following Remark 3.7 and using the Bayesian motivation discussed in the Supplementary Material, if* $\boldsymbol{X}_1$ *is not of full rank (e.g., due to a rare event covariate), we may consider a regularized least squares estimator by setting* $\boldsymbol{V}_0 \neq \mathbf{0}_p$*. For example, setting* $\boldsymbol{V}_0 = \lambda \boldsymbol{I}_p, \lambda > 0$*, with* $\boldsymbol{\mu}_0 = \mathbf{0}$ *would correspond to a ridge estimator and could be used at the beginning of the online estimation process until enough data has accumulated; once enough data has accumulated, the biasing term* $\boldsymbol{V}_0 = \lambda \boldsymbol{I}_p$ *may be removed such that the remaining sequence of updated estimators* $\hat{\boldsymbol{\beta}}_k$ *and* MSE_k *are unbiased for* $\boldsymbol{\beta}$ *and* σ^2*, respectively. Further details are provided in the Supplementary Material.*

Suppose N independent observations $(y_i, \mathbf{x}_i)$, $i = 1, \ldots, N$, where y is a scalar response and $\mathbf{x}$ is a p-dimensional vector of predictor variables. Using the same notation

from the linear model setting, let $(y_{ki}, \mathbf{x}_{ki})$, $i = 1, \ldots, n_k$, be the observations from the k^{th} subset where $\mathbf{y}_k = (y_{k1}, y_{k2}, \ldots, y_{kn_k})^\top$ and $\mathbf{X}_k = (\mathbf{x}_{k1}, \mathbf{x}_{k2}, \ldots, \mathbf{x}_{kn_k})^\top$. For subsets k in which $\mathbf{X}_k$ is not of full rank, we may have difficulty in solving the subset EE to obtain $\hat{\boldsymbol{\beta}}_{n_k,k}$, which is used to compute both the AEE/CEE and CUEE estimators for $\boldsymbol{\beta}$ in (3.12) and (3.18), respectively. However, just as in the linear model case, we can show under certain conditions that if $\mathbf{X} = (\mathbf{X}_1^\top, \mathbf{X}_2^\top, \ldots, \mathbf{X}_K^\top)^\top$ has full column rank p, then the estimators $\hat{\boldsymbol{\beta}}_{NK}$ in (3.12) and $\tilde{\boldsymbol{\beta}}_K$ in (3.18) for some terminal K will be unique.

Specifically, consider observations $(\mathbf{y}_k, \mathbf{X}_k)$ such that $E(y_{ki}) = \mu_{ki} = g(\eta_{ki})$ with $\eta_{ki} = \mathbf{x}_{ki}^\top \boldsymbol{\beta}$ for some known function g. The estimating function ψ for the k^{th} dataset is of the form $\psi(\mathbf{w}_{ki}, \boldsymbol{\beta}) = \mathbf{x}_{ki} S_{ki} W_{ki} (y_{ki} - \mu_{ki}),\ i = 1, \ldots, n_k$, where $S_{ki} = \partial \mu_{ki} / \partial \eta_{ki}$, and W_{ki} is a positive and possibly data dependent weight. Specifically, W_{ki} may depend on $\boldsymbol{\beta}$ only through η_{ki}. In matrix form, the estimating equation becomes

$$\mathbf{X}_k^\top \mathbf{S}_k^\top \mathbf{W}_k (\mathbf{y}_k - \boldsymbol{\mu}_k) \quad = \quad \mathbf{0}, \tag{3.25}$$

where $\mathbf{S}_k = \mathrm{Diag}(S_{k1}, \ldots, S_{kn_k})$, $\mathbf{W}_k = \mathrm{Diag}(W_{k1}, \ldots, W_{kn_k})$, and $\boldsymbol{\mu}_k = \big(\mu_{k1}, \ldots, \mu_{kn_k}\big)^\top$.

With $\mathbf{S}_k$, $\mathbf{W}_k$, and $\boldsymbol{\mu}_k$ evaluated at some initial value $\boldsymbol{\beta}^{(0)}$, the standard Newton–Raphson method for the iterative solution of (3.25) solves the linear equations

$$\mathbf{X}_k^\top \mathbf{S}_k^\top \mathbf{W}_k \mathbf{S}_k \mathbf{X}_k (\boldsymbol{\beta} - \boldsymbol{\beta}^{(0)}) = \mathbf{X}_k^\top \mathbf{S}_k^\top \mathbf{W}_k (\mathbf{y}_k - \boldsymbol{\mu}_k) \tag{3.26}$$

for an updated $\boldsymbol{\beta}$. Rewrite equation (3.26) as $\mathbf{X}_k^\top \mathbf{S}_k^\top \mathbf{W}_k \mathbf{S}_k \mathbf{X}_k \boldsymbol{\beta} = \mathbf{X}_k^\top \mathbf{S}_k^\top \mathbf{W}_k \mathbf{v}_k$ where $\mathbf{v}_k = \mathbf{y}_k - \boldsymbol{\mu}_k + \mathbf{S}_k \mathbf{X}_k \boldsymbol{\beta}^{(0)}$; this can be recognized as the normal equation of a weighted least squares regression with response $\mathbf{v}_k$, design matrix $\mathbf{S}_k \mathbf{X}_k$, and weight $\mathbf{W}_k$. Therefore the iterative reweighted least squares approach (IRLS) can be used to implement the Newton–Raphson method for an iterative solution to (3.25) (e.g., Green, 1984).

Rank deficiency in $\mathbf{X}_k$ calls for a generalized inverse of $\mathbf{X}_k^\top \mathbf{S}_k^\top \mathbf{W}_k \mathbf{S}_k \mathbf{X}_k$. In order to show uniqueness of estimators $\hat{\boldsymbol{\beta}}_{NK}$ in (3.12) and $\tilde{\boldsymbol{\beta}}_K$ in (3.18) for some terminal K, we must first establish that the IRLS algorithm will work and converge for subset k given the same initial value $\boldsymbol{\beta}^{(0)}$ when $\mathbf{X}_k$ is not of full rank. Upon convergence of IRLS at subset k with solution $\hat{\boldsymbol{\beta}}_{n_k,k}$, we must then verify that the CEE and CUEE estimators that rely on $\hat{\boldsymbol{\beta}}_{n_k,k}$ are unique. The following proposition summarizes the result; the proof is provided in the Supplementary Material.

Proposition 3.9. *Under the above formulation, assuming that conditions (C1-C3) hold for a full-rank sub-column matrix of $\boldsymbol{X}_k$, estimators $\hat{\boldsymbol{\beta}}_{NK}$ in* (3.12) *and $\tilde{\boldsymbol{\beta}}_K$ in* (3.18) *for some terminal K will be unique provided $\boldsymbol{X}$ is of full rank.*

The simulations in Section 3.6.2 consider rank deficiencies in binary logistic regression and Poisson regression. Note that for these models, the variance of the estimators $\hat{\boldsymbol{\beta}}_K$ and $\tilde{\boldsymbol{\beta}}_K$ are given by $\mathbf{A}_K^{-1} = (\sum_{k=1}^K \mathbf{A}_{n_k,k})^{-1}$ or $\tilde{\mathbf{A}}_K^{-1} = (\sum_{k=1}^K \tilde{\mathbf{A}}_{n_k,k})^{-1}$. For robust sandwich estimators, for those subsets k in which $\mathbf{A}_{n_k,k}$ is not invertible, we replace $\mathbf{A}_{n_k,k} \hat{\mathbf{V}}_{n_k,k} \mathbf{A}_{n_k,k}^\top$ and $\tilde{\mathbf{A}}_{n_k,k} \hat{\mathbf{V}}_{n_k,k} \tilde{\mathbf{A}}_{n_k,k}^\top$ in the "meat" of equations (4.11) and

(3.21), respectively, with an estimate of $\mathbf{Q}_{n_k,k}$ from (3.14). In particular, we use $\hat{\mathbf{Q}}_{n_k,k} = \sum_{i=1}^{n_k} \psi(\mathbf{w}_{ki}, \hat{\boldsymbol{\beta}}_k)\psi(\mathbf{w}_{ki}, \hat{\boldsymbol{\beta}}_k)^\top$ for the CEE variance and $\tilde{\mathbf{Q}}_{n_k,k} = \sum_{i=1}^{n_k} \psi(\mathbf{w}_{ki}, \tilde{\boldsymbol{\beta}}_k)\psi(\mathbf{w}_{ki}, \tilde{\boldsymbol{\beta}}_k)^\top$ for the CUEE variance. We use these modifications in the robust Poisson regression simulations in Section 3.6.2 for the CEE and CUEE estimators, as by design, we include binary covariates with somewhat low success probabilities. Consequently, not all subsets k will observe both successes and failures, particularly for covariates with success probabilities of 0.1 or 0.01, and the corresponding design matrices $\mathbf{X}_k$ will not always be of full rank. Thus $\mathbf{A}_{n_k,k}$ will not always be invertible for finite n_k, but will be invertible for large enough n_k. We also present results of a proof-of-concept simulation for binary logistic regression in the Supplementary Material, where we compare CUEE estimators under different choices of generalized inverses.

3.5 Criterion-Based Variable Selection with Online Updating

To the best of our knowledge, criterion-based variable selection has not yet been considered in the online updating context. This problem is well worth investigating especially when access/storage of the historical data is limited. Follow the same settings in Section 3.2.1 and consider the standard linear regression model (3.1) for the whole data. Let $\mathcal{M}$ denote the model space. We enumerate the models in $\mathcal{M}$ by $m = 1, 2, ..., 2^p$, where 2^p is the dimension of $\mathcal{M}$. For the full model, the least squares estimate of $\boldsymbol{\beta}$ and

the sum of squared errors based on the k^{th} subset is given by $\hat{\boldsymbol{\beta}}_{n_k,k} = (\mathbf{X}_k^\top \mathbf{X}_k)^- \mathbf{X}_k^\top \mathbf{y}_k$ and $\mathsf{SSE}_{n_k,k}$. In the sequential setting, we only need to store and update the cumulative estimates at each k (see, e.g. Schifano et al., 2016).

Let $\boldsymbol{\beta}_k^{(m)} = (\beta_0^{(m)}, \beta_1^{(m)}, \ldots, \beta_{p_m}^{(m)})^\top$ and $\mathsf{SSE}_k^{(m)}$ denote the cumulative estimates based on all data through subset k for model m, where p_m is the number of covariates for model m. We further introduce the $(p+1) \times (p_m+1)$ selection matrix $P^{(m)} = (e_{m_0}, \quad e_{m_1}, \quad \ldots \quad e_{m_{p_m}})$, where e_{m_0} is a vector with length $(p+1)$ and the first element as 1, and e_{m_j} denotes a vector of length $(p+1)$ with 1 in the m_jth position and 0 in every other position for all $j > 0$. Here $(m_1, ..., m_{p_m})$ are not necessarily in sequence, but represents the index of selected variables in the full design matrix $\mathbf{X}_k$. Define $\mathbf{X}_k^{(m)} = \mathbf{X}_k P^{(m)}$. Update a $(p_m+1) \times (p_m+1)$ matrix

$$V_k^{(m)} = \mathbf{X}_k^{(m)^\top} \mathbf{X}_k^{(m)} + V_{k-1}^{(m)},$$

where $V_0^{(m)} = 0$, and a $(p_m+1) \times 1$ vector

$$A_k^{(m)} = \mathbf{X}_k^{(m)^\top} \mathbf{X}_k \hat{\boldsymbol{\beta}}_{n_k,k} + V_{k-1}^{(m)} \hat{\boldsymbol{\beta}}_{k-1}^{(m)},$$

where $\hat{\boldsymbol{\beta}}_0^{(m)} = 0$. After some algebra, we have

$$\hat{\boldsymbol{\beta}}_k^{(m)} = (V_k^{(m)})^{-1} A_k^{(m)},$$

and

$$\mathsf{SSE}_k^{(m)} = \mathsf{SSE}_{n_k k} + \hat{\boldsymbol{\beta}}_{n_k k}^{\top} \mathbf{X}_k^{\top} \mathbf{X}_k \hat{\boldsymbol{\beta}}_{n_k k} + \hat{\boldsymbol{\beta}}_{k-1}^{(m)\top} V_{k-1}^{(m)} \hat{\boldsymbol{\beta}}_{k-1}^{(m)} - \hat{\boldsymbol{\beta}}_k^{(m)\top} V_k^{(m)} \hat{\boldsymbol{\beta}}_k^{(m)} + \mathsf{SSE}_{k-1}^{(m)}.$$

With θ unknown, letting

$$B_k^{(m)} = N \log \frac{2\pi \mathsf{SSE}_k^{(m)}}{N - p_m - 1},$$

the Akaike information criterion (AIC) and Bayesian information criterion (BIC) are updated by

$$\mathsf{AIC}_k^{(m)} = B_k^{(m)} + N + p_m + 1,$$

$$\mathsf{BIC}_k^{(m)} = B_k^{(m)} + N - p_m - 1 + (p_m + 1) \log N.$$

To study the Bayesian variable selection criteria, assume a joint conjugate prior for $(\boldsymbol{\beta}^{(m)}, \theta^{(m)})$ as follows: $\boldsymbol{\beta}^{(m)} | \theta^{(m)}$ follows normal distribution with mean $\boldsymbol{\mu}_0$, and precision matrix $\mathbf{V}_0$, $\theta^{(m)}$ follows Inverse Gamma distribution with shape parameter $\nu_0/2$ and scale

parameter $\tau_0/2$, e.g,

$$\pi(\boldsymbol{\beta}^{(m)}, \theta^{(m)} | \boldsymbol{\mu}_0, \mathbf{V}_0, \nu_0, \tau_0)$$
$$= \pi(\boldsymbol{\beta}^{(m)} | \theta^{(m)}, \boldsymbol{\mu}_0, \mathbf{V}_0) \pi(\theta^{(m)} | \nu_0, \tau_0),$$

where $\boldsymbol{\mu}_0$ is a prespecified $(p_m + 1)$-dimensional vector, $\mathbf{V}_0$ is a $(p_m + 1) \times (p_m + 1)$ positive definite matrix, $\nu_0 > 0$, $\tau_0 > 0$. It can be shown that the deviance information criterion (DIC) (Spiegelhalter et al., 2002) is updated by

$$\mathsf{DIC}_k^{(m)} = N \log \frac{\pi(N-2)\mathsf{SSE}_k^{(m)}}{2} + 2N\psi(\frac{n}{2}) + 2p_m + N + 4,$$

where $\psi(x) = \mathrm{dlog}\,\Gamma(x)/\mathrm{d}x$ is the digamma function.

3.6 Simulation Study

3.6.1 Normal Linear Regression

Residual Diagnostic Performance

In this section we evaluate the performance of the outlier tests discussed in Section 3.2.3. Let k^* denote the index of the single subset of data containing any outliers. We generated the data according to the model $y_{ki} = \mathbf{x}_{ki}^\top \boldsymbol{\beta} + \epsilon_{ki} + b_k \delta \eta_{ki}$, $i = 1, \ldots, n_k$, where $b_k = 0$ if

$k \neq k^*$ and $b_k \sim$ Bernoulli(0.05) otherwise. Notice that the first two terms on the right-hand-side correspond to the usual linear model with $\boldsymbol{\beta} = (1, 2, 3, 4, 5)^\top$, $x_{ki[2:5]} \sim N(\mathbf{0}, \mathbf{I}_4)$ independently, $x_{ki[1]} = 1$, and ϵ_{ki} are the independent errors, while the final term is responsible for generating the outliers. Here, $\eta_{ki} \sim \text{Exp}(1)$ independently and δ is the scale parameter controlling magnitude or strength of the outliers. We set $\delta \in \{0, 2, 4, 6\}$ corresponding to "no", "small", "medium", and "large" outliers.

To evaluate the performance of the individual outlier t-test in (3.9), we generated the random errors as $\epsilon_{ki} \sim \text{N}(0, 1)$. To evaluate the performance of the global outlier F-tests in (3.9) and (3.10), we additionally considered ϵ_{ki} as independent skew-t variates with degrees of freedom $\nu = 3$ and skewing parameter $\gamma = 1.5$, standardized to have mean 0 and variance 1. To be precise, we use the skew t density, $g(x) = \frac{2}{\gamma + \frac{1}{\gamma}} f(\gamma x)$ for $x < 0$ and $g(x) = \frac{2}{\gamma + \frac{1}{\gamma}} f(\frac{x}{\gamma})$ for $x \geq 0$, where $f(x)$ is the density of the t distribution with ν degrees of freedom.

For all outlier simulations, we varied k^*, the location along the data stream in which the outliers occur. We also varied $n_k = n_k* \in \{100, 500\}$ which additionally controls the number of outliers in dataset k^*. For each subset $\ell = 1, \ldots, k^* - 1$ and for 95% of observations in subset k^*, the data did not contain any other outliers.

To evaluate the global outlier F-tests (3.9) and (3.10) with $m = 2$, we estimated power using $B = 500$ simulated data sets with significance level $\alpha = 0.05$, where power was estimated as the proportion of 500 datasets in which $\check{F}_{k^*} \geq F(0.95, n_{k^*}, N_{k^*-1} - 5)$ or $\check{F}^a_{k^*} \geq F(0.95, 2, N_{k^*-1} - 1)$. The power estimates for the various subset sample sizes n_{k^*},

Table 4: Power of the outlier tests for various locations of outliers (k^*), subset sample sizes ($n_k = n_{k*}$), and outlier strengths (no, small, medium, large). Within each cell, the top entry corresponds to the normal-based F test and the bottom entry corresponds to the asymptotic F test that does not rely on normality of the errors.

Outlier	$n_{k*} = 100$ (5 true outliers)				$n_{k*} = 500$ (25 true outliers)			
Strength	$k^* = 5$	$k^* = 10$	$k^* = 25$	$k^* = 100$	$k^* = 5$	$k^* = 10$	$k^* = 25$	$k^* = 100$
	F Test/Asymptotic F Test(m=2)				F Test/Asymptotic F Test(m=2)			
Standard Normal Errors								
none	0.0626	0.0596	0.0524	0.0438	0.0580	0.0442	0.0508	0.0538
	0.0526	0.0526	0.0492	0.0528	0.0490	0.0450	0.0488	0.0552
small	0.5500	0.5690	0.5798	0.5718	0.9510	0.9630	0.9726	0.9710
	0.2162	0.2404	0.2650	0.2578	0.6904	0.7484	0.7756	0.7726
medium	0.9000	0.8982	0.9094	0.9152	1.0000	1.0000	1.0000	1.0000
	0.5812	0.6048	0.6152	0.6304	0.9904	0.9952	0.9930	0.9964
large	0.9680	0.9746	0.9764	0.9726	1.0000	1.0000	1.0000	1.0000
	0.5812	0.6048	0.6152	0.6304	0.9998	1.0000	1.0000	1.0000
Standardized Skew t Errors								
none	0.2400	0.2040	0.1922	0.1656	0.2830	0.2552	0.2454	0.2058
	0.0702	0.0630	0.0566	0.0580	0.0644	0.0580	0.0556	0.0500
small	0.5252	0.4996	0.4766	0.4520	0.7678	0.7598	0.7664	0.7598
	0.2418	0.2552	0.2416	0.2520	0.6962	0.7400	0.7720	0.7716
medium	0.8302	0.8280	0.8232	0.8232	0.9816	0.9866	0.9928	0.9932
	0.5746	0.5922	0.6102	0.6134	0.9860	0.9946	0.9966	0.9960
large	0.9296	0.9362	0.9362	0.9376	0.9972	0.9970	0.9978	0.9990
	0.7838	0.8176	0.8316	0.8222	0.9988	0.9992	0.9998	1.0000

Power with "outlier strength = no" are Type I errors.

locations of outliers k^*, and outlier strengths δ appear in Table 4. When the errors were normally distributed, notice that the Type I error rate was controlled in all scenarios for both the F test and asymptotic F test. As expected, power tends to increase as outlier strength and/or the number of outliers increase. Furthermore, larger values of k^*, and hence greater proportions of "good" outlier-free data, also tend to have higher power; however, the magnitude of improvement decreases once the denominator degrees of freedom ($N_{k^*-1} - p$ or $N_{k^*-1} - m + 1$) become large enough, and the F tests essentially

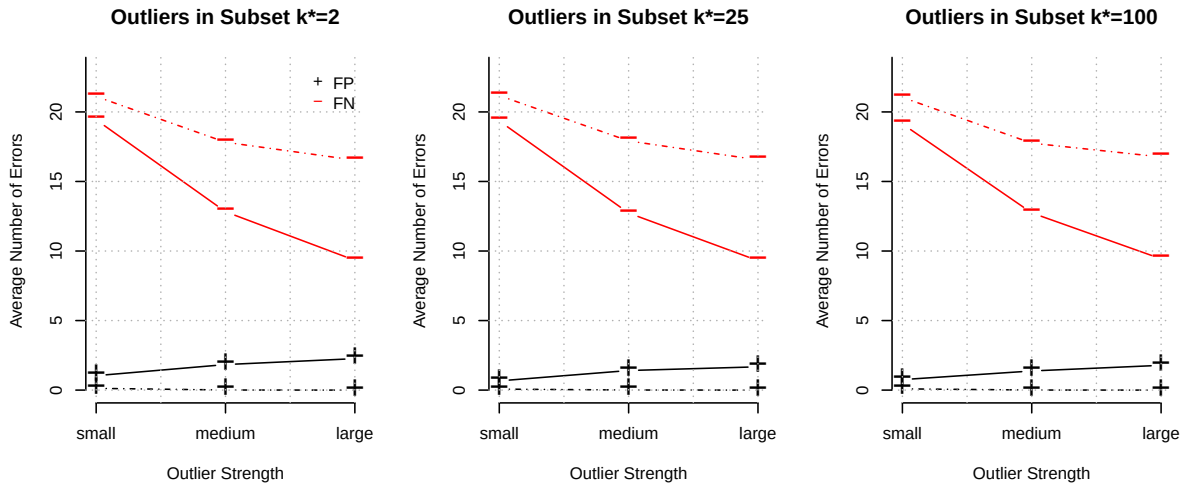

Figure 1: Average numbers of False Positives and False Negatives for outlier t-tests for $n_{k^*} = 500$. Solid lines correspond to the predictive residual test while dotted lines correspond to the externally studentized residuals test using only data from subset k^*.

reduce to χ^2 tests. Also as expected, the F test given by (3.9) is more powerful than the asymptotic F test given in (3.10) when, in fact, the errors were normally distributed. When the errors were not normally distributed, the empirical type I error rates of the F test given by (3.9) are severely inflated and hence, its empirical power in the presence of outliers cannot be trusted. The asymptotic F test, however, maintains the appropriate size.

For the outlier t-test in (3.9), we examined the average number of false negatives (FN) and average number of false positives (FP) across the $B = 500$ simulations. False negatives and false positives were declared based on a BH adjusted p-value threshold of 0.10. These values were plotted in solid lines against outlier strength in Figure 1 for $n_{k^*} = 500$ for various values of k^* and δ; the corresponding plot for $n_{k^*} = 100$ is given in the Supplementary Material. Within each plot the FN decreases as outlier

strength increases, and also tends to decrease slightly across the plots as k^* increases. FP increases slightly as outlier strength increases, but decreases as k^* increases. As with the outlier F test, once the degrees of freedom $N_{k^*-1} - p$ get large enough, the t-test behaves more like a z-test based on the standard normal distribution. For comparison, we also considered FN and FP for an outlier test based upon the externally studentized residuals t_{k^*i} from subset k^* only. Specifically, under the assumed linear model, t_{k^*i} as given by (3.6) follow a t distribution with $n_{k^*} - p - 1$ degrees of freedom. Again, false negatives and false positives were declared based on a BH adjusted p-value threshold of 0.10, and the FN and FP for the externally studentized residual (ESR) test are plotted in dashed lines in Figure 1 for $n_{k^*} = 500$; the plot for $n_{k^*} = 100$ may be found in the Supplementary Materials. This ESR test tends to have a lower FP, but higher FN than the predictive residual test that uses the previous data. Also, the FN and FP for the ESR test are essentially constant across k^* for fixed n_{k^*}, as the ESR test relies on only the current dataset of size n_{k^*} and not the amount of previous data controlled by k^*. Consequently, the predictive residual test has improved power over the ESR test, while still maintaining a low number of FP.

Variable Selection

In a simulation for the variable selection, we examined the performance of AIC, BIC and DIC under the online updating scenario. Each dataset was generated from linear model $y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \epsilon_i$, where ϵ_i's were independently generated from $N(0, 100)$,

$x_i = (1, x_{i1}, x_{i2}, x_{i3}, x_{i4})$ were identically distributed random vectors from a multivariate normal distribution with mean $(1, 0, 0, 0, 0)$ and marginal variances $(0, 16, 9, 0.3, 3)$. Two correlation structures of $(x_{i1}, x_{i2}, x_{i3}, x_{i4})$ were considered: 1) independent and 2) AR(1) with correlation coefficient 0.9. Four different models as determined by the nonzeroness of $\boldsymbol{\beta}$ were considered: $(-1, 3, 0, 0, 0)$, $(-1, 3, 0, -1.5, 0)$, $(-1, 3, 2, -1.5, 0)$, and $(-1, 3, 2, -1.5, 1)$. The corresponding signal-to-noise ratios were 1.44, 1.45, 1.81, and 1.83 in the independent case and 1.44, 1.29, 2.85, and 3.33 under the dependent case. The sample size of each block was set as $n_k = 100$. The performance of the criteria was investigated with the cumulative estimates at block $k \in \{2, 25, 100\}$. For each scenario, 10,000 independent datasets were generated.

The percentages of models selected among the 2^4 models by each of the three criteria are summarized in Table 5. The entire row in bold represents the true model. Based on the simulation results, BIC performs extremely well when the number of blocks (k) is large, which is consistent with known results that the probability of selecting the true model by BIC approaches 1 as $n \to \infty$ (e.g., Schwarz, 1978; Nishii, 1984). The BIC also performs better than AIC and DIC when the covariates are independent, even for small sample sizes. When covariates are highly dependent, AIC and DIC provide more reliable results when sample size is small. The performance of AIC and DIC is always very similar. The simulation results also confirm the existing theorem that AIC is not consistent (e.g., Woodroofe, 1982). In the big data setting with large sample size, BIC is generally preferable, especially when the covariates are not highly correlated.

Table 5: Percentages of the simulations that identify the variables indicated on the left for various number of blocks (k), subset sample sizes ($n_k = 100$) and correlation within the design matrix $\mathbf{X}$ (independent or dependent).

	independent									dependent								
True	$k=2$			$k=25$			$k=100$			$k=2$			$k=25$			$k=100$		
Model	AIC	BIC	DIC	AIC	BIC	DIC	AIC	BIC	DIC	AIC	BIC	DIC	AIC	BIC	DIC	AIC	BIC	DIC
$\boldsymbol{\beta} = (-1, 3, 0, 0, 0)$, signal-to-noise ratios are 1.44 for both independent and dependent.																		
none	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1)	**59**	**93**	**59**	**60**	**98**	**60**	**59**	**99**	**59**	**63**	**94**	**62**	**64**	**99**	**64**	**64**	**99**	**64**
(x_2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2)	11	2	11	11	1	11	12	0	12	10	2	10	9	1	9	10	0	10
(x_3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_3)	11	2	11	11	1	11	11	0	11	8	2	8	8	0	8	8	0	8
(x_2, x_3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2, x_3)	2	0	3	2	0	2	2	0	2	4	0	4	3	0	3	3	0	3
(x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_4)	11	2	11	11	0	11	11	0	11	9	2	9	8	0	9	8	0	8
(x_2, x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2, x_4)	2	0	2	2	0	2	2	0	2	3	0	3	3	0	3	3	0	3
(x_3, x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_3, x_4)	2	0	2	2	0	2	2	0	2	4	0	4	4	0	4	4	0	4
(x_2, x_3, x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2, x_3, x_4)	1	0	1	0	0	0	0	0	0	1	0	1	1	0	1	1	0	1
$\boldsymbol{\beta} = (-1, 3, 0, -1.5, 0)$, signal-to-noise ratios are 1.45 for independent and 1.29 for dependent.																		
none	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1)	42	83	42	0	9	0	0	0	0	55	89	55	10	60	10	0	3	0
(x_2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2)	8	2	8	0	0	0	0	0	0	11	3	11	10	4	10	1	2	1
(x_3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_3)	**28**	**12**	**27**	**71**	**90**	**71**	**70**	**100**	**70**	**13**	**4**	**13**	**50**	**30**	**50**	**69**	**90**	**69**
(x_2, x_3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2, x_3)	6	0	6	13	0	13	14	0	14	4	0	4	6	0	6	12	0	12
(x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_4)	8	2	8	0	0	0	0	0	0	10	3	10	14	6	14	3	5	3
(x_2, x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2, x_4)	2	0	2	0	0	0	0	0	0	3	0	3	2	0	2	2	0	2
(x_3, x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_3, x_4)	6	0	6	13	0	13	13	0	13	4	0	5	6	0	6	11	0	11
(x_2, x_3, x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2, x_3, x_4)	1	0	1	2	0	3	3	0	3	1	0	1	1	0	1	2	0	2
$\boldsymbol{\beta} = (-1, 3, 2, -1.5, 0)$, signal-to-noise ratios are 1.81 for independent and 2.85 for dependent.																		
none	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1)	0	0	0	0	0	0	0	0	0	2	17	2	0	0	0	0	0	0
(x_2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2)	50	85	50	0	9	0	0	0	0	64	74	64	28	83	28	1	29	1
(x_3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_3)	0	0	0	0	0	0	0	0	0	3	2	3	0	0	0	0	0	0
(x_2, x_3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2, x_3)	**33**	**13**	**33**	**84**	**90**	**84**	**84**	**100**	**84**	**14**	**3**	**14**	**50**	**14**	**50**	**81**	**67**	**81**
(x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_4)	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
(x_2, x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2, x_4)	10	2	10	0	0	0	0	0	0	11	2	11	15	3	15	6	4	6
(x_3, x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_3, x_4)	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
(x_2, x_3, x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2, x_3, x_4)	7	0	7	15	0	15	16	0	16	4	0	5	7	0	7	13	0	13
$\boldsymbol{\beta} = (-1, 3, 2, -1.5, 1)$, signal-to-noise ratios are 1.84 for independent and 3.33 for dependent.																		
none	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1)	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
(x_2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2)	9	40	9	0	0	0	0	0	0	51	75	51	0	13	0	0	0	0
(x_3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_3)	0	0	0	0	0	0	0	0	0	4	6	4	0	0	0	0	0	0
(x_2, x_3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2, x_3)	6	6	6	0	0	0	0	0	0	7	1	7	0	0	0	0	0	0
(x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_4)	0	0	0	0	0	0	0	0	0	4	10	4	0	0	0	0	0	0
(x_2, x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2, x_4)	50	47	50	0	9	0	0	0	0	24	4	25	51	80	51	11	65	11
(x_3, x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_3, x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_2, x_3, x_4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(x_1, x_2, x_3, x_4)	**34**	**7**	**34**	**100**	**91**	**100**	**100**	**100**	**100**	**10**	**1**	**10**	**48**	**7**	**48**	**89**	**35**	**89**

3.6.2 Estimating Equations

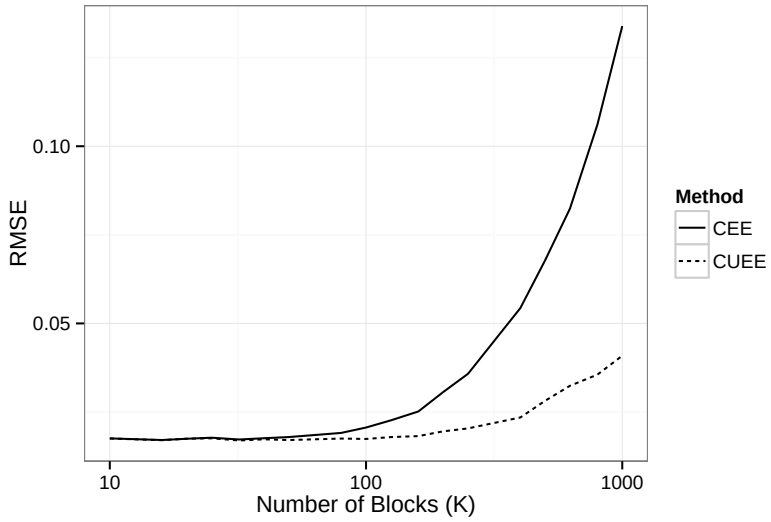

Figure 2: RMSE of CEE and CUEE estimators for different numbers of blocks.

Logistic Regression

To examine the effect of the total number of blocks K on the performance of the CEE and CUEE estimators, we generated $y_i \sim \text{Bernoulli}(\mu_i)$, independently for $i = 1, \ldots, 100000$, with $\text{logit}(\mu_i) = \mathbf{x}_i^\top \boldsymbol{\beta}$ where $\boldsymbol{\beta} = (1, 1, 1, 1, 1, 1)^\top$, $x_{i[2:4]} \sim \text{Bernoulli}(0.5)$ independently, $x_{i[5:6]} \sim N(\mathbf{0}, \mathbf{I}_2)$ independently, and $x_{ki[1]} = 1$. The total sample size was fixed at $N = 100,000$, but in computing the CEE and CUEE estimates, the number of blocks K varied from 10 to $1,000$ where N could be divided evenly by K. At each value of K, the root-mean square error (RMSE) of both the CEE and CUEE estimators were calculated as $\sqrt{\frac{\sum_{j=1}^{6}(\breve{\beta}_{Kj}-1)^2}{6}}$, where $\breve{\beta}_{Kj}$ represents the j^{th} coefficient in either the CEE or CUEE terminal estimate. The averaged RMSEs are obtained with 200 replicates.

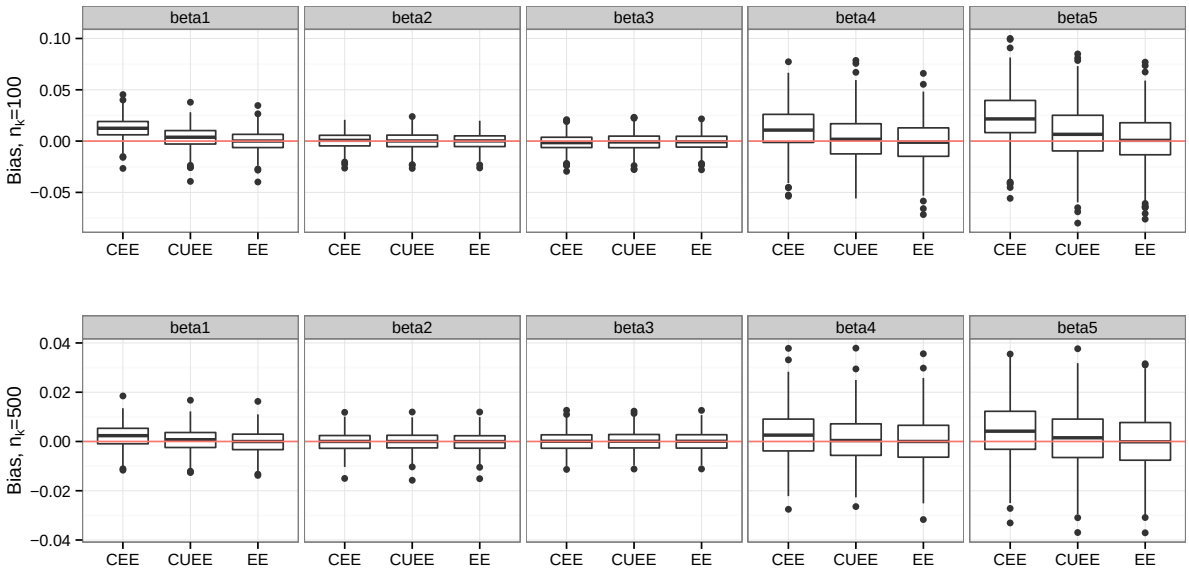

Figure 3: Boxplots of biases for CEE, CUEE, EE estimators of β_j (estimated β_j - true β_j), $j = 1, \ldots, 5$, for varying n_k.

Figure 2 shows the plot of averaged RMSEs versus the number of blocks K. It is clear that as the number of blocks increases (block size decreases), RMSE from CEE method increases very fast while RMSE from the CUEE method remains relatively stable.

Robust Poisson Regression

In these simulations, we compared the performance of the (terminal) CEE and CUEE estimators with the EE estimator based on all of the data. We generated $B = 500$ datasets of $y_i \sim \text{Poisson}(\mu_i)$, independently for $i = 1, \ldots, N$ with $\log(\mu_i) = \mathbf{x}_i^\top \boldsymbol{\beta}$ where $\boldsymbol{\beta} = (0.3, -0.3, 0.3, -0.3, 0.3)^\top$, $x_{ki[1]} = 1$, $x_{i[2:3]} \sim N(\mathbf{0}, \mathbf{I}_2)$ independently, $x_{i[4]} \sim Bernoulli(0.25)$ independently, and $x_{i[5]} \sim Bernoulli(0.1)$ independently. We fixed $K = 100$, but varied $n_k = n \in \{100, 500\}$.

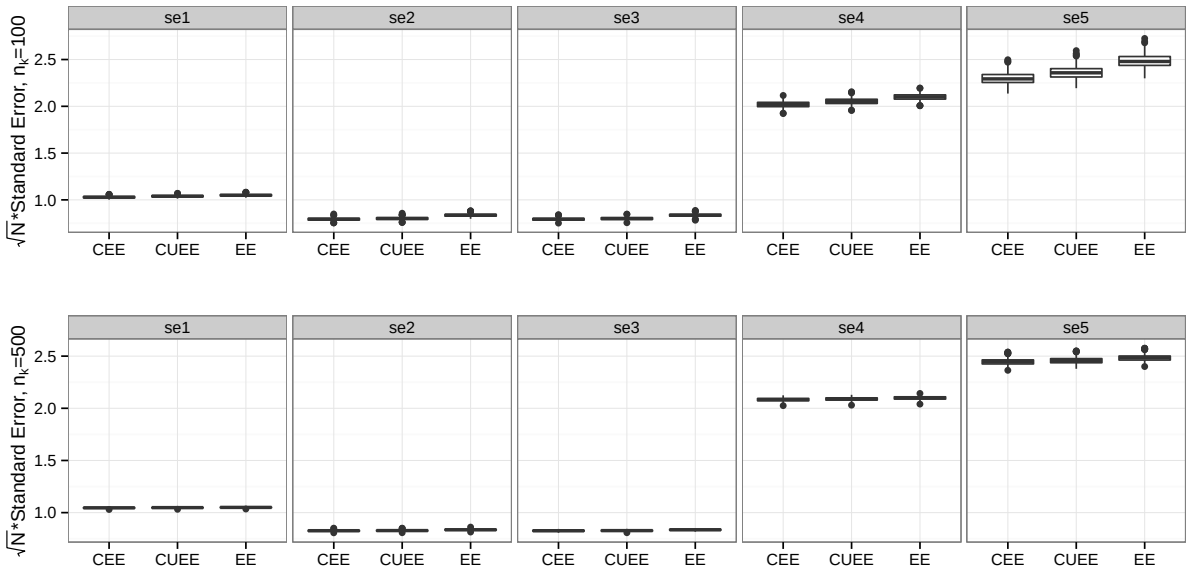

Figure 4: Boxplots of standard errors CEE, CUEE, EE estimators of β_j, $j = 1, \ldots, 5$, for varying n_k. Standard errors have been multiplied by $\sqrt{Kn_k} = \sqrt{N}$ for comparability.

Figure 3 shows boxplots of the biases in the 3 types of estimators (CEE, CUEE, EE) of β_j, $j = 1, \ldots, 5$, for varying n_k. The CEE estimator tends to be the most biased, particularly in the intercept, but also in the coefficients corresponding to binary covariates. The CUEE estimator also suffers from slight bias, while the EE estimator performs quite well, as expected. Also as expected, as n_k increases, bias decreases. The corresponding robust (sandwich-based) standard errors are shown in Figure 4, but the results were very similar for variances estimated by $\mathbf{A}_K^{-1}$ and $\tilde{\mathbf{A}}_K^{-1}$. In the plot, as n_k increases, the standard errors become quite similar for the three methods.

Table 6 shows the RMSE ratios, RMSE(CEE)/RMSE(EE) and RMSE(CUEE)/RMSE(EE), for each coefficient. The RMSE ratios for CEE and CUEE estimators confirm the boxplot results as the intercept and the coefficients corresponding to binary covariates (β_4,

Table 6: Root Mean Square Error Ratios of CEE and CUEE with EE

		β_1	β_2	β_3	β_4	β_5
$n_k = 100$	CEE	2.414	1.029	1.036	1.299	1.810
	CUEE	1.172	1.092	1.088	1.118	1.205
$n_k = 500$	CEE	1.225	1.002	1.002	1.060	1.146
	CUEE	0.999	1.010	1.016	0.993	1.057

β_5) tend to be the most problematic for both estimators, but more so for the CEE estimator.

For this particular simulation, it appears $n_k = 500$ is sufficient to adequately reduce the bias. However, the appropriate subset size n_k, if given the choice, is relative to the data at hand. For example, if we alter the data generation of the simulation to instead have $x_{i[5]} \sim Bernoulli(0.01)$ independently, but keep all other simulation parameters the same, the bias, particularly for β_5, still exists at $n_k = 500$ (see Figure 5) but diminishes substantially with $n_k = 5000$.

3.7 Airline Data Analysis

The same airline on-time statistics as described in Section 2.4 is investigated here.

We first used logistic regression to model the probability of late arrival (binary; 1 if late by more than 15 minutes, 0 otherwise) as a function of departure time (continuous); distance (continuous, in thousands of miles), day/night flight status (binary; 1 if departure between 8pm and 5am, 0 otherwise); weekend/weekday status (binary; 1 if departure occurred during the weekend, 0 otherwise), and distance type (categorical;

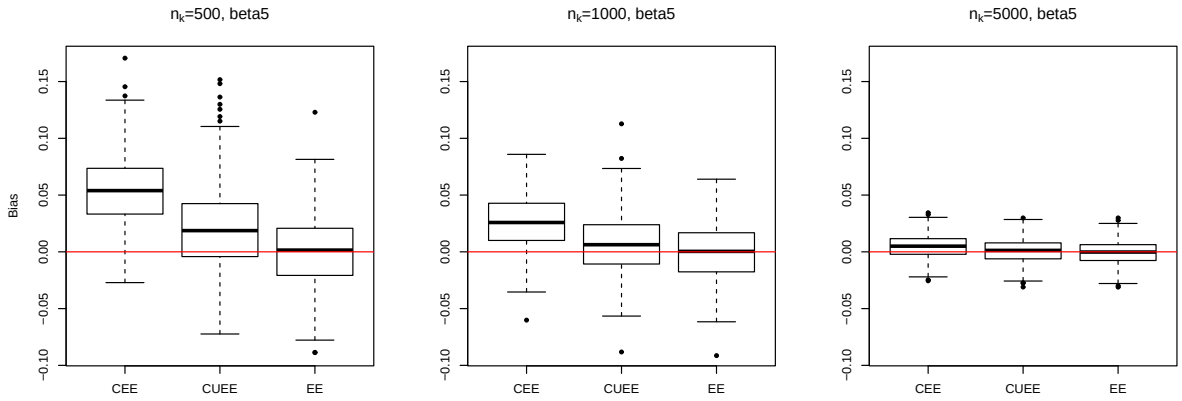

Figure 5: Boxplots of biases for 3 types of estimators (CEE, CUEE, EE) of β_5 (estimated β_5 - true β_5), for varying n_k, when $x_{i[5]} \sim Bernoulli(0.01)$.

'typical distance' for distances less than 4200 miles, the reference level 'large distance' for distances between 4200 and 4300 miles, and 'extreme distance' for distances greater than 4300 miles) for $N = 120,748,239$ observations with complete data.

For CEE and CUEE, we used a subset size of $n_k = 50,000$ for $k = 1, \ldots, K-1$, and $n_K = 48239$ to estimate the data in the online-updating framework. However, to avoid potential data separation problems due to rare events (extreme distance; 0.021% of the data with 26,021 observations), a detection mechanism has been introduced at each block. If such a problem exists, the next block of data will be combined until the problem disappears. We also computed EE estimates and standard errors using the commercial software Revolution R.

All three methods agree that all covariates except extreme distance are highly associated with late flight arrival ($p < 0.00001$), with later departure times and longer distances corresponding to a higher likelihood for late arrival, and night-time and weekend

flights corresponding to a lower likelihood for late flight arrival (see Table 7). However, extreme distance is not associated with the late flight arrival ($p = 0.613$). The large p value also indicates that even if number of observations is huge, there is no guarantee that all covariates must be significant. As we do not know the truth in this real data example, we compare the estimates and standard errors of CEE and CUEE with those from Revolution R, which computes the EE estimates, but notably not in an online-updating framework. In Table 7, the CUEE and Revolution R regression coefficients tend to be the most similar. The regression coefficient estimates and standard errors for CEE are also close to those from Revolution R, with the most discrepancy in the regression coefficients again appearing in the intercept and coefficients corresponding to binary covariates.

We finally considered arrival delay ($ArrDelay$) as a continuous variable by modeling $\log(ArrDelay - min(ArrDelay) + 1)$ as a function of departure time, distance, day/night flight status, and weekend/weekday flight status for United Airline flights ($N = 13,299,817$), and applied the global predictive residual outlier tests discussed in Section 3.2.3. Using only complete observations and setting $n_k = 1000$, $m = 3$, and $\alpha = 0.05$, we found that the normality-based F test in (3.9) and asymptotic F test in (3.10) overwhelmingly agreed upon whether or not there was at least one outlier in a given subset of data (96% agreement across $K = 12803$ subsets). As in the simulations, the normality-based F test rejects more often than the asymptotic F test: in the 4% of

Table 7: Estimates and standard errors ($\times 10^5$) from the Airline On-Time data for EE (computed by Revolution R), CEE, and CUEE estimators.

	EE		CEE		CUEE	
	$\hat{\boldsymbol{\beta}}_N$	$\mathsf{SE}(\hat{\boldsymbol{\beta}}_N)$	$\hat{\boldsymbol{\beta}}_K$	$\mathsf{SE}(\hat{\boldsymbol{\beta}}_K)$	$\tilde{\boldsymbol{\beta}}_K$	$\mathsf{SE}(\tilde{\boldsymbol{\beta}}_K)$
Intercept	−3.8680	1395.65	−3.7060	1434.60	−3.8801	1403.49
Depart	0.1040	6.01	0.1024	6.02	0.1017	5.70
Distance	0.2409	40.89	0.2374	41.44	0.2526	38.98
Night	−0.4484	81.74	−0.4318	82.15	−0.4335	80.72
Weekend	−0.1769	54.13	−0.1694	54.62	−0.1779	53.95
TypDist	0.8785	1389.11	0.7676	1428.26	0.9231	1397.46
ExDist	−0.0103	2045.71	−0.0405	2114.17	−0.0093	2073.99

subsets in which the two tests did not agree, the normality-based F test alone identified 488 additional subsets with at least one outlier, while the asymptotic F test alone identified 23 additional subsets with at least one outlier.

Chapter 4

Online Updating Algorithm with New Variables

4.1 Introduction

In a typical regression setting, emergence of new variables is common, due to, for example, negligence in data collection in the past, change of protocol, or advances in information technology. Most current methods for big data such as subsampling approach and divide-and-conquer approach can not be applied to such condition directly. One possible solution is to treat information of the new variable from early time as missing data and modern missing data techniques can be combined with the big data methods. Nevertheless, previous data are not saved according to the online updating method, so the naive online updating analyses would start from the scratch, discarding the possibly useful information contained in the existing data and, hence, loosing efficiency in statistical inferences. In this chapter, we propose a modification to the online

updating algorithm for the added variable situation. When new variables become available, under that assumption that the true model contains the new variables, the previous cumulated estimates for the coefficients of existing variables are biased. To make good use of the existing information, we correct the bias in the cumulative coefficient estimates and variance estimates; the online updating process resumes from the next block. Section 4.2 focuses on the linear model setting and introduces the correction formula along with the associated online updating algorithm, and illustrate the benefit of our method in a three-variable-and-two-block case under some mild conditions. Section 4.3 generalizes the method to the GLM with a unified estimating equations approach. Simulation results are reported in Section 4.4. A case study on the airline data is presented in Section 4.5.

4.2 Linear Model

4.2.1 Challenges from New Covariates

Recall from Section 3.2 that under online updating setting, when data are arriving in blocks $k = 1, 2, \cdots$, the linear regression model is

$$y_{ki} = \mathbf{x}_{ki}^{\top}\mathbf{b} + \epsilon_{ki}, \tag{4.1}$$

where $\mathbf{b}$ is a p-dimensional vector of regression coefficients, ϵ_{ki} has mean 0 and standard deviation σ_ϵ independently. We use $\mathbf{b}$ instead of $\boldsymbol{\beta}$ here to distinguish it from the true coefficient of $\mathbf{X}$ in Equation (4.4) later. The least squares (LS) estimate of $\mathbf{b}_k$ and σ_ϵ^2 are

$$\hat{\mathbf{b}}_k = (\mathbf{X}_k^\top \mathbf{X}_k)^{-1}\mathbf{X}_k^\top \mathbf{y}_k \quad \text{and} \quad \hat{\sigma}^2_{\epsilon,n_k,k} = \frac{1}{n_k - p}\mathbf{y}_k^\top(\mathbf{I}_{n_k} - \mathbf{H}_k)\mathbf{y}_k, \tag{4.2}$$

At the k^{th} block, the cumulative coefficient estimator of $\mathbf{b}$ is

$$\tilde{\mathbf{b}}_k = (\mathbf{X}_k^\top \mathbf{X}_k + \mathbf{V}_{k-1})^{-1}(\mathbf{X}_k^\top \mathbf{X}_k \hat{\mathbf{b}}_k + \mathbf{V}_{k-1}\tilde{\mathbf{b}}_{k-1}) \tag{4.3}$$

where $\tilde{\mathbf{b}}_0 = \mathbf{0}$, $\mathbf{V}_0 = \mathbf{0}_p$, a $p \times p$ matrix of zeros and $\mathbf{V}_k = \sum_{\ell=1}^k \mathbf{X}_\ell^\top \mathbf{X}_\ell$ for $k = 1, 2, \ldots$. This can be viewed as an average of $\tilde{\mathbf{b}}_{k-1}$ and $\hat{\mathbf{b}}_k$ weighted by the inverses of their variance matrix

$$\tilde{\mathbf{b}}_k = \big(\mathsf{Var}^{-1}(\hat{\mathbf{b}}_k) + \mathsf{Var}^{-1}(\tilde{\mathbf{b}}_{k-1})\big)^{-1}\big(\mathsf{Var}^{-1}(\hat{\mathbf{b}}_k)\hat{\mathbf{b}}_k + \mathsf{Var}^{-1}(\tilde{\mathbf{b}}_{k-1})\tilde{\mathbf{b}}_{k-1}\big).$$

Suppose that covariate $\mathbf{X}_k$ is available for each block $k = 1, 2, \ldots$, but a new set of covariates $\mathbf{Z}_k = (\mathbf{z}_{k1}, \ldots, \mathbf{z}_{kn_k})^\top$ become available only after K blocks, where $\mathbf{z}_{ki}$ is a $q \times 1$ covariate vector. That is $\mathbf{Z}_k$ is only observed for $k = K+1, K+2, \ldots$. The new design matrix $(\mathbf{X}_k, \mathbf{Z}_k)$ is assumed to have full column rank $p+q$. Suppose that, instead

of (4.1), the true regression model is

$$y_{ki} = \mathbf{x}_{ki}^\top \boldsymbol{\beta} + \mathbf{z}_{ki}^\top \boldsymbol{\theta} + \nu_{ki}, \quad i = 1, 2, \ldots, n_k, \tag{4.4}$$

where ν_{ki} has mean 0 and standard deviation σ_ν, $\boldsymbol{\beta}$ is a $p \times 1$ coefficient vector of $\mathbf{X}_k$, and $\boldsymbol{\theta}$ is a $q \times 1$ coefficient vector of $\mathbf{Z}_k$, for all $k = 1, 2, \ldots$.

Let H_R be the reduced model (4.1) and H_F the full model (4.4). Under true model H_F, the updated estimate $\tilde{\mathbf{b}}_K$ from the first K blocks has not been estimating $\boldsymbol{\beta}$ in general. It is an biased estimator for β with omitted variables in standard texts (e.g., Greene, 2003, p.148). Let $(\boldsymbol{\mathcal{X}}_K, \boldsymbol{\mathcal{Z}}_K)$ be the cumulative covariate until block K, where $\boldsymbol{\mathcal{Z}}_K$ is unobserved. Define $\boldsymbol{\delta} = \lim_{N_K \to \infty} (\boldsymbol{\mathcal{X}}_K^\top \boldsymbol{\mathcal{X}}_K)^{-1} \boldsymbol{\mathcal{X}}_K^\top \boldsymbol{\mathcal{Z}}_K \boldsymbol{\theta}$, and $\mathbf{b} = \boldsymbol{\beta} + \boldsymbol{\delta}$. Then $\mathbf{b}$ is the limit of $\tilde{\mathbf{b}}_K$ as $N_K \to \infty$. With the new data on $\mathbf{Z}$ available starting from block $K+1$, a naive approach would be to start the updating process from scratch at block $K+1$, throwing away the information from earlier blocks. Note that at block $K+1$, we may obtain $\hat{\boldsymbol{\beta}}_{K+1}$ and $\hat{\boldsymbol{\theta}}_{K+1}$ from fitting H_F, as well as $\hat{\mathbf{b}}_{K+1}$ from fitting H_R. With H_F being the true model, $\hat{\boldsymbol{\beta}}_{K+1}$ and $\hat{\boldsymbol{\theta}}_{K+1}$ are unbiased estimators of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, respectively, and $\hat{\mathbf{b}}_{K+1}$ is an unbiased estimator of $\mathbf{b}$.

Our goal is to fix the online updating at block $K+1$ such that the algorithm can be carried over as before after this block. For $\boldsymbol{\theta}$, since there is no previous data for $\mathbf{Z}$, its estimate can only be obtained from current block $K+1$: $\tilde{\boldsymbol{\theta}}_{K+1} = \hat{\boldsymbol{\theta}}_{K+1}$. For $\boldsymbol{\beta}$, however, it is more challenging. We propose to 1) use information from block $K+1$ to

obtain $\tilde{\boldsymbol{\beta}}_K$ and its variance; and 2) update from $\tilde{\boldsymbol{\beta}}_K$ to $\tilde{\boldsymbol{\beta}}_{K+1}$ and compute its covariance with $\tilde{\boldsymbol{\theta}}_{K+1}$. We will show that the cumulative estimate $\tilde{\boldsymbol{\beta}}_{K+1}$ is still unbiased for $\boldsymbol{\beta}$ and the corresponding variance $\mathsf{Var}(\tilde{\boldsymbol{\beta}}_{K+1})$ is smaller than $\mathsf{Var}(\hat{\boldsymbol{\beta}}_{K+1})$, where all previous information has been discarded naively. The online update procedure (4.3) goes on with $(\mathbf{X}, \mathbf{Z})$ in place of $\mathbf{X}$, and $(\boldsymbol{\beta}^\top, \boldsymbol{\theta}^\top)^\top$ in place of $\mathbf{b}$.

4.2.2 Updating at the Changing Block

The bias of the online updating estimator $\tilde{\mathbf{b}}_K$ up to block K in estimating $\boldsymbol{\beta}$ is $\boldsymbol{\delta} = \mathbf{b} - \boldsymbol{\beta}$. This bias can be estimated using data in block $K+1$ as $\hat{\boldsymbol{\delta}}_{K+1} = \hat{\mathbf{b}}_{K+1} - \hat{\boldsymbol{\beta}}_{K+1}$. This unbiased estimator of $\boldsymbol{\delta}$ from block $K+1$ can be used to correct the bias of $\tilde{\mathbf{b}}_K$, such that

$$\tilde{\boldsymbol{\beta}}_K = \tilde{\mathbf{b}}_K - \hat{\boldsymbol{\delta}}_{K+1} \tag{4.5}$$

is an unbiased estimator of $\boldsymbol{\beta}$. Using weighted average as in (4.3), the updated estimate at block $K+1$ is

$$\tilde{\boldsymbol{\beta}}_{K+1} = \big(\mathsf{Var}^{-1}(\tilde{\boldsymbol{\beta}}_K) + \mathsf{Var}^{-1}(\hat{\boldsymbol{\beta}}_{K+1})\big)^{-1}\big[\mathsf{Var}^{-1}(\tilde{\boldsymbol{\beta}}_K)\tilde{\boldsymbol{\beta}}_K + \mathsf{Var}^{-1}(\hat{\boldsymbol{\beta}}_{K+1})\hat{\boldsymbol{\beta}}_{K+1}\big], \tag{4.6}$$

where $\mathsf{Var}(\hat{\boldsymbol{\beta}}_{K+1})$ can be estimated from block $K+1$, but $\mathsf{Var}(\tilde{\boldsymbol{\beta}}_K)$ needs extra care. As the blocks are mutually independent, we have

$$\mathsf{Var}(\tilde{\boldsymbol{\beta}}_K) = \mathsf{Var}(\tilde{\mathbf{b}}_K - \hat{\boldsymbol{\delta}}_{K+1}) = \mathsf{Var}(\tilde{\mathbf{b}}_K) + \mathsf{Var}(\hat{\boldsymbol{\delta}}_{K+1}). \tag{4.7}$$

Both terms should be evaluated under H_F, despite that $\tilde{\mathbf{b}}_K$ and $\hat{\mathbf{b}}_{K+1}$ in $\hat{\boldsymbol{\delta}}_{K+1}$ are obtained under H_R.

The first component in Equation (4.7), $\mathsf{Var}(\tilde{\mathbf{b}}_K)$ is solved by Clogg et al. (1995) and Allison (1995) in comparing regression coefficients between nested models. Clogg et al. (1995) assume that $\mathbf{X}$ and $\mathbf{Z}$ are fixed, while Allison (1995) regards them as random as we do. Let $\mathbf{H}_{K+1} = (\mathbf{X}_{K+1}^\top \mathbf{X}_{K+1})^{-1}\mathbf{X}_{K+1}^\top \mathbf{Z}_{K+1}$, thus by Allison (1995),

$$\begin{aligned}\mathsf{Var}(\hat{\boldsymbol{\delta}}_{K+1}) &= \mathsf{E}[\mathsf{Var}(\hat{\boldsymbol{\delta}}_{K+1}|\mathbf{X}_{K+1}, \mathbf{Z}_{K+1})] + \mathsf{Var}[\mathsf{E}(\hat{\boldsymbol{\delta}}_{K+1}|\mathbf{X}_{K+1}, \mathbf{Z}_{K+1})] \\ &= \mathsf{E}[\mathbf{H}_{K+1}\mathsf{Var}(\hat{\boldsymbol{\theta}}_{K+1})\mathbf{H}_{K+1}^\top] + \boldsymbol{\theta}^\top \Omega \boldsymbol{\theta} \mathsf{E}[(\mathbf{X}_{K+1}^\top \mathbf{X}_{K+1})^{-1}],\end{aligned}$$

where Ω contains the variance and covariance when regressing $\mathbf{Z}$ on $\mathbf{X}$. The variance of $\hat{\boldsymbol{\delta}}_{K+1}$ is estimated by

$$\widehat{\mathsf{Var}}(\hat{\boldsymbol{\delta}}_{K+1}) = \mathbf{H}_{K+1}\widehat{\mathsf{Var}}(\hat{\boldsymbol{\theta}}_{K+1})\mathbf{H}_{K+1}^\top + \hat{\boldsymbol{\theta}}_{K+1}^\top \mathbf{W}_{K+1}\hat{\boldsymbol{\theta}}_{K+1}(\mathbf{X}_{K+1}^\top \mathbf{X}_{K+1})^{-1}, \tag{4.8}$$

where $\mathbf{W}_{K+1}$ is a matrix of variances and covariances of the residuals obtained from regressing $\mathbf{Z}_{K+1}$ on $\mathbf{X}_{K+1}$.

Similarly, we can express $\mathsf{Var}(\tilde{\mathbf{b}}_K)$ as

$$\mathsf{Var}(\tilde{\mathbf{b}}_K) = \mathsf{E}[\mathsf{Var}(\tilde{\mathbf{b}}_K|\boldsymbol{\mathcal{X}}_K, \boldsymbol{\mathcal{Z}}_K)] + \mathsf{Var}[\mathsf{E}(\tilde{\mathbf{b}}_K|\boldsymbol{\mathcal{X}}_K, \boldsymbol{\mathcal{Z}}_K)],$$

pretending that $\mathbf{Z}_k$'s are available for $k = 1, \ldots, K$. The conditional variance in the first

component is shown to be $\mathsf{Var}(\tilde{\mathbf{b}}_K|H_R)\sigma^2_\nu/\sigma^2_\epsilon$, where $\mathsf{Var}(\tilde{\mathbf{b}}_K|H_R)$ is the variance of $\tilde{\mathbf{b}}_K$ under the assumption of the reduced model H_R (Clogg et al., 1995, p.1274). For the second term, $\mathsf{Var}[\mathsf{E}(\tilde{\mathbf{b}}_K|\boldsymbol{\mathcal{X}}_K, \boldsymbol{\mathcal{Z}}_K)] = \mathsf{Var}[\boldsymbol{\beta}+\boldsymbol{\mathcal{H}}_K\boldsymbol{\theta}_K] = \mathsf{Var}[\boldsymbol{\mathcal{H}}_K\boldsymbol{\theta}_K] = \boldsymbol{\theta}^\top\Omega\boldsymbol{\theta}\mathsf{E}[(\boldsymbol{\mathcal{X}}^\top_K\boldsymbol{\mathcal{X}}_K)^{-1}]$, where $\boldsymbol{\mathcal{H}}_K = (\boldsymbol{\mathcal{X}}^\top_K\boldsymbol{\mathcal{X}}_K)^{-1}\boldsymbol{\mathcal{X}}^\top_k\boldsymbol{\mathcal{Z}}_K$, the cumulative version of $\mathbf{H}_K$ up to block K. Thus

$$\mathsf{Var}(\tilde{\mathbf{b}}_K) = \mathsf{Var}(\tilde{\mathbf{b}}_K|H_R)\sigma^2_\nu/\sigma^2_\epsilon + \boldsymbol{\theta}^\top\Omega\boldsymbol{\theta}\mathsf{E}[(\boldsymbol{\mathcal{X}}^\top_K\boldsymbol{\mathcal{X}}_K)^{-1}],$$

Note that σ^2_ν cannot be estimated until block $K+1$ because $\mathbf{Z}_\kappa$ is not observed for $\kappa = 1, \ldots, K$. So the first term can be estimated by $\hat{\mathsf{Var}}(\tilde{\mathbf{b}}_K|H_R)\hat{\sigma}^2_{\nu,K+1}/\tilde{\sigma}^2_{\epsilon,K+1}$, where $\hat{\mathsf{Var}}(\tilde{\mathbf{b}}_K|H_R)$ is the cumulative variance estimate of $\tilde{\mathbf{b}}_K$ at block K under H_R, $\hat{\sigma}^2_{\nu,K+1}$ is the estimate of σ^2_ν from block $K+1$, and $\tilde{\sigma}^2_{\epsilon,K+1}$ is the cumulative estimate of σ^2_ϵ up to block $K+1$. Similarly, we estimate $\boldsymbol{\theta}$ by $\hat{\boldsymbol{\theta}}_{K+1}$ and Ω by $\mathbf{W}_{K+1}$ from block $K+1$. Since $\mathbf{X}$ is available in all blocks, we want to use as much information as we can to estimate $\mathsf{E}[(\boldsymbol{\mathcal{X}}^\top_K\boldsymbol{\mathcal{X}}_K)^{-1}]$. That is $(\boldsymbol{\mathcal{X}}^\top_{K+1}\boldsymbol{\mathcal{X}}_{K+1})^{-1}$, scaled by the ratio in sample size N_{K+1}/N_K. Therefore, an estimator of $\mathsf{Var}(\tilde{\mathbf{b}}_K)$ is

$$\widehat{\mathsf{Var}}(\tilde{\mathbf{b}}_K) = \widehat{\mathsf{Var}}(\tilde{\mathbf{b}}_K|H_R)\frac{\hat{\sigma}^2_{\nu,K+1}}{\tilde{\sigma}^2_{\epsilon,K+1}} + \frac{N_{K+1}}{N_K}\hat{\boldsymbol{\theta}}^\top_{K+1}\mathbf{W}_{K+1}\hat{\boldsymbol{\theta}}_{K+1}(\boldsymbol{\mathcal{X}}^\top_{K+1}\boldsymbol{\mathcal{X}}_{K+1})^{-1}.$$

At this point, every term in Equation (4.6) that is needed for updating the estimates to block $K+1$ is available. Note that computation of $\mathsf{Var}(\tilde{\boldsymbol{\beta}}_K)$ under the linear model is a special case of the one under the GLM by using influence functions in Section 4.3.

4.2.3 Continue Updating with New Variables

In order to continue updating for future blocks, the variance-covariance matrix of the updated estimator $\tilde{\boldsymbol{\gamma}}_{K+1} = (\tilde{\boldsymbol{\beta}}_{K+1}, \tilde{\boldsymbol{\theta}}_{K+1})$ from the last subsection is needed. That is

$$\mathsf{Var}(\tilde{\boldsymbol{\gamma}}_{K+1}) = \mathsf{Var}\begin{pmatrix} \tilde{\boldsymbol{\beta}}_{K+1} \\ \tilde{\boldsymbol{\theta}}_{K+1} \end{pmatrix} = \begin{pmatrix} \mathsf{Var}(\tilde{\boldsymbol{\beta}}_{K+1}) & \mathsf{Cov}(\tilde{\boldsymbol{\beta}}_{K+1}, \tilde{\boldsymbol{\theta}}_{K+1}) \\ \mathsf{Cov}(\tilde{\boldsymbol{\beta}}_{K+1}, \tilde{\boldsymbol{\theta}}_{K+1})^\top & \mathsf{Var}(\tilde{\boldsymbol{\theta}}_{K+1}) \end{pmatrix}.$$

Obviously, $\mathsf{Var}(\tilde{\boldsymbol{\theta}}_{K+1})$ can be estimated from fitting the full model with data in block $K+1$. Detailed derivations for estimates of $\mathsf{Var}(\tilde{\boldsymbol{\beta}}_{K+1})$ and $\mathsf{Cov}(\tilde{\boldsymbol{\beta}}_{K+1}, \tilde{\boldsymbol{\theta}}_{K+1})$ are relegated to Appendix B.1. With these variance estimates available, the inverse variance estimate weighted online updating algorithm in Equation (4.3) can be continued with future blocks with $(\mathbf{X}, \mathbf{Z})$ in place of $\mathbf{X}$ and $\boldsymbol{\gamma}$ in place of $\mathbf{b}$.

4.2.4 The Three-Variable Case

We consider a simple case with $p = q = 1$ and $K = 1$. That is, covariates x and z are both scalars, z becomes available after block 1, and we want to compare the naive estimator $\hat{\beta}_2$ and the bias-corrected online updating estimator $\tilde{\beta}_2$ at block 2. Assume that x and z are centered and that the correlation coefficient between x and z is ρ_{xz}. Let σ_η^2 be the error variance in the regression of z on x. The ratio of the variance of the naive estimator $\hat{\beta}_2$ (throwing away block 1) to the variance of the corrected estimator

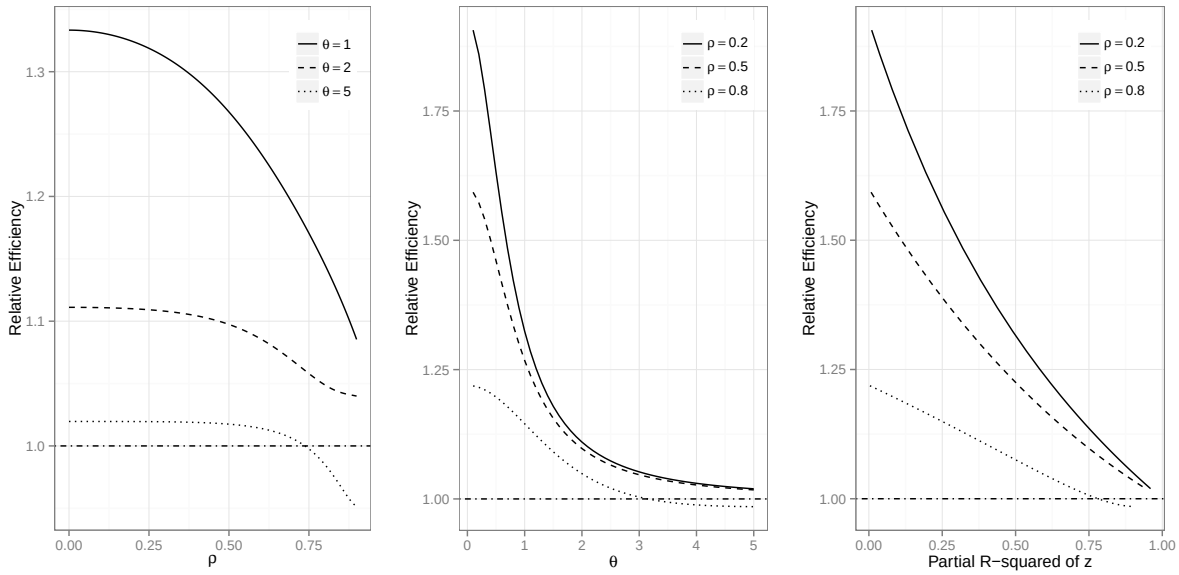

Figure 6: Under the linear model, fixing n_1/n_2 at 1, the relative efficiency of $\tilde{\beta}_2$ with respect to $\hat{\beta}_2$ decreases as either the correlation ρ_{xz} between x and z or θ increases. When both ρ_{xz} and θ are large, the relative efficiency can be less than 1.

$\tilde{\beta}_2$, or the relative efficiency of $\tilde{\beta}_2$ in comparison to $\hat{\beta}_2$ is

$$\mathsf{RE}(\tilde{\beta}_2, \hat{\beta}_2) = \frac{\mathsf{Var}(\hat{\beta}_2)}{\mathsf{Var}(\tilde{\beta}_2)} = \frac{(\frac{n_1}{n_2} + \Delta)^2}{\frac{n_1}{n_2}\Delta + \Delta^2 + 2\frac{n_1}{n_2}\Delta\rho_{xz}^2}, \tag{4.9}$$

where $\Delta = (1 - \rho_{xz}^2) + \theta^2(\sigma_\eta^2/\sigma_\nu^2)(1 - \rho_{xz}^2)(n_1/n_2) + \theta^2(\sigma_\eta^2/\sigma_\nu^2)(1 - \rho_{xz}^2) + \rho_{xz}^2(n_1/n_2)$.

See Appendix B.2 for detailed derivations.

It is interesting to see how the relative efficiency of $\tilde{\beta}_2$ with respect to $\hat{\beta}_2$ changes. When both x and z are standardized, $\sigma_\eta^2 = 1 - \rho_{xz}^2$. When σ_ν^2 is fixed, say at 1, $\sigma_\eta^2/\sigma_\nu^2 = 1 - \rho_{xz}^2$. Note that $1 - \rho_{xz}^2$ is the inverse of the variance inflation factor (VIF). Then, the relative efficiency becomes a function of three quantities: ρ_{xz}, θ, and n_1/n_2. Figure 6 shows the relative efficiency when $n_1/n_2 = 1$: the left panel shows

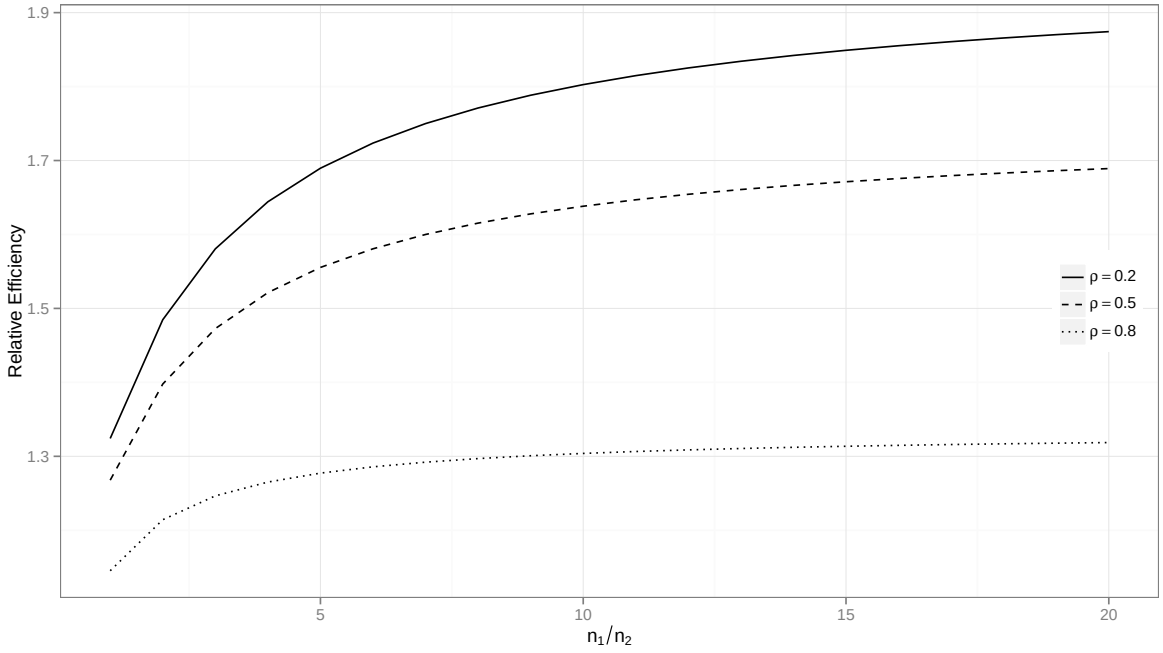

Figure 7: Under the linear model, fixing θ as 1, the relative efficiency of $\tilde{\beta}_2$ with respect to $\hat{\beta}_2$ increases as n_1/n_2 increases for different correlations ρ_{xz} between x and z. The smaller ρ_{xz} is, the higher relative efficiency.

it as a function of ρ_{xz} for $\theta \in \{1, 2, 5\}$; the middle panel show it as a function of θ for $\rho_{xz} \in \{0.2, 0.5, 0.8\}$. The relative efficiency of the corrected estimator decreases as either ρ_{xz} or θ increases, which are as expected because greater ρ_{xz} or θ suggests more importance of z. The relative efficiency is over 1 in most situations, but can be less than 1 when either θ or ρ_{xz} is sufficiently large. Alternatively, the importance of z can be measured as the partial R-squared of z given x in the model, which can be computed by $R^2_{y,z|x} = (\mathsf{SSE}(x) - \mathsf{SSE}(x, z))/\mathsf{SSE}(x)$, where $\mathsf{SSE}(x)$ is the SSE from the reduced model and $\mathsf{SSE}(x, z)$ is the SSE from the full model. Under the linear model setting, the partial R-squared has closed form. The right panel in Figure 6 shows the relative efficiency as a function of the partial R-squared of z given x. When z contributes more and more in the model, the relative efficiency decreases. Figure 7 shows the relative efficiency as a function of n_1/n_2 for $\rho_{xz} \in \{0.2, 0.5, 0.8\}$ with $\theta = 1$. As expected, the relative efficiency increases as n_1/n_2 increases.

Proposition 4.1. *Let* $\mathsf{RE}(\tilde{\beta}_2, \hat{\beta}_2)$ *be the relative efficiency of* $\tilde{\beta}_2$ *over* $\hat{\beta}_2$ *in the three-variable linear regression case, given in* (4.9).

1. *If* $\rho^2_{xz} \leq 0.5$, *then* $\mathsf{RE}(\tilde{\beta}_2, \hat{\beta}_2) > 1$.

2. *If* $0.5 < \rho^2_{xz} < 1$, *then* $\mathsf{RE}(\tilde{\beta}_2, \hat{\beta}_2) > 1$ *if*

$$\theta^2 < \frac{n_2(2\rho^2_{xz} - 1) - n_1(2\rho^2_{xz} + 1)}{(n_1 + n_2)(1 - 2\rho^2_{xz})} \frac{\sigma^2_\nu}{\sigma^2_\eta} = \frac{n_1}{n_1 + n_2} \frac{2\rho^2_{xz} + 1}{2\rho^2_{xz} - 1} \frac{\sigma^2_\nu}{\sigma^2_\eta} - \frac{n_2}{n_1 + n_2} \frac{\sigma^2_\nu}{\sigma^2_\eta}.$$

The proof is provided in Appendix B.2. When both covariates are standardized and the $\sigma_\nu^2 = 1$, if $0.5 < \rho_{xz}^2 < 1$, in order to see an efficiency gain in $\tilde{\beta}_2$, we need

$$\theta^2 < \frac{n_2(2\rho_{xz}^2 - 1) - n_1(2\rho_{xz}^2 + 1)}{(n_1 + n_2)(1 - 2\rho_{xz}^2)(1 - \rho_{xz}^2)}.$$

That is, when the x and z are highly correlated, the magnitude of θ needs to be sufficiently small to ensure an efficiency gain. Otherwise, throwing away the first block may be preferred. The closed form of $R_{y,z|x}^2$ under this simplified situation can be written as

$$R_{y,z|x}^2 = 1 - \frac{1}{\theta^2(1 - \rho_{xz}^2) + 1}.$$

It is obvious that the magnitude of θ has a positive relationship with $R_{y,z|x}^2$. It is reasonable because a larger θ indicates that z has stronger influence on y. Thus adding z into the model will increase the R-squared more, but as z becomes more important, the willing to utilize information of x from previous blocks becomes less, as shown in the middle and right panels of Figure 6.

4.3 Generalized Linear Model

4.3.1 Challenges from New Covariates

Recall from Section 3.3 that in a GLM setting with a known link function g, the reduced model H_R is

$$g(\mu_{ki,R}) = \eta_{ki,R} = \mathbf{x}_{ki}^{\top}\mathbf{b}$$

at each block k, where $\mu_{ki,R} = \mathsf{E}[y_{ki}|\mathbf{x}_{ki}]$. Schifano et al. (2016) proposed an online updating cumulative EE (CEE) estimator for $\mathbf{b}$. At block k, let $\mathbf{w}_{ki} = (y_{ki}, \mathbf{x}_{ki})$, $i = 1, ..., n_k$. Let $\hat{\mathbf{b}}_{n_k,k}$ denote the solution to EE

$$\mathbf{M}_{n_k,k}(\mathbf{b}) = \sum_{i=1}^{n_k} \psi(\mathbf{w}_{ki}, \mathbf{b}) = \mathbf{0},$$

and $\hat{\mathbf{V}}_{n_k,k}$ its corresponding variance estimate at block k. The CEE estimator $\tilde{\mathbf{b}}_k$ takes the form

$$\tilde{\mathbf{b}}_k = (\mathbf{A}_{k-1} + \mathbf{A}_{n_k,k})^{-1}(\mathbf{A}_{k-1}\tilde{\mathbf{b}}_{k-1} + \mathbf{A}_{n_k,k}\hat{\mathbf{b}}_{n_k,k}), \tag{4.10}$$

where

$$\mathbf{A}_{n_k,k} = -\sum_{i=1}^{n_k} \frac{\partial \psi(\mathbf{w}_{ki}, \hat{\mathbf{b}}_{n_k,k})}{\partial \mathbf{b}},$$

$\tilde{\mathbf{b}}_0 = \mathbf{0}$, $\mathbf{A}_0 = \mathbf{0}_p$, and $\mathbf{A}_k = \sum_{\ell=1}^{k} \mathbf{A}_{k_\ell,\ell} = \mathbf{A}_{k-1} + \mathbf{A}_{n_k,k}$. The cumulative variance estimator for $\tilde{\mathbf{b}}_k$ is

$$\tilde{\mathbf{V}}_k = (\mathbf{A}_{k-1} + \mathbf{A}_{n_k,k})^{-1}(\mathbf{A}_{k-1}\tilde{\mathbf{V}}_{k-1}\mathbf{A}_{k-1}^\top + \mathbf{A}_{n_k,k}\hat{\mathbf{V}}_{n_k,k}\mathbf{A}_{n_k,k}^\top)[(\mathbf{A}_{k-1} + \mathbf{A}_{n_k,k})^{-1}]^\top, \quad (4.11)$$

with $\tilde{\mathbf{V}}_0 = \mathbf{0}_p$ and $\mathbf{A}_0 = \mathbf{0}_p$. Specifically, if $\mathbf{M}_{n_k,k}(\mathbf{b})$ is the score equation under likelihood inference, then $\mathbf{A}_{n_k,k} = \mathbf{V}_{n_k,k}^{-1}$. Thus from Equation (4.11), we have $\mathbf{A}_k = \tilde{\mathbf{V}}_k^{-1}$, and Equation (4.10) becomes

$$\tilde{\mathbf{b}}_k = (\mathbf{V}_{k-1}^{-1} + \mathbf{V}_{n_k,k}^{-1})^{-1}(\mathbf{V}_{k-1}^{-1}\tilde{\mathbf{b}}_{k-1} + \mathbf{V}_{n_k,k}^{-1}\hat{\mathbf{b}}_{n_k,k}). \quad (4.12)$$

Equation (4.6) for the LM is a special case of Equation (4.12).

When a new set of covariates becomes available after block K, we assume that the correct, full model H_F should include them:

$$g(\mu_{ki,F}) = \eta_{ki,F} = \mathbf{x}_{ki}^\top\boldsymbol{\beta} + \mathbf{z}_{ki}^\top\boldsymbol{\theta}, \qquad i = 1, ..., n_k, \quad (4.13)$$

where $\mu_{ki,F} = \mathsf{E}(y_{ki}|\mathbf{x}_{ki}, \mathbf{z}_{ki})$. As in the LM case, we still assume that design matrix $(\mathbf{X}_k, \mathbf{Z}_k)$ is of full column rank $p + q$. This raises a question: if model (4.13) is the true model, what has $\tilde{\mathbf{b}}_K$ been estimating up to block K?

As the cumulative sample size up to block K, $N_K \to \infty$, $\tilde{\mathbf{b}}_K$ converges to some limit $\mathbf{b}$, which solves the EE $\mathbf{M}_{n_K,K}(\mathbf{b}) = 0$ as $n_K \to \infty$. In the LM case with identity

link function g, $\mathbf{b}$ has a closed-form expression in terms of $\boldsymbol{\beta}$ added by some bias. In the GLM case with a general link function g, we need to assume that $\mathbf{b}$ is the unique root of $\mathbf{M}_{n_K,K}(\mathbf{b})$ as $n_K \to \infty$. It is $\mathbf{b}$ that $\tilde{\mathbf{b}}_K$ has been estimating. Our strategy is the same as in the LM case. For $\boldsymbol{\theta}$, we start the updating with $\tilde{\boldsymbol{\theta}}_{K+1} = \hat{\boldsymbol{\theta}}_{K+1}$; for $\boldsymbol{\beta}$, we construct $\tilde{\boldsymbol{\beta}}_{K+1}$ by correcting the bias in $\tilde{\mathbf{b}}$ in estimating $\boldsymbol{\beta}$. The bias of $\hat{\mathbf{b}}$ in estimating $\boldsymbol{\beta}$ is $\boldsymbol{\delta} = \mathbf{b} - \boldsymbol{\beta}$. The bias can be estimated using the data in block $K+1$ as $\hat{\boldsymbol{\delta}}_{K+1} = \hat{\mathbf{b}}_{K+1} - \hat{\boldsymbol{\beta}}_{K+1}$.

4.3.2 Updating at the Changing Block

The correcting formulas (4.5)–(4.7) are still valid under GLM, but estimators of the terms need to be derived. We extend the influence functions approach for the LM used by Yan et al. (2013) to the GLM setting for these terms. The method could be further extended to handle clustered data with generalized estimating equations (GEE), and an implementation is available in R package `geepack` (Halekoh et al., 2006). Here we focus, however, on independent data. The quantities to estimate at the changing block are $\mathsf{Var}(\tilde{\mathbf{b}}_K)$ and $\mathsf{Var}(\hat{\boldsymbol{\delta}}_{K+1})$, both under the full model H_F.

Assuming both $\mathbf{X}$ and $\mathbf{Z}$ are available at block k, the EE for $\boldsymbol{\gamma} = (\boldsymbol{\beta}^\top, \boldsymbol{\theta}^\top)^\top$ under H_F is

$$\sum_{i=1}^{n_k} \mathbf{U}_{ki,F}(\boldsymbol{\gamma}) = \sum_{i=1}^{n_k} \mathbf{D}_{ki,F}^\top V_{ki,F}^{-1}\Big(y_{ki} - \mu_{ki,F}\Big) = \mathbf{0},$$

where $\mathbf{D}_{ki,F} = \partial \mu_{ki,F}/\partial \boldsymbol{\gamma}$ is a $(p+q) \times 1$ vector, $V_{(k+1)i,F}$ is the variance as a function

of $\mu_{ki,F}$. The estimator $\hat{\boldsymbol{\gamma}}_k$ has an asymptotic representation

$$\sqrt{n_k}(\hat{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}) = \frac{1}{\sqrt{n_k}} \sum_{i=1}^{n_k} \imath_{ki,F}(\boldsymbol{\gamma}) + o_p(1),$$

where $\imath_{ki,F}(\boldsymbol{\gamma}) = \Gamma_F(\boldsymbol{\gamma})\mathbf{U}_{ki,F}(\boldsymbol{\gamma})$, and $\Gamma_F(\boldsymbol{\gamma}) = \left[\lim_{n_k\to\infty} n_k^{-1} \sum_{i=1}^{n_k} \mathbf{D}_{ki,F}^\top V_{ki,F}^{-1} \mathbf{D}_{ki,F}\right]^{-1}$ is evaluated at the true value $\boldsymbol{\gamma}$. The $(p+q)\times 1$ vector $\imath_{ki}$'s are the influence functions of $\hat{\boldsymbol{\gamma}}$. Thus, $\sqrt{n_k}(\hat{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma})$ is asymptotically a normal distribution with mean $\mathbf{0}$ and variance-covariance matrix $\Gamma_F(\boldsymbol{\gamma})\Omega_F(\boldsymbol{\gamma})\Gamma_F(\boldsymbol{\gamma})$, where $\Omega_F(\boldsymbol{\gamma}) = \lim_{n_k\to\infty} n_k^{-1} \sum_{i=1}^{n_k} \mathbf{U}_{ki,F}\mathbf{U}_{ki,F}^\top$.

From the sample, we can estimate $\hat{\mathbf{D}}_{ki,F}$, $\hat{V}_{ki,F}$, and $\hat{U}_{ki,F}(\hat{\boldsymbol{\gamma}}_k)$. Thus $\Gamma_F(\boldsymbol{\gamma})$ can be estimated as $\hat{\Gamma}_{k,F}(\hat{\boldsymbol{\gamma}}_k) = \left[n_k^{-1} \sum_{i=1}^{n_k} \hat{\mathbf{D}}_{ki,F}^\top \hat{V}_{ki,F}^{-1} \hat{\mathbf{D}}_{ki,F}\right]^{-1}$, $\imath_{ki,F}(\boldsymbol{\gamma})$ can be estimated as $\hat{\imath}_{ki,F}(\hat{\boldsymbol{\gamma}}_k) = \hat{\Gamma}_{k,F}(\hat{\boldsymbol{\gamma}}_k)\hat{\mathbf{U}}_{ki,F}(\hat{\boldsymbol{\gamma}}_k)$, $\Omega_F(\boldsymbol{\gamma})$ can be estimated as $\hat{\Omega}_{k,F}(\hat{\boldsymbol{\gamma}}_k) = n_k^{-1} \sum_{i=1}^{n_k} \hat{\mathbf{U}}_{ki,F}\hat{\mathbf{U}}_{ki,F}^\top$, and the variance-covariance matrix can be estimated by $\widehat{\mathsf{Var}}(\hat{\boldsymbol{\gamma}}_k) = n_k^{-1}\hat{\Gamma}_{k,F}(\hat{\boldsymbol{\gamma}}_k)\hat{\Omega}_{k,F}(\hat{\boldsymbol{\gamma}}_k)\hat{\Gamma}_{k,F}(\hat{\boldsymbol{\gamma}}_k)$.

Similarly, under the reduced model H_R, the EE is

$$\sum_{i=1}^{n_k} \mathbf{U}_{ki,R} = \sum_{i=1}^{n_k} \mathbf{D}_{ki,R}^\top V_{ki,R}^{-1}\Big(y_{ki} - \mu_{ki,R}\Big) = \mathbf{0},$$

where $\mathbf{D}_{ki,R} = \partial\mu_{ki,R}/\partial\mathbf{b}$ is a $p \times 1$ vector, $V_{ki,R}$ is the variance as a function of $\mu_{ki,R}$, and subscript R is used to denote explicitly the corresponding quantities under H_R. Under the assumption that $\mathbf{b}$ is the solution to this EE as $n_k \to \infty$, the asymptotic

representation of the resulting estimator $\hat{\mathbf{b}}_k$ is

$$\sqrt{n_k}(\hat{\mathbf{b}}_k - \mathbf{b}) = \frac{1}{\sqrt{n_k}} \sum_{i=1}^{n_k} \imath_{ki,R}(\mathbf{b}) + o_p(1),$$

where $\imath_{ki,R}(\mathbf{b}) = \Gamma_R(\mathbf{b}) U_{ki,R}(\mathbf{b})$, and $\Gamma_R(\mathbf{b}) = \left[\lim_{n_k \to \infty} n_k^{-1} \sum_{i=1}^{n_k} \mathbf{D}_{ki,R}^\top V_{ki,R}^{-1} \mathbf{D}_{ki,R}\right]^{-1}$ is evaluated at $\mathbf{b}$. Thus, $\sqrt{n_\kappa}(\hat{\mathbf{b}}_\kappa - \mathbf{b})$ is asymptotically a normal distribution with mean $\mathbf{0}$ and variance-covariance matrix $\Gamma_R(\mathbf{b})\Omega_R(\mathbf{b})\Gamma_R(\mathbf{b})$, where $\Omega_R(\mathbf{b}) = \lim_{n_k \to \infty} n_k^{-1} \sum_{i=1}^{n_k} \mathbf{U}_{ki,R} \mathbf{U}_{ki,R}^\top$.

From the sample, we can estimate $\hat{\mathbf{D}}_{ki,R}$, $\hat{V}_{ki,R}$, and $\hat{U}_{ki,R}(\hat{\mathbf{b}}_k)$. $\Gamma_R(\mathbf{b})$ can be estimated as $\hat{\Gamma}_{k,R}(\hat{\mathbf{b}}_k) = \left[n_k^{-1} \sum_{i=1}^{n_k} \hat{\mathbf{D}}_{ki,R}^\top \hat{V}_{ki,R}^{-1} \hat{\mathbf{D}}_{ki,R}\right]^{-1}$, $\imath_{ki,R}(\mathbf{b})$ can be estimated as $\hat{\imath}_{ki,R}(\hat{\mathbf{b}}_k) = \hat{\Gamma}_{k,R}(\hat{\mathbf{b}}_k)\hat{\mathbf{U}}_{ki,R}(\hat{\mathbf{b}}_k)$, $\Omega_R(\mathbf{b})$ can be estimated as $\hat{\Omega}_{k,R}(\hat{\mathbf{b}}_k) = n_k^{-1} \sum_{i=1}^{n_k} \hat{\mathbf{U}}_{(k)i,R} \hat{\mathbf{U}}_{(k)i,R}^\top$, and the variance-covariance matrix can be estimated by $\widehat{\mathsf{Var}}(\hat{\mathbf{b}}_k) = n_k^{-1} \hat{\Gamma}_{k,R}(\hat{\mathbf{b}}_k) \hat{\Omega}_{k,R}(\hat{\mathbf{b}}_k) \hat{\Gamma}_{k,R}(\hat{\mathbf{b}}_k)$.

The influence functions in the asymptotic representation make the variance estimation of the needed quantities very easy. Notice that the influence functions of $\hat{\mathbf{b}}_k$, and, hence, its variance and variance estimator, are the same as those obtained assuming H_R is the true model because of the definition of $\mathbf{b}$. Therefore, the estimator of $\mathsf{Var}(\tilde{\mathbf{b}}_K)$ is the online updated $\widehat{\mathsf{Var}}(\tilde{\mathbf{b}}_K)$ at block K under H_R. The variance of $\hat{\boldsymbol{\delta}}_{K+1}$ is estimated by

$$\widehat{\mathsf{Var}}(\hat{\boldsymbol{\delta}}_{K+1}) = \frac{1}{n_{K+1}^2} \sum_{i=1}^{n_{K+1}} \hat{\jmath}_{(K+1)i,\delta}(\hat{\boldsymbol{\gamma}}_{K+1}, \hat{\mathbf{b}}_{K+1}) \hat{\jmath}_{(K+1)i,\delta}^\top(\hat{\boldsymbol{\gamma}}_{K+1}, \hat{\mathbf{b}}_{K+1}),$$

where $\hat{\jmath}_{(K+1)i,\delta}(\hat{\boldsymbol{\gamma}}_{K+1}, \hat{\mathbf{b}}_{K+1}) = \hat{\imath}_{(K+1)i,R}(\hat{\mathbf{b}}_{K+1}) - \hat{\imath}_{(K+1)i,F,\beta}(\hat{\boldsymbol{\gamma}}_{K+1})$, and $\hat{\imath}_{(K+1)i,F,\beta}(\hat{\boldsymbol{\gamma}}_{K+1})$ is the subvector of $\hat{\imath}_{(K+1)i,F}(\hat{\boldsymbol{\gamma}}_{K+1})$ containing its first p elements.

4.3.3 Continue Updating with New Variables

To continue updating with new variables, we adapt the derivations for the LM case in Appendix B.1 to find two key quantities, $\mathsf{Cov}(\hat{\mathbf{b}}_{K+1}, \hat{\boldsymbol{\beta}}_{K+1})$ and $\mathsf{Cov}(\hat{\mathbf{b}}_{K+1}, \hat{\boldsymbol{\theta}}_{K+1})$. Again, they can be easily estimated with the influence functions. First, we estimate $\mathsf{Cov}(\hat{\mathbf{b}}_{K+1}, \hat{\boldsymbol{\beta}}_{K+1})$ by

$$\widehat{\mathsf{Cov}}(\hat{\mathbf{b}}_{K+1}, \hat{\boldsymbol{\beta}}_{K+1}) = \frac{1}{n_{K+1}^2} \sum_{i=1}^{n_{K+1}} \hat{\imath}_{(K+1)i,R}(\hat{\mathbf{b}}_{K+1}) \hat{\imath}_{(K+1)i,F,\beta}^{\top}(\hat{\boldsymbol{\gamma}}_{K+1})$$

Further define $\hat{\imath}_{(K+1)i,F,\theta}(\hat{\boldsymbol{\gamma}}_{K+1})$ as the subvector of $\hat{\imath}_{(K+1)i,F}(\hat{\boldsymbol{\gamma}}_{K+1})$ containing its last q elements. Then, we estimate $\mathsf{Cov}(\hat{\mathbf{b}}_{K+1}, \hat{\boldsymbol{\theta}}_{K+1})$ by

$$\widehat{\mathsf{Cov}}(\hat{\mathbf{b}}_{K+1}, \hat{\boldsymbol{\theta}}_{K+1}) = \frac{1}{n_{K+1}^2} \sum_{i=1}^{n_{K+1}} \hat{\imath}_{(K+1)i,R}(\hat{\mathbf{b}}_{K+1}) \hat{\imath}_{(K+1)i,F,\theta}^{\top}(\hat{\boldsymbol{\gamma}}_{K+1}).$$

All the other derivations are the same as in the LM setting.

4.4 Simulation Study

We consider case where there are three variables (y, x, z) and two blocks of data, as in Section 4.2.4. Covariates x and z were generated from a bivariate normal distribution with standard normal margins and correlation $\rho_{xz} \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. The first data generating model is a LM with $y = \beta_0 + \beta_1 x + \theta z + \epsilon$, where $\beta_0 = \beta_1 = \theta = 1$ and ϵ follows a standard normal distribution. The second data generating model is a logistic

Table 8: Under linear or logistic models, average of standard errors ($\times 10$) of $\tilde{\beta}_2$ and $\tilde{\theta}_2$ and the correlation between them.

ρ_{xz}	$\mathsf{SE}(\tilde{\beta}_2)$		$\mathsf{SE}(\tilde{\theta}_2)$		$\mathsf{Corr}(\tilde{\beta}_2, \tilde{\theta}_2)$	
	Average	Empirical	Average	Empirical	Average	Empirical
Linear Model						
0.1	0.236	0.236	0.318	0.320	−0.134	−0.130
0.3	0.250	0.250	0.332	0.338	−0.400	−0.396
0.5	0.286	0.285	0.366	0.362	−0.640	−0.648
0.7	0.370	0.372	0.444	0.449	−0.839	−0.834
0.9	0.677	0.678	0.726	0.729	−0.966	−0.967
EE Approach under Logistic Model						
0.1	0.477	0.484	0.938	0.961	0.399	0.422
0.3	0.448	0.440	0.983	0.994	0.074	0.099
0.5	0.491	0.488	1.072	1.085	−0.371	−0.372
0.7	0.723	0.712	1.272	1.259	−0.751	−0.732
0.9	1.649	1.690	2.001	2.043	−0.940	−0.941

model with y generated from Bernoulli distributions with probability $\text{logit}^{-1}(\beta_0 + \beta_1 x + \theta z)$. We use the same parameter values as in the LM.

First we verify that the variance estimates of the updated estimator $(\tilde{\beta}_2, \tilde{\theta}_2)$ are consistent with their empirical variance matrix under LM or GLM. The sample sizes of the two blocks were set at $n_1 = 10,000$ and $n_2 = 1,000$, respectively. For each scenario, 5,000 replicates were generated. The averaged and empirical version of $\mathsf{SE}(\tilde{\beta}_2)$, $\mathsf{SE}(\tilde{\theta}_2)$, and $\mathsf{Corr}(\tilde{\beta}_2, \tilde{\theta}_2)$ are summarized in Table 8. A close agreement between the average and empirical SE is observed for the linear regression. For the logistic regression model, the agreement is not as tight as in the linear regression setting, but still generally good. One exception with a larger relative difference is for $\mathsf{Corr}(\tilde{\beta}_2, \tilde{\theta}_2)$ when the $\rho_{xz} = 0.3$, which

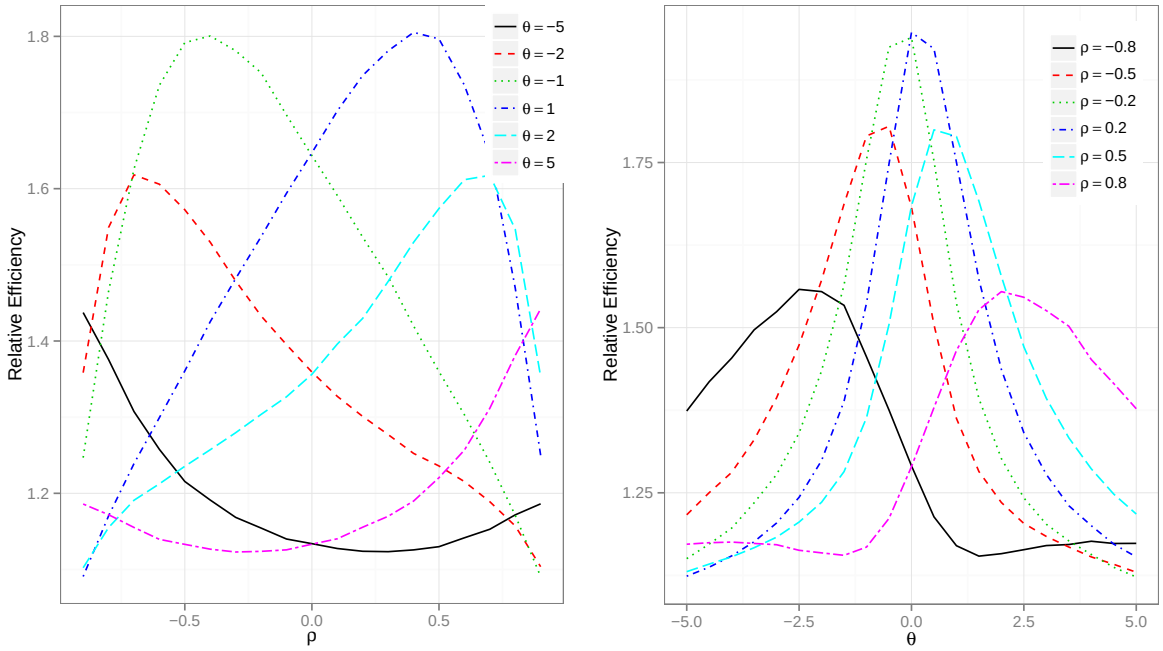

Figure 8: The empirical relative efficiency of $\tilde{\beta}_2$ with respect to $\hat{\beta}_2$ when ρ_{xz} and θ change under the logistic model, with $n_1 = n_2 = 1,000$.

might be explained by that $\mathsf{Corr}(\tilde{\beta}_2, \tilde{\theta}_2)$ in this scenario is indeed close to zero.

Next we study the efficiency gain of the bias-corrected cumulative estimators under the GLM settings relative to the naive estimator obtained from discarding previous data under different conditions. Unlike in the LM setting, there is no closed-form expression about the relative efficiency $\mathsf{RE}(\tilde{\beta}_2, \hat{\beta}_2)$. First, we set $n_1 = n_2 = 1,000$ and plot $\mathsf{RE}(\tilde{\beta}_2, \hat{\beta}_2)$, which was calculated using the empirical variances of the two estimators from 500 replicates, against ρ_{xz} and θ in Figure 8, which is an analog of Figure 6. The left panel shows the change in the relative efficiency as ρ_{xz} increases from -0.9 to 0.9 with $\theta \in \{-5, -2, -1, 1, 2, 5\}$. The curves for negative θ's and positive θ's are

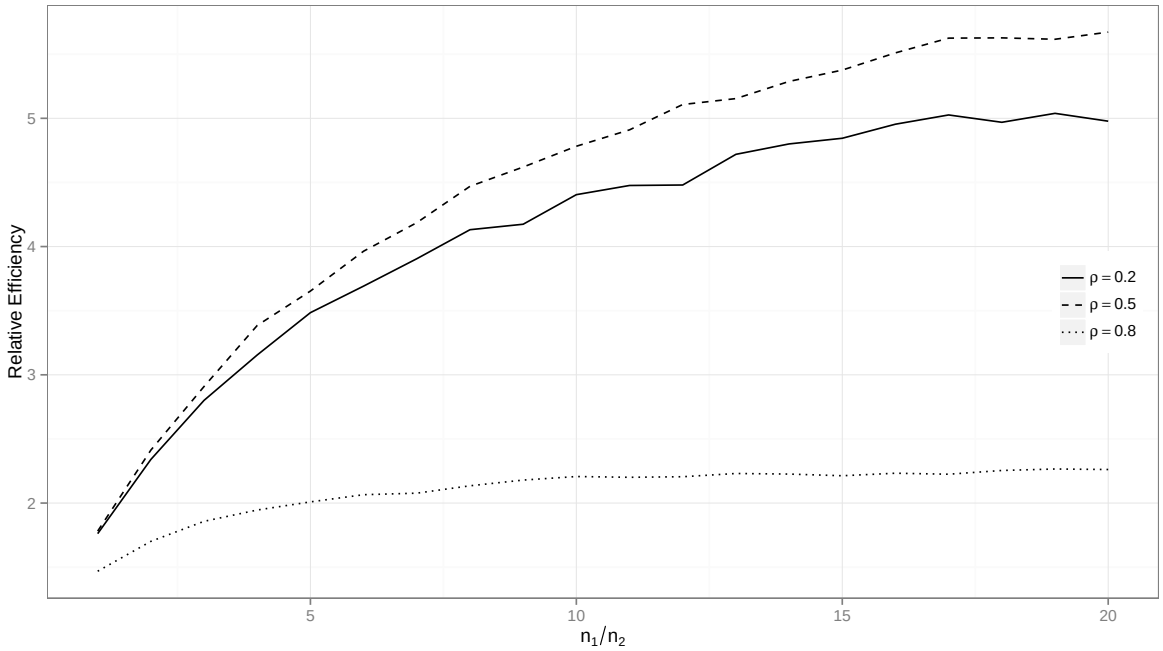

Figure 9: Under the logistic model, fixing θ as 1, the relative efficiency of $\tilde{\beta}_2$ with respect to $\hat{\beta}_2$ increases as n_1/n_2 increases for different correlations ρ_{xz} between x and z.

symmetric about $\rho_{xz} = 0$. All three curves are above 1 in the range shown. The pattern seems to be in agreement with the change of $\mathsf{Corr}(\tilde{\beta}_2, \tilde{\theta}_2)$ in Table 8. The closer to zero $\mathsf{Corr}(\tilde{\beta}_2, \tilde{\theta}_2)$ is, the larger the relative efficiency, which is expected. The right panel of Figure 8 shows the change in relative efficiency as θ varies from -5 to 5 for $\rho_{xz} \in \{-0.8, -0.5, -0.2, 0.2, 0.5, 0.8\}$. The six curves are symmetric to $\theta = 0$ and all have the increasing-first-then-decreasing trend. The peak of the curve is shifting to the left as ρ_{xz} decreases, and shifting to the right as ρ_{xz} increases. Again, all the curves are above 1 in the range shown.

Figure 9 is the analog of Figure 7 under the logistic model using empirical relative efficiencies. With $\theta = 1$ and $n_2 = 1,000$, the empirical relative efficiency $\mathsf{RE}(\tilde{\beta}_2, \hat{\beta}_2)$ obtained

from 100 replicates is shown as n_1/n_2 increases from 1 to 20 for $\rho_{xz} \in \{0.2, 0.5, 0.8\}$. All three curves are above 1, increasing as n_1/n_2 increases. The sample size appears to have a larger effect on $\rho_{xz} = 0.5$ which is consistent with the results from Figure 8. The magnitude in the relative efficiency here seems much greater than that in Figure 7 in the LM setting, suggesting that using the previous data blocks in the logistic model can be even more beneficial than in the LM setting.

4.5 Airline Data Analysis

In the same airline data as we used in previous chapters, since June 2003, the US Department of Transportation Bureau of Transportation Statistics began collecting the causes of flight delays. The categories of delays created by the Air Carrier On-Time Reporting Advisory Committee are defined as:

- Carrier: The cause of the cancellation or delay was due to circumstances within the airline's control.
- Weather: Significant meteorological conditions (actual or forecasted) that, in the judgment of the carrier, delays or prevents the operation of a flight such as tornado, blizzard or hurricane.
- National Aviation System (NAS): Delays and cancellations attributable to the national aviation system that refer to a broad set of conditions, such as non-extreme weather conditions, airport operations, heavy traffic volume, and air traffic

Table 9: In the linear model for the airline data analysis, y is the delay time, x is the distance, and the newly added variable z is one of the five types of delay. The variance ratio is defined as $\mathsf{Var}(\tilde{\boldsymbol{\beta}}_2)/\mathsf{Var}(\hat{\boldsymbol{\beta}}_2)$.

	ρ_{xz}	z-Statistics	Variance Ratio
Security	0.004	48.52	0.49
Weather	−0.006	231.79	0.53
NAS	0.016	414.26	0.61
LateAircraft	−0.015	484.90	0.65
Carrier	0.017	538.63	0.67

control.

- LateAircraft: A previous flight with same aircraft arrived late, causing the present flight to depart late.

- Security: Delays or cancellations caused by evacuation of a terminal or concourse, re-boarding of aircraft because of security breach, inoperative screening equipment and/or long lines in excess of 29 minutes at screening areas.

These five variables that were not available until June 2003 are apparently important in predicting flight delays. Two models were considered: a linear regression for the arrival delay in minutes and a logistic model for whether or not the arrival delay is longer than 15 minutes.

For the linear regression, we first consider the three-variable case where x is the distance in thousand of miles, and added each of the delay causes as z into the model with data of May–June, 2003. Table 9 shows the correlation coefficient of x and z, the z-statistics for the newly added variable z, and the ratio of the variance estimate of $\tilde{\boldsymbol{\beta}}_2$

Table 10: In the multiple linear regressions where y is the delay time, x is the distance, types of delay are added into the model one by one as new variables. The variance ratio is defined as $\mathsf{Var}(\tilde{\boldsymbol{\beta}}_2)/\mathsf{Var}(\hat{\boldsymbol{\beta}}_2)$.

New Variables (Type of Delay)	Variance Ratio
Security	0.49
Security, Weather	0.54
Security, Weather, NAS	0.66
Security, Weather, NAS, LateAircraft	0.79
Security, Weather, NAS, LateAircraft, Carrier	0.96

to that of $\hat{\boldsymbol{\beta}}_2$. The variables are sorted in the order of their z-statistics. Note that, because this is a single data analysis, we can only report the variance ratio estimate instead of the relative efficiency. This variance ratio is much smaller than 1, ranging from 0.49 to 0.67, suggesting that, by including the data in May 2003, the cumulative estimate of the coefficient of distance have a smaller standard error than the estimate from the June 2003 data alone. As the value of z-statistics increase, the variance ratio also increases, which is expected because a larger z value indicates a more important z or less relative importance of x, and, hence, bringing in information from existing data becomes less effective.

Next we consider a multiple linear regression with $\mathbf{x}$ being the distance and $\mathbf{z}$ being all five delay causes, and add the five delay causes in the same order as in Table 9. The variance ratios of the coefficient estimate of distance at all five steps are summarized in Table 10. As the number of new variables q increases from 1 to 5, the importance of the whole set of $\mathbf{z}$ relative to x increases, and consequently, the relative advantage of the adjusted cumulative estimate decreases, and the variance ratio increases. The impact is

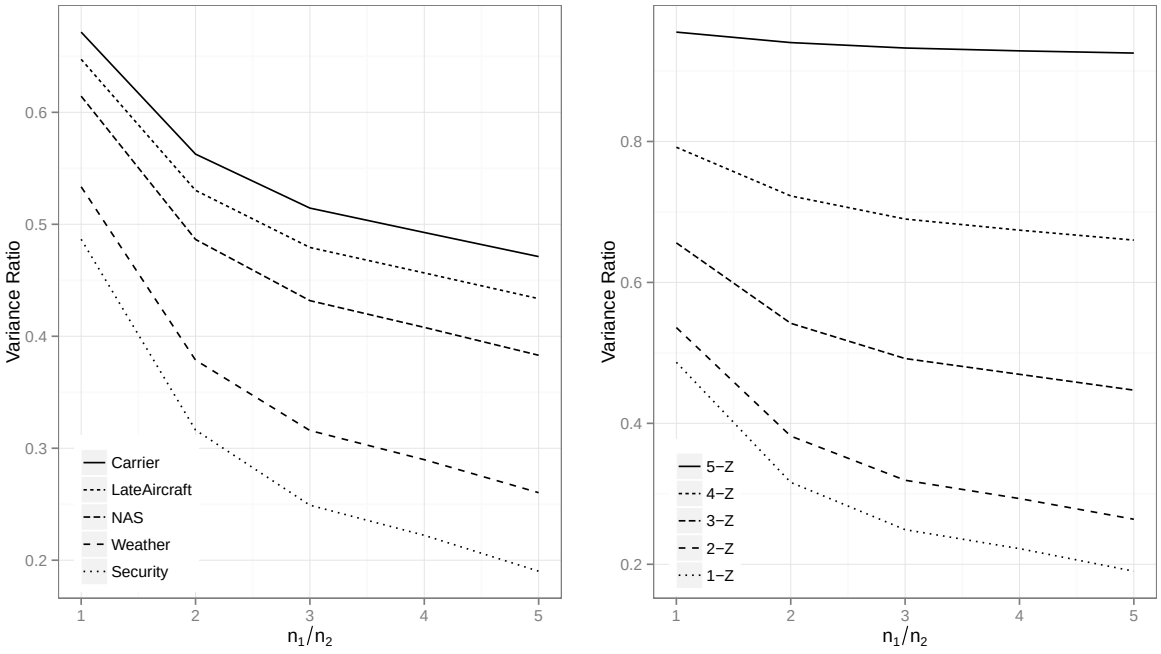

Figure 10: Under the linear model, block 2 data is fixed as the data of June 2003, while block 1 data varies from May 2003 to January-May 2003. That is the ratio of sample sizes n_1/n_2 changes from 1 to 5. In the left plot, each line comes from a three-variable case where the new variable z changes among the five types of delay. In the right plot, each line is a multiple linear regression with different number of new variable $\mathbf{z}$. The variance ratio $\mathsf{Var}(\hat{\boldsymbol{\beta}}_2)/\mathsf{Var}(\hat{\boldsymbol{\beta}}_2)$ decreases as n_1/n_2 increases.

also related to the significance of the added variable. From Table 9, we have seen that Security Delay is the least significant covariate in all the five causes of delay. That is why that in Table 9, the variance ratio for the weather delay is 0.53, and in Table 10, adding SecurityDelay has almost no impact on the ratio (0.54). This suggests that, in practice, when too many new variables or some extremely significant variables show up, it may be preferable to discard the previous information and start from the scratch.

The impact of the sample sizes of the previous data and current data can be investigated by including more data before June 2003. We kept the data of June 2003 as the

Table 11: In a logistic regression where y is the arrival delay (binary), x is the distance, and the newly added variable z is one of the five types of delay, the variance ratio The variance ratio is defined as $\mathsf{Var}(\tilde{\boldsymbol{\beta}}_2)/\mathsf{Var}(\hat{\boldsymbol{\beta}}_2)$.

	z-Statistics	Variance Ratio
Security	23.35	0.55
Weather	39.00	0.57
LateAircraft	113.34	0.67
Carrier	126.47	0.70
NAS	170.76	0.79

Table 12: In the logistic regreesion where y is the arrival delay (binary), x is the distance, types of delay are added into the model one by one as new variables. The variance ratio is defined as $\mathsf{Var}(\tilde{\boldsymbol{\beta}}_2)/\mathsf{Var}(\hat{\boldsymbol{\beta}}_2)$.

New Variables (Type of Delay)	Variance Ratio
Security	0.55
Security, Weather	0.58
Security, Weather, LateAircraft	0.69
Security, Weather, LateAircraft, Carrier	0.82
Security, Weather, LateAircraft, Carrier, NAS	1.00

new block of data, and expanded the previous data to include those from January–May 2003, which increases the ratio of sample sizes n_1/n_2 roughly from 1 to 5. The left panel of Figure 10 shows the variance ratio against n_1/n_2 for the three-variable case as each delay cause serves as z. As more months of data are included as existing data, the variance ratio drops below 0.5 for even the most important delay cause. The right panel of Figure 10 shows the variance ratio against n_1/n_2 in the multiple linear regression with different number of new variable $\mathbf{z}$. The more the new variables there are, the higher the lines. Even for the model with 5 new variables, the sample size still has an effect on the variance ratio.

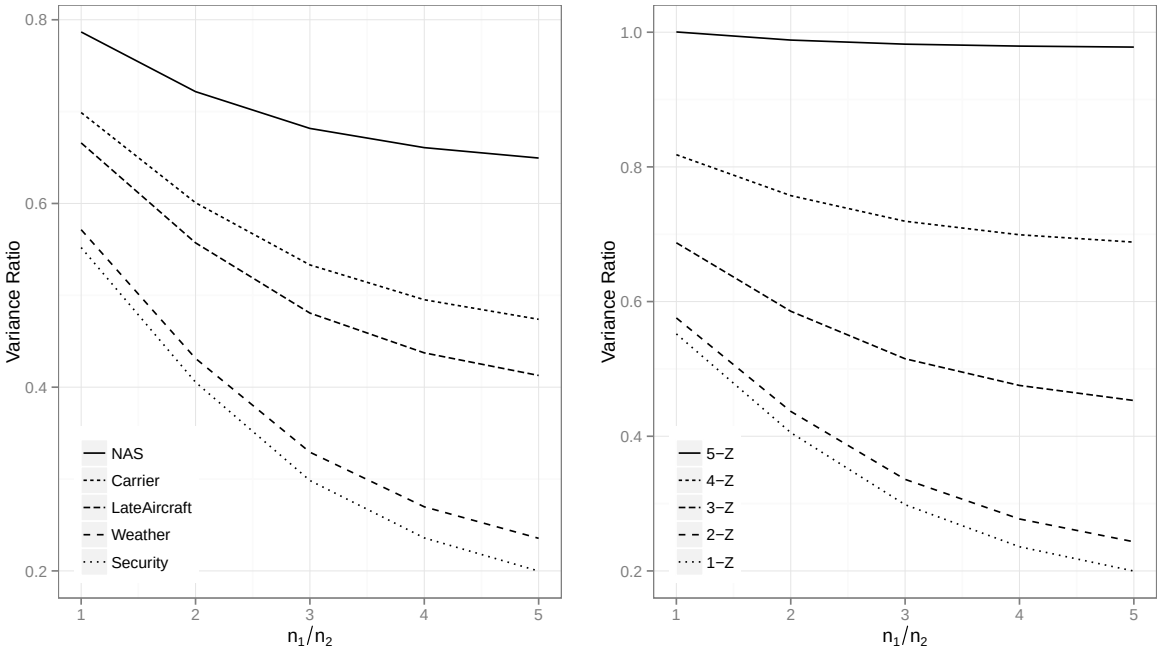

Figure 11: Under the logistic model, block 2 data is fixed as the data of June 2003, while block 1 data varies from May 2003 to January-May 2003. That is the ratio of sample sizes n_1/n_2 changes from 1 to 5. In the left plot, each line comes from a three-variable case where the new variable z changes among the five types of delay. In the right plot, each line is a regression with different number of new variable $\mathbf{z}$. The variance ratio is defined as $\mathsf{Var}(\tilde{\boldsymbol{\beta}}_2)/\mathsf{Var}(\hat{\boldsymbol{\beta}}_2)$.

Similar analyses were performed for the logistic regression model for the chance of over 15 minutes arrival delay; that is, the response variable is binary, 1 if late by more than 15 minutes and 0 otherwise. Table 11 and Table 12 are, respectively, the analogs of Table 9 and Table 10. Figure 11 is the analog of Figure 10. Note that in Table 11, the order of importance of the delay causes is different from that in Table 9, suggesting that the dichotomized arrival delay does have quite different behavior than the pre-dichotomized continuous version. The conclusions about the impact of the added variables, however, remain the same as those in the LM setting.

Chapter 5

Discussion

This dissertation summarizes the recent developments on statistical analysis with big data that exceed the memory and computing capacity of a single computer, presents the online updating algorithms and inferences applicable for linear models and estimating equations which is an ideal solution for big data streams. In certain conditions when the online updating method does not suffice (e.g., rank deficiencies for rare-event situation, added variables situation), modifications on the method are given.

From the computing perspective, Albeit under-appreciated by the general public or even mainstream academic community, computational statisticians have made respectable progress in extending standard statistical analysis to big data, with the most notable achievements in the open source R community. Packages **bigmemory** and **ff** make it possible in principle to implement any statistical analysis with their data structure. Nonetheless, for anything that has not been already implemented (e.g., survival analysis, generalized estimating equations, mixed effects model, etc.), one would need to implement an EMA version of the computation task, which may not be straightforward and may involve some steep learning curves. Hadoop allows easy extension of algorithms

that do not require multiple passes of the data, but such analyses are mostly descriptive. An example is visualization, an important tool in exploratory analysis. With big data, the bottleneck is the number of pixels in the screen. The bin-summarize-smooth framework for visualization of large data of Wickham (2014a) with package **bigvis** (Wickham, 2013) may be adapted to work with Hadoop.

From the online-updating methodological perspective we provided a method for outlier detection using predictive residuals. Our simulations suggested that the predictive residual tests are more powerful than a test that uses only the current dataset in the stream. In the EE setting, we may similarly consider outlier tests also based on standardized predictive residuals. For example in generalized linear models, one may consider the sum of squared predictive Pearson or Deviance residuals, computed using the coefficient estimate from the cumulative data (i.e., $\tilde{\boldsymbol{\beta}}_{k-1}$ or $\hat{\boldsymbol{\beta}}_{k-1}$). It remains an open question in both settings, however, regarding how to handle such outliers when they are detected. This is an area of future research.

In the estimating equation setting, we also proposed a new online-updated estimator of the regression coefficients that borrows information from previous datasets in the data stream. The simulations indicated that in finite samples, the proposed CUEE estimator is less biased than the AEE/CEE estimator of Lin and Xi (2011). However, both estimators were shown to be asymptotically consistent.

The methods in this paper were designed for small to moderate covariate dimensionality p, but large N. The use of penalization in the large p setting is an interesting

consideration, and has been explored in the divide-and-conquer context in Chen and Xie (2014) with popular sparsity inducing penalty functions. In our online-updating framework, inference for penalized parameters would be challenging, however, as the computation of their variance estimates is quite complicated and is also an area of future work.

The proposed methods are particularly useful for data that is obtained sequentially and without access to historical data. Notably, under the normal linear regression model, the proposed scheme does not lead to any information loss for inferences involving $\boldsymbol{\beta}$, as when the design matrix is of full rank, $\hat{\boldsymbol{\beta}}_{n_k,k}$ and $\text{MSE}_{n_k,k}$ are sufficient and complete statistics for $\boldsymbol{\beta}$ and σ^2. However, under the estimating equation setting, some information will be lost. Precisely how much information needs to be retained at each subset for specific types of inferences is an open question, and an area devoted for future research.

When new variables become available, we have shown that by keeping the saved information from previous data before the changing block but with some correction will improve the cumulative estimates of regression coefficients under a mild condition. The efficiency gain is large when the correlation between existing covariates and newly added covariates are not too large, the newly added variables are not dominating the predictions and cumulative sample size is relatively larger than the one from the changing block. These situations are very common in practice. Another problem which may arise in addition to the added variable problem is the rank deficiency. To implement the method in this paper, one has to make sure that the new variable is not singular at the

changing block or the design matrix at the changing block is of full rank. The method proposed here has covered Linear Regression and Generalized Linear Models. Estimating Equations and Generalized Estimating Equations are more complicated especially with the working correlations involved. We leave these as our future work.

Big data present challenges much further beyond the territory of classic statistics, requiring joint workforce with domain knowledge, computing skills, and statistical thinking (Yu, 2014). Statisticians have much to contribute to both the intellectual vitality and the practical utility of big data, but will have to expand their comfort zone to engage high-impact, real world problems which are often less structured or with ambiguity (Jordan and Lin, 2014). Examples are to provide structure for poorly defined problems, or to develop methods/models for new types of data such as image or network. As suggested by Yu (2014), to play a critical role in the arena of big data or own data science, statisticians need to work on real problems and relevant methodology and theory will follow naturally.

Appendix A

Online Updating Supplementation

A.1 Bayesian Insight into Online Updating

A Bayesian perspective provides some insight into how we may construct our online-updating estimators. Under a Bayesian framework, using the previous $k-1$ subsets of data to construct a prior distribution for the current data in subset k, we immediate identify the appropriate online updating formulae for estimating the regression coefficients and the error variance. Conveniently, these formulae require storage of only a few low-dimensional quantities computed only within the current subset; storage of these quantities is not required across all subsets.

We first assume a joint conjugate prior for $(\boldsymbol{\beta}, \sigma^2)$ as follows:

$$\pi(\boldsymbol{\beta}, \sigma^2 | \boldsymbol{\mu}_0, \mathbf{V}_0, \nu_0, \tau_0) = \pi(\boldsymbol{\beta} | \sigma^2, \boldsymbol{\mu}_0, \mathbf{V}_0) \pi(\sigma^2 | \nu_0, \tau_0), \tag{A.1}$$

where $\boldsymbol{\mu}_0$ is a prespecified p-dimensional vector, $\mathbf{V}_0$ is a $p \times p$ positive definite precision

matrix, $\nu_0 > 0$, $\tau_0 > 0$, and

$$\begin{aligned}\pi(\boldsymbol{\beta}|\sigma^2, \boldsymbol{\mu}_0, \mathbf{V}_0) &= \frac{|\mathbf{V}_0|^{1/2}}{(2\pi\sigma^2)^{p/2}} \exp\Big\{ -\frac{1}{2\sigma^2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)'\mathbf{V}_0(\boldsymbol{\beta}-\boldsymbol{\mu}_0)\Big\},\\ \pi(\sigma^2|\nu_0, \tau_0) &\propto (\sigma^2)^{-(\nu_0/2+1)} \exp\Big\{ -\frac{\tau_0}{2\sigma^2}\Big\}.\end{aligned}$$

When the data $D_1 = \{(\mathbf{y}_1, \mathbf{X}_1)\}$ is available, the likelihood is given by

$$L(\boldsymbol{\beta}, \sigma^2|D_1) \propto \frac{1}{(\sigma^2)^{n_1/2}} \exp\Big\{ -\frac{1}{2\sigma^2}(\mathbf{y}_1 - \mathbf{X}_1\boldsymbol{\beta})'(\mathbf{y}_1 - \mathbf{X}_1\boldsymbol{\beta})\Big\}.$$

After some algebra, we can show that the posterior distribution of $(\boldsymbol{\beta}, \sigma^2)$ is then given by

$$\pi(\boldsymbol{\beta}, \sigma^2|D_1, \boldsymbol{\mu}_0, \mathbf{V}_0, \nu_0, \tau_0) = \pi(\boldsymbol{\beta}|\sigma^2, \boldsymbol{\mu}_1, \mathbf{V}_1)\pi(\sigma^2|\nu_1, \tau_1),$$

where $\boldsymbol{\mu}_1 = (\mathbf{X}_1'\mathbf{X}_1+\mathbf{V}_0)^{-1}(\mathbf{X}_1'\mathbf{X}_1\hat{\boldsymbol{\beta}}_{n_1,1}+\mathbf{V}_0\boldsymbol{\mu}_0)$, $\mathbf{V}_1 = \mathbf{X}_1'\mathbf{X}_1+\mathbf{V}_0$, $\nu_1 = n_1+\nu_0$, and $\tau_1 = \tau_0+(\mathbf{y}_1-\mathbf{X}_1\hat{\boldsymbol{\beta}}_{n_1,1})'(\mathbf{y}_1-\mathbf{X}_1\hat{\boldsymbol{\beta}}_{n_1,1})+\boldsymbol{\mu}_0'\mathbf{V}_0\boldsymbol{\mu}_0+\hat{\boldsymbol{\beta}}_{n_1,1}'\mathbf{X}_1'\mathbf{X}_1\hat{\boldsymbol{\beta}}_{n_1,1}-\boldsymbol{\mu}_1'\mathbf{V}_1\boldsymbol{\mu}_1$; see, for example, Section 8.6 of DeGroot and Schevish (2012). Using mathematical induction, we can show that given the data $D_k = \{(\mathbf{y}_\ell, \mathbf{X}_\ell),\ \ell = 1, 2, \ldots, k\}$, the posterior distribution of $(\boldsymbol{\beta}, \sigma^2)$ is $\pi(\boldsymbol{\beta}, \sigma^2|\boldsymbol{\mu}_k, \mathbf{V}_k, \nu_k, \tau_k)$, which has the same form as in (A.1) with $(\boldsymbol{\mu}_0, \mathbf{V}_0, \nu_0, \tau_0)$

updated by $(\boldsymbol{\mu}_k, \mathbf{V}_k, \nu_k, \tau_k)$, where

$$\begin{aligned}
\boldsymbol{\mu}_k &= (\mathbf{X}_k'\mathbf{X}_k + \mathbf{V}_{k-1})^{-1}(\mathbf{X}_k'\mathbf{X}_k\hat{\boldsymbol{\beta}}_{n_k,k} + \mathbf{V}_{k-1}\boldsymbol{\mu}_{k-1}), \\
\mathbf{V}_k &= \mathbf{X}_k'\mathbf{X}_k + \mathbf{V}_{k-1}, \\
\nu_k &= n_k + \nu_{k-1}, \\
\tau_k &= \tau_{k-1} + (\mathbf{y}_k - \mathbf{X}_k\hat{\boldsymbol{\beta}}_{n_k,k})'(\mathbf{y}_k - \mathbf{X}_k\hat{\boldsymbol{\beta}}_{n_k,k}) \\
&\quad + \boldsymbol{\mu}_{k-1}'\mathbf{V}_{k-1}\boldsymbol{\mu}_{k-1} + \hat{\boldsymbol{\beta}}_{n_k,1}'\mathbf{X}_k'\mathbf{X}_k\hat{\boldsymbol{\beta}}_{n_k,k} - \boldsymbol{\mu}_k'\mathbf{V}_k\boldsymbol{\mu}_k,
\end{aligned} \tag{A.2}$$

for $k = 1, 2, \ldots$. The data stream structure fits the Bayesian paradigm perfectly and the Bayesian online updating sheds light on the online updating of LS estimators. Let $\hat{\boldsymbol{\beta}}_k$ and MSE_k denote the LS estimate of $\boldsymbol{\beta}$ and the corresponding MSE based on the cumulative data $D_k = \{(\mathbf{y}_\ell, \mathbf{X}_\ell),\ \ell = 1, 2, \ldots, k\}$. As a special case of Bayesian online update, we can derive the online updates of $\hat{\boldsymbol{\beta}}_k$ and MSE_k. Specifically, we take $\hat{\boldsymbol{\beta}}_1 = \hat{\boldsymbol{\beta}}_{n_1,1}$ and use the updating formula for $\boldsymbol{\mu}_k$ in (A.2). That is, taking $\boldsymbol{\mu}_0 = \mathbf{0}$ and $\mathbf{V}_0 = \mathbf{0}_p$ in (A.2), we obtain

$$\hat{\boldsymbol{\beta}}_k = (\mathbf{X}_k'\mathbf{X}_k + \mathbf{V}_{k-1})^{-1}(\mathbf{X}_k'\mathbf{X}_k\hat{\boldsymbol{\beta}}_{n_k k} + \mathbf{V}_{k-1}\hat{\boldsymbol{\beta}}_{k-1}), \tag{A.3}$$

where $\hat{\boldsymbol{\beta}}_0 = \mathbf{0}$, and $\mathbf{V}_k = \sum_{\ell=1}^{k} \mathbf{X}_\ell'\mathbf{X}_\ell$ for $k = 1, 2, \ldots$.

Similarly, taking $\nu_0 = n_0 = 0$, $\tau_0 = \mathsf{SSE}_0 = 0$, and using the updating formula for τ_k in (A.2), we have

$$\mathsf{SSE}_k = \mathsf{SSE}_{k-1} + \mathsf{SSE}_{n_k,k} + \hat{\boldsymbol{\beta}}_{k-1}'\mathbf{V}_{k-1}\hat{\boldsymbol{\beta}}_{k-1} + \hat{\boldsymbol{\beta}}_{n_k,k}'\mathbf{X}_k'\mathbf{X}_k\hat{\boldsymbol{\beta}}_{n_k,k} - \hat{\boldsymbol{\beta}}_k'\mathbf{V}_k\hat{\boldsymbol{\beta}}_k \tag{A.4}$$

where $\mathsf{SSE}_{n_k,k}$ is the residual sum of squares from the k^{th} dataset, with corresponding residual mean square $\mathsf{MSE}_{n_k,k} = \mathsf{SSE}_{n_k,k}/(n_k - p)$. The MSE based on the data D_k is then $\mathsf{MSE}_k = \mathsf{SSE}_k/(N_k - p)$ where $N_k = \sum_{\ell=1}^{k} n_\ell$ $(= n_k + N_{k-1})$ for $k = 1, 2, \ldots$. Note that for $k = K$, equations (A.3) and (A.4) are identical to those in (3.2) and (3.3), respectively.

Remark A.1 *Following Remark 3.8, if $\boldsymbol{X}_1$ is not of full rank (e.g., due to a rare event covariate), we may consider a regularized least squares estimator by setting $\boldsymbol{V}_0 \neq \boldsymbol{0}_p$. For example, setting $\boldsymbol{V}_0 = \lambda \boldsymbol{I}_p, \lambda > 0$, with $\boldsymbol{\mu}_0 = \boldsymbol{0}$ would correspond to a ridge estimator and could be used at the beginning of the online estimation process until enough data has accumulated; once enough data has accumulated, the biasing term $\boldsymbol{V}_0 = \lambda \boldsymbol{I}_p$ may be removed such that the remaining sequence of updated estimators $\hat{\boldsymbol{\beta}}_k$ and MSE_k are unbiased for $\boldsymbol{\beta}$ and σ^2, respectively. More specifically, set $\boldsymbol{V}_k = \sum_{\ell=0}^{k} \boldsymbol{X}_\ell' \boldsymbol{X}_\ell$ (note that the summation starts at $\ell = 0$ rather than $\ell = 1$) where $\boldsymbol{X}_0' \boldsymbol{X}_0 \equiv \boldsymbol{V}_0$, keep $\hat{\boldsymbol{\beta}}_0 = \boldsymbol{0}$, and suppose at accumulation point κ we have accumulated enough data such that $\boldsymbol{X}_\kappa$ is of full rank. For $k < \kappa$ and $\boldsymbol{V}_0 = \lambda \boldsymbol{I}_p, \lambda > 0$, we obtain a (biased) ridge estimator and corresponding sum of squared errors by using* (3.4) *and* (3.5) *or* (3.23) *and* (3.24)*. At $k = \kappa$, we can remove the bias with, e.g.,*

$$\hat{\boldsymbol{\beta}}_\kappa = (\boldsymbol{X}_\kappa' \boldsymbol{X}_\kappa + \boldsymbol{V}_{\kappa-1} - \boldsymbol{V}_0)^{-1} (\boldsymbol{X}_\kappa' \boldsymbol{y}_\kappa + \boldsymbol{W}_{\kappa-1}) \tag{A.5}$$

$$\mathsf{SSE}_\kappa = \mathsf{SSE}_{\kappa-1} + \boldsymbol{y}_\kappa' \boldsymbol{y}_\kappa + \hat{\boldsymbol{\beta}}_{\kappa-1}' \boldsymbol{V}_{\kappa-1} \hat{\boldsymbol{\beta}}_{\kappa-1} - \hat{\boldsymbol{\beta}}_\kappa' (\boldsymbol{V}_\kappa - \boldsymbol{V}_0) \hat{\boldsymbol{\beta}}_\kappa, \tag{A.6}$$

and then use the original updating procedure for $k > \kappa$ to obtain unbiased estimators of $\boldsymbol{\beta}$ and σ^2.

A.2 Online Updating Statistics in Linear Models

Below we provide online-updated t-tests for the regression parameter estimates, the online-updated ANOVA table, and online-updated general linear hypothesis F-tests. Please refer to Section 3.2.2 of the main text for the relevant notation.

Online Updating for Parameter Estimate t-tests in Linear Models. If our interest is only in performing t-tests for the regression coefficients, we only need to save the current values $(\mathbf{V}_k, \hat{\boldsymbol{\beta}}_k, N_k, \mathsf{MSE}_k)$ to proceed. Recall that $\text{var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$ and $\widehat{\text{var}}(\hat{\boldsymbol{\beta}}) = \mathsf{MSE}(\mathbf{X}'\mathbf{X})^{-1}$. At the k^{th} update, $\widehat{\text{var}}(\hat{\boldsymbol{\beta}}_k) = \mathsf{MSE}_k\mathbf{V}_k^{-1}$. Thus, to test $H_0 : \beta_j = 0$ at the k^{th} update $(j = 1, \ldots, p)$, we may use $t^*_{k,j} = \hat{\beta}_{k,j}/se(\hat{\beta}_{k,j})$, where the standard error $se(\hat{\beta}_{k,j})$ is the square root of the j^{th} diagonal element of $\widehat{\text{var}}(\hat{\boldsymbol{\beta}}_k)$. The corresponding p-value is $P(|t_{N_k-p}| \geq |t^*_{k,j}|)$.

Online Updating for ANOVA Table in Linear Models. Observe that SSE is given by (3.3),

$$\mathsf{SST} = \mathbf{y}'\mathbf{y} - N\bar{y}^2 = \sum_{k=1}^{K} \mathbf{y}_k'\mathbf{y}_k - N^{-1}(\sum_{k=1}^{K} \mathbf{y}_k'\mathbf{1}_{n_k})^2,$$

where $\mathbf{1}_{n_k}$ is an n_k length vector of ones, and SSR = SST-SSE. If we wish to construct an online-updated ANOVA table, we must save two additional easily computable, low dimensional quantities: $S_{yy,k} = \sum_{\ell=1}^{k} \mathbf{y}_\ell'\mathbf{y}_\ell$ and $S_{y,k} = \sum_{\ell=1}^{k} \mathbf{y}_\ell'\mathbf{1}_{n_\ell} = \sum_{\ell=1}^{k}\sum_{i=1}^{n_\ell} y_{\ell i}$.

The online-updated ANOVA table at the k^{th} update for the cumulative data D_k is constructed as in Table 13. Note that SSE_k is computed as in (A.4). The table may be completed upon determination of an updating formula SST_k. Towards this end, write $S_{yy,k} = \mathbf{y}_k'\mathbf{y}_k + S_{yy,k-1}$ and $S_{y,k} = \mathbf{y}_k' 1_{n_k} + S_{y,k-1}$, for $k = 1, \ldots, K$ and $S_{yy,0} = S_{y,0} = 0$, so that $\mathsf{SST}_k = S_{yy,k} - N_k^{-1} S_{y,k}^2$

Table 13: Online-updated ANOVA Table

ANOVA_{N_k} Table Source	df	SS	MS	F	P-value
Regression	$p-1$	SSR_k	$\mathsf{MSR}_k = \frac{\mathsf{SSR}_k}{p-1}$	$F^* = \frac{\mathsf{MSR}_k}{\mathsf{MSE}_k}$	$P(F_{p-1,N_k-p} \geq F^*)$
Error	$N_k - p$	SSE_k	$\mathsf{MSE}_k = \frac{\mathsf{SSE}_k}{N_k-p}$		
C Total	$N_k - 1$	SST_k			

Online updated testing of General Linear Hypotheses ($H_0 : \mathbf{C}\boldsymbol{\beta} = \mathbf{0}$) are also possible: if $\mathbf{C}$ $(q \times p)$ is a full rank $(q \leq p)$ contrast matrix, under H_0,

$$F_k = (\frac{\hat{\boldsymbol{\beta}}_k' \mathbf{C}' (\mathbf{C}\mathbf{V}_k^{-1}\mathbf{C}')^{-1} \mathbf{C}\hat{\boldsymbol{\beta}}_k}{q}) / (\frac{\mathsf{SSE}_k}{N_k - p}) \sim F_{q,N_k-p}.$$

Similarly, we may also obtain online updated coefficients of multiple determination, $R_k^2 = \mathsf{SSR}_k / \mathsf{SST}_k$.

To summarize, we need only save $(\mathbf{V}_{k-1}, \hat{\boldsymbol{\beta}}_{k-1}, N_{k-1}, \mathsf{MSE}_{k-1}, S_{yy,k-1}, S_{y,k-1})$ from the previous accumulation point $k-1$ to perform online-updated t-tests for $H_0 : \beta_j = 0$, $j = 1, \ldots, p$ and online-updated F-tests for the current accumulation point k; we do not need to retain $(\mathbf{V}_\ell, \hat{\boldsymbol{\beta}}_\ell, N_\ell, \mathsf{MSE}_\ell, S_{yy,\ell}, S_{y,\ell})$ for $\ell = 1, \ldots, k-2$.

A.3 Proof of Proposition 3.1

We first show that $\mathsf{MSE}_{k-1} \xrightarrow{p} \sigma^2$. Since $\mathsf{SSE}_{k-1} = \boldsymbol{\varepsilon}'_{k-1}(\mathbf{I}_{N_{k-1}} - \boldsymbol{\mathcal{X}}_{k-1}(\boldsymbol{\mathcal{X}}'_{k-1}\boldsymbol{\mathcal{X}}_{k-1})^{-1}\boldsymbol{\mathcal{X}}'_{k-1})\boldsymbol{\varepsilon}_{k-1}$, we have

$$\begin{aligned}
\operatorname*{plim}_{N_{k-1}\to\infty} \mathsf{MSE}_{k-1} &= \operatorname*{plim}_{N_{k-1}\to\infty} \frac{\mathsf{SSE}_{k-1}}{N_{k-1}-p} \\
&= \operatorname*{plim}_{N_{k-1}\to\infty} \frac{\boldsymbol{\varepsilon}'_{k-1}\boldsymbol{\varepsilon}_{k-1}}{N_{k-1}} - \operatorname*{plim}_{N_{k-1}\to\infty} \frac{\boldsymbol{\varepsilon}'_{k-1}\boldsymbol{\mathcal{X}}_{k-1}(\boldsymbol{\mathcal{X}}'_{k-1}\boldsymbol{\mathcal{X}}_{k-1})^{-1}\boldsymbol{\mathcal{X}}'_{k-1}\boldsymbol{\varepsilon}_{k-1}}{N_{k-1}} \\
&= \sigma^2 - \operatorname*{plim}_{N_{k-1}\to\infty} \frac{\boldsymbol{\varepsilon}'_{k-1}\boldsymbol{\mathcal{X}}_{k-1}}{N_{k-1}} \operatorname*{plim}_{N_{k-1}\to\infty} \left(\frac{\boldsymbol{\mathcal{X}}'_{k-1}\boldsymbol{\mathcal{X}}_{k-1}}{N_{k-1}}\right)^{-1} \operatorname*{plim}_{N_{k-1}\to\infty} \frac{\boldsymbol{\mathcal{X}}'_{k-1}\boldsymbol{\varepsilon}_{k-1}}{N_{k-1}}
\end{aligned}$$

Let $\mathcal{X}_j$ denote the column vector of $\boldsymbol{\mathcal{X}}_{k-1}$, for $j = 1, \ldots, p$. Since $E(\epsilon_i) = 0$, $\forall i$ and all the elements of $\boldsymbol{\mathcal{X}}_{k-1}$ are bounded by C, by Chebyshev's Inequality we have for any ℓ and column vector $\mathcal{X}_j$,

$$P(|\frac{\boldsymbol{\varepsilon}'_{k-1}\mathcal{X}_j}{N_{k-1}}| \geq \ell) \leq \frac{Var(\boldsymbol{\varepsilon}'_{k-1}\mathcal{X}_j)}{\ell^2 N_{k-1}^2} \leq \frac{C^2\sigma^2}{\ell^2 N_{k-1}},$$

and thus $\operatorname*{plim}_{N_{k-1}\to\infty} \frac{\boldsymbol{\varepsilon}'_{k-1}\boldsymbol{\mathcal{X}}_{k-1}}{N_{k-1}} = 0$ and

$$\operatorname*{plim}_{N_{k-1}\to\infty} \mathsf{MSE}_{k-1} = \sigma^2 - 0 \cdot \mathbf{Q}^{-1} \cdot 0 = \sigma^2.$$

Next we show $\frac{\sum_{i=1}^{m}\frac{1}{n_{k_i}}(\mathbf{1}'_{k_i}\check{\mathbf{e}}^*_{k_i})^2}{\sigma^2} \xrightarrow{d} \chi^2_m$. First, recall that

$$\begin{aligned}
\check{\mathbf{e}}_k &= \mathbf{y}_k - \check{\mathbf{y}}_k \\
&= \mathbf{X}_k\boldsymbol{\beta} + \boldsymbol{\epsilon}_k - \mathbf{X}_k\hat{\boldsymbol{\beta}}_{k-1} \\
&= \mathbf{X}_k\boldsymbol{\beta} + \boldsymbol{\epsilon}_k - \mathbf{X}_k(\boldsymbol{\mathcal{X}}'_{k-1}\boldsymbol{\mathcal{X}}_{k-1})^{-1}\boldsymbol{\mathcal{X}}'_{k-1}\dagger_{k-1} \\
&= \boldsymbol{\epsilon}_k - \mathbf{X}_k(\boldsymbol{\mathcal{X}}'_{k-1}\boldsymbol{\mathcal{X}}_{k-1})^{-1}\boldsymbol{\mathcal{X}}'_{k-1}\boldsymbol{\varepsilon}_{k-1}.
\end{aligned}$$

Consequently, $\text{var}(\check{\mathbf{e}}_k) = (\mathbf{I}_{n_k} + \mathbf{X}_k(\boldsymbol{\mathcal{X}}'_{k-1}\boldsymbol{\mathcal{X}}_{k-1})^{-1}\mathbf{X}'_k)\sigma^2 \triangleq \boldsymbol{\Gamma}'\boldsymbol{\Gamma}\sigma^2$, where $\boldsymbol{\Gamma}$ is an $n_k \times n_k$ invertible matrix. Let $\check{\mathbf{e}}^*_k = (\boldsymbol{\Gamma}')^{-1}\check{\mathbf{e}}_k$ with $\text{var}(\check{\mathbf{e}}^*_k) = \sigma^2\mathbf{I}_{n_k}$. Therefore, each component of $\check{\mathbf{e}}^*_k$ is independent and identically distributed.

By the Central Limit Theorem and condition (iv), we have for all $i = 1, \ldots, m$,

$$\frac{\frac{1}{n_{k_i}}(\mathbf{1}'_{k_i}\check{\mathbf{e}}^*_{k_i})^2}{\sigma^2} \xrightarrow{d} \chi^2_1, \qquad \text{as } n_k \to \infty.$$

Since each subgroup is also independent,

$$\frac{\sum_{i=1}^{m}\frac{1}{n_{k_i}}(\mathbf{1}'_{k_i}\check{\mathbf{e}}^*_{k_i})^2}{\sigma^2} \xrightarrow{d} \chi^2_m, \qquad \text{as } n_k \to \infty.$$

By Slutsky's theorem,

$$\frac{\sum_{i=1}^{m}\frac{1}{n_{k_i}}(\mathbf{1}'_{k_i}\check{\mathbf{e}}^*_{k_i})^2}{\mathsf{MSE}_{k-1}} \xrightarrow{d} \chi^2_m, \qquad \text{as } n_k, N_{k-1} \to \infty.$$

■

A.4 Computation of Γ for Asymptotic F test

Recall that $\text{var}(\check{\mathbf{e}}_k) = (\mathbf{I}_{n_k} + \mathbf{X}_k(\boldsymbol{\mathcal{X}}'_{k-1}\boldsymbol{\mathcal{X}}_{k-1})^{-1}\mathbf{X}'_k)\sigma^2 \triangleq \boldsymbol{\Gamma}\boldsymbol{\Gamma}'\sigma^2$, where $\boldsymbol{\Gamma}$ is an $n_k \times n_k$ invertible matrix. For large n_k, it may be challenging to compute the Cholesky decomposition $\text{var}(\check{\mathbf{e}}_k)$. One possible solution that avoids the large n_k issue is given as follows.

First, we can easily obtain the Cholesky decomposition of $(\boldsymbol{\mathcal{X}}'_{k-1}\boldsymbol{\mathcal{X}}_{k-1})^{-1} = \mathbf{V}_{k-1}^{-1} \triangleq \mathbf{P}'\mathbf{P}$ since it is a $p \times p$ matrix. Thus, we have

$$\text{var}(\check{\mathbf{e}}_k) = (\mathbf{I}_{n_k} + \mathbf{X}_k\mathbf{P}'\mathbf{P}\mathbf{X}'_k)^{-1}\sigma^2 = (\mathbf{I}_{n_k} + \tilde{\mathbf{X}}_k\tilde{\mathbf{X}}'_k)^{-1}\sigma^2,$$

where $\tilde{\mathbf{X}}_k = \mathbf{X}_k\mathbf{P}'$ is an $n_k \times p$ matrix.

Next, we compute the singular value decomposition on $\tilde{\mathbf{X}}_k$, i.e., $\tilde{\mathbf{X}}_k = \mathbf{U}\mathbf{D}\mathbf{V}'$ where $\mathbf{U}$ is an $n_k \times n_k$ unitary matrix, $\mathbf{D}$ is an $n_k \times n_k$ diagonal matrix, and $\mathbf{V}$ is a $n_k \times p$ unitary matrix. Therefore,

$$\text{var}(\check{\mathbf{e}}_k) = (\mathbf{I}_{n_k} + \mathbf{U}\mathbf{D}\mathbf{D}'\mathbf{U}')^{-1}\sigma^2 = \mathbf{U}(\mathbf{I}_{n_k} + \mathbf{D}\mathbf{D}')^{-1}\mathbf{U}'\sigma^2$$

Since $(\mathbf{I}_{n_k} + \mathbf{D}\mathbf{D}')^{-1}$ is a diagonal matrix, we can find the matrix $\mathbf{Q}$ such that $(\mathbf{I}_{n_k} + \mathbf{D}\mathbf{D}')^{-1} \triangleq \mathbf{Q}'\mathbf{Q}$ by straightforward calculation. One possible choice of $\boldsymbol{\Gamma}$ is $\mathbf{U}\mathbf{Q}'$.

A.5 Proof of Theorem 3.3.1

The following technical conditions were provided in Lin and Xi (2011).

(C1) The score function ψ is measurable for any fixed $\boldsymbol{\beta}$ and is twice continuously differentiable with respect to $\boldsymbol{\beta}$.

(C2) The matrix $-\frac{\partial\psi(\mathbf{z}_i,\boldsymbol{\beta})}{\partial\boldsymbol{\beta}}$ is semi-positive definite, and $-\sum_{i=1}^{n}\frac{\partial\psi(\mathbf{z}_i,\boldsymbol{\beta})}{\partial\boldsymbol{\beta}}$ is positive definite (p.d.) in a neighborhood of $\boldsymbol{\beta}_0$ when n is large enough.

(C3) The EE estimator $\hat{\boldsymbol{\beta}}_{n,k}$ is strongly consistent, i.e. $\hat{\boldsymbol{\beta}}_{n,k} \to \boldsymbol{\beta}_0$ almost surely as $n \to \infty$.

(C4) There exists two p.d. matrices, $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$, such that $\boldsymbol{\Lambda}_1 \leq n^{-1}\mathbf{A}_{n,k} \leq \boldsymbol{\Lambda}_2$ for all $k = 1, \ldots, K$, i.e. for any $\mathbf{v} \in \mathbb{R}^p$, $\mathbf{v}'\boldsymbol{\Lambda}_1\mathbf{v} \leq n^{-1}\mathbf{v}'\mathbf{A}_{n,k}\mathbf{v} \leq \mathbf{v}'\boldsymbol{\Lambda}_2\mathbf{v}$, where $\mathbf{A}_{n,k}$ is given in (3.11).

(C5) In a neighborhood of $\boldsymbol{\beta}_0$, the norm of the second-order derivatives $\frac{\partial^2\psi_j(\mathbf{z}_i,\boldsymbol{\beta})}{\partial\boldsymbol{\beta}^2}$ is bounded uniformly, i.e. $\|\frac{\partial^2\psi_j(\mathbf{z}_i,\boldsymbol{\beta})}{\partial\boldsymbol{\beta}^2}\| \leq C_2$ for all i, j, where C_2 is a constant.

(C6) There exists a real number $\alpha \in (1/4, 1/2)$ such that for any $\eta > 0$, the EE estimator $\hat{\boldsymbol{\beta}}_{n,k}$ satisfies $P(n^{\alpha}\|\hat{\boldsymbol{\beta}}_{n,k} - \boldsymbol{\beta}_0\| > \eta) \leq C_\eta n^{2\alpha-1}$, where $C_\eta > 0$ is a constant only depending on η.

Rather than using condition (C4), we will use a slightly modified version which focuses on the behavior of $\mathbf{A}_{n,k}(\boldsymbol{\beta})$ for all $\boldsymbol{\beta}$ in the neighborhood of $\boldsymbol{\beta}_0$ (as in (C5)), rather than just at the subset estimate $\hat{\boldsymbol{\beta}}_{n,k}$.

(C4') In a neighborhood of $\boldsymbol{\beta}_0$, there exists two p.d. matrices $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$ such that $\boldsymbol{\Lambda}_1 \leq n^{-1}\mathbf{A}_{n,k}(\boldsymbol{\beta}) \leq \boldsymbol{\Lambda}_2$ for all $\boldsymbol{\beta}$ in the neighborhood of $\boldsymbol{\beta}_0$ and for all $k = 1, ..., K$.

We also use the same definition and two facts provided by Lin and Xi (2011), given below for completeness.

Definition E.1 *Let $\boldsymbol{A}$ be a $d \times d$ positive definite matrix. The norm of $\boldsymbol{A}$ is defined as* $\|\boldsymbol{A}\| = sup_{\boldsymbol{v}\in\mathbb{R}^d, \boldsymbol{v}\neq\mathbf{0}} \frac{\|\boldsymbol{A}\boldsymbol{v}\|}{\boldsymbol{v}}$.

Using the definition of the above matrix norm, one may verify the following two facts.

Fact E.1. *Suppose that $\boldsymbol{A}$ is a $d \times d$ positive definite matrix. Let λ be the smallest eigenvalue of $\boldsymbol{A}$, then we have $\boldsymbol{v}'\boldsymbol{A}\boldsymbol{v} \geq \lambda\boldsymbol{v}'\boldsymbol{v} = \lambda \|\boldsymbol{v}\|^2$ for any vector $\boldsymbol{v} \in \mathbb{R}^d$. On the contrary, if there exists a constant $C > 0$ such that $\boldsymbol{v}'\boldsymbol{A}\boldsymbol{v} \geq C \|\boldsymbol{v}\|^2$ for any vector $\boldsymbol{v} \in \mathbb{R}^d$, then $C \leq \lambda$.*

Fact E.2. *Let $\boldsymbol{A}$ be a $d \times d$ positive definite matrix and λ is the smallest eigenvalue of $\boldsymbol{A}$. If $\lambda \geq c > 0$ for some constant c, one has $\left\|\boldsymbol{A}^{-1}\right\| \leq c^{-1}$.*

In order to prove Theorem 3.3.1, we need the following Lemma.

Lemma E.2 *Under (C4') and (C6), $\check{\boldsymbol{\beta}}_{n,k}$ satisfies the following condition: for any $\eta > 0$, $n^{-2\alpha+1}P(n^\alpha\|\check{\boldsymbol{\beta}}_{n,k} - \boldsymbol{\beta}_0\| > \eta) = O(1)$.*

Proof of Lemma E.2 (By induction)

First notice that (C6) is equivalent to writing, for any $\eta > 0$, $n^{-2\alpha+1}P(n^\alpha\|\hat{\boldsymbol{\beta}}_{n,k} - \boldsymbol{\beta}_0\| > \eta) = O(1)$.

Take $k = 1$, $\check{\boldsymbol{\beta}}_{n,1} = \hat{\boldsymbol{\beta}}_{n,1}$ and thus $n^{-2\alpha+1}P(n^\alpha\|\check{\boldsymbol{\beta}}_{n,1} - \boldsymbol{\beta}_0\| > \eta) = O(1)$.

Assume the condition holds for accumulation point $k-1$: $n^{-2\alpha+1}P(n^{\alpha}\|\check{\boldsymbol{\beta}}_{n,k-1}-\boldsymbol{\beta}_0\|>\eta)=O(1)$. Write

$$\check{\boldsymbol{\beta}}_{n,k-1}=(\tilde{\mathbf{A}}_{k-2}+\mathbf{A}_{n,k-1})^{-1}(\sum_{\ell=1}^{k-2}\tilde{\mathbf{A}}_{n,\ell}\check{\boldsymbol{\beta}}_{n,\ell}+\mathbf{A}_{n,k-1}\hat{\boldsymbol{\beta}}_{n,k-1})$$

so that, rearranging terms, we have

$$\sum_{\ell=1}^{k-2}\tilde{\mathbf{A}}_{n,\ell}\check{\boldsymbol{\beta}}_{n,\ell}=(\tilde{\mathbf{A}}_{k-2}+\mathbf{A}_{n,k-1})\check{\boldsymbol{\beta}}_{n,k-1}-\mathbf{A}_{n,k-1}\hat{\boldsymbol{\beta}}_{n,k-1}.$$

Using the previous relation, we may write $\check{\boldsymbol{\beta}}_{n,k}$ as

$$\begin{aligned}\check{\boldsymbol{\beta}}_{n,k}&=(\tilde{\mathbf{A}}_{k-1}+\mathbf{A}_{n,k})^{-1}(\tilde{\mathbf{A}}_{k-2}\check{\boldsymbol{\beta}}_{n,k-1}+\tilde{\mathbf{A}}_{n,k-1}\check{\boldsymbol{\beta}}_{n,k-1}+\\&\qquad\mathbf{A}_{n,k}\hat{\boldsymbol{\beta}}_{n,k}+\mathbf{A}_{n,k-1}(\check{\boldsymbol{\beta}}_{n,k-1}-\hat{\boldsymbol{\beta}}_{n,k-1}))\\&=(\tilde{\mathbf{A}}_{k-1}+\mathbf{A}_{n,k})^{-1}(\tilde{\mathbf{A}}_{k-1}\check{\boldsymbol{\beta}}_{n,k-1}+\mathbf{A}_{n,k}\hat{\boldsymbol{\beta}}_{n,k}+\mathbf{A}_{n,k-1}(\check{\boldsymbol{\beta}}_{n,k-1}-\hat{\boldsymbol{\beta}}_{n,k-1})).\end{aligned}$$

Therefore,

$$\begin{aligned}\check{\boldsymbol{\beta}}_{n,k}-\boldsymbol{\beta}_0&=(\tilde{\mathbf{A}}_{k-1}+\mathbf{A}_{n,k})^{-1}(\tilde{\mathbf{A}}_{k-1}(\check{\boldsymbol{\beta}}_{n,k-1}-\boldsymbol{\beta}_0)+\mathbf{A}_{n,k}(\hat{\boldsymbol{\beta}}_{n,k}-\boldsymbol{\beta}_0)+\\&\qquad\mathbf{A}_{n,k-1}(\check{\boldsymbol{\beta}}_{n,k-1}-\boldsymbol{\beta}_0+\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}_{n,k-1}))\end{aligned}$$

and

$$\begin{aligned}\|\check{\boldsymbol{\beta}}_{n,k}-\boldsymbol{\beta}_0\|&\le\|(\tilde{\mathbf{A}}_{k-1}+\mathbf{A}_{n,k})^{-1}\tilde{\mathbf{A}}_{k-1}\|\|\check{\boldsymbol{\beta}}_{n,k-1}-\boldsymbol{\beta}_0\|+\\&\qquad\|(\tilde{\mathbf{A}}_{k-1}+\mathbf{A}_{n,k})^{-1}\mathbf{A}_{n,k}\|\|\hat{\boldsymbol{\beta}}_{n,k}-\boldsymbol{\beta}_0\|+\\&\qquad\|(\tilde{\mathbf{A}}_{k-1}+\mathbf{A}_{n,k})^{-1}\mathbf{A}_{n,k-1}\|\|\check{\boldsymbol{\beta}}_{n,k-1}-\boldsymbol{\beta}_0\|+\\&\qquad\|(\tilde{\mathbf{A}}_{k-1}+\mathbf{A}_{n,k})^{-1}\mathbf{A}_{n,k-1}\|\|\hat{\boldsymbol{\beta}}_{n,k-1}-\boldsymbol{\beta}_0\|\end{aligned}$$

Note that $\|(\tilde{\mathbf{A}}_{k-1}+\mathbf{A}_{n,k})^{-1}\tilde{\mathbf{A}}_{k-1}\| \leq 1$ and $\|(\tilde{\mathbf{A}}_{k-1}+\mathbf{A}_{n,k})^{-1}\mathbf{A}_{n,k}\| \leq 1$. Under (C4'), $\|(\tilde{\mathbf{A}}_{k-1}+\mathbf{A}_{n,k})^{-1}\mathbf{A}_{n,k-1}\| \leq \|(\mathbf{A}_{n,k})^{-1}\mathbf{A}_{n,k-1}\| \leq \frac{\lambda_2}{\lambda_1} \leq C$, where C is a constant, $\lambda_1 > 0$ is the smallest eigenvalue of $\mathbf{\Lambda}_1$, and λ_2 is the largest eigenvalue of $\mathbf{\Lambda}_2$. Note that if $n_k \neq n$ for all k, then $\|(\tilde{\mathbf{A}}_{k-1}+\mathbf{A}_{n,k})^{-1}\mathbf{A}_{n,k-1}\| \leq \|(\mathbf{A}_{n,k})^{-1}\mathbf{A}_{n,k-1}\| \leq \frac{n_{k-1}}{n_k}\frac{\lambda_2}{\lambda_1} \leq C$, where n_{k-1}/n_k is bounded and C is a constant. Thus,

$$\|\check{\boldsymbol{\beta}}_{n,k}-\boldsymbol{\beta}_0\| \leq \|\check{\boldsymbol{\beta}}_{n,k-1}-\boldsymbol{\beta}_0\| + \|\hat{\boldsymbol{\beta}}_{n,k}-\boldsymbol{\beta}_0\| + \|C(\check{\boldsymbol{\beta}}_{n,k-1}-\boldsymbol{\beta}_0)\| + \|C(\hat{\boldsymbol{\beta}}_{n,k-1}-\boldsymbol{\beta}_0)\|$$

Under (C6) and the induction hypothesis, then for any $\eta > 0$,

$$\begin{aligned} n^{-2\alpha+1}P(\|\check{\boldsymbol{\beta}}_{n,k}-\boldsymbol{\beta}_0\| > \frac{\eta}{n^\alpha}) \leq\ & n^{-2\alpha+1}P(\|\check{\boldsymbol{\beta}}_{n,k-1}-\boldsymbol{\beta}_0\| > \frac{\eta}{4n^\alpha}) + \\ & n^{-2\alpha+1}P(\|\hat{\boldsymbol{\beta}}_{n,k}-\boldsymbol{\beta}_0\| > \frac{\eta}{4n^\alpha}) + \\ & n^{-2\alpha+1}\quad P(\|\check{\boldsymbol{\beta}}_{n,k-1}-\boldsymbol{\beta}_0\| > \frac{\eta}{4Cn^\alpha}) + \\ & n^{-2\alpha+1}P(\|\hat{\boldsymbol{\beta}}_{n,k-1}-\boldsymbol{\beta}_0\| > \frac{\eta}{4Cn^\alpha}) \end{aligned}$$

Since all the four terms on the right hand side are $O(1)$ by assumption, $n^{-2\alpha+1}P(\|\check{\boldsymbol{\beta}}_{n,k}-\boldsymbol{\beta}_0\| > \frac{\eta}{n^\alpha}) = O(1)$. ■

We are now ready to prove Theorem 3.3.1:

Proof of Theorem 3.3.1

First, suppose that all the random variables are defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

Let

$$\Omega_{n,k,\eta} = \{\omega | n^{\alpha} \| \check{\boldsymbol{\beta}}_{n,k} - \boldsymbol{\beta}_0 \| \leq \eta\},$$

$$\Omega_{N,\eta} = \{\omega | N^{\alpha} \| \hat{\boldsymbol{\beta}}_{N} - \boldsymbol{\beta}_0 \| \leq \eta\},$$

$$\Gamma_{N,k,\eta} = \cap_{k=1}^{K} \Omega_{n,k,\eta} \cap \Omega_{N,\eta}.$$

From Lemma E.2, for any $\omega > 0$, we have

$$\begin{aligned} P(\Gamma^c_{N,k,\eta}) &\leq P(\Omega^c_{N,\eta}) + \sum_{k=1}^{K} P(\Omega^c_{n,k,\eta}) \\ &\leq n^{2\alpha-1}(O(1) + K \cdot O(1)) \end{aligned}$$

Since K=$O(n^{\gamma})$, $\gamma < 1 - 2\alpha$ and $\frac{1}{4} \leq \alpha \leq \frac{1}{2}$ by assumption, we have $\lim_{n\to\infty} P(\Gamma^c_{N,k,\eta}) \to 0$.

Next, we wish to show $\Gamma_{N,k,\eta} \subseteq \{\omega | \sqrt{N} \| \hat{\boldsymbol{\beta}}_N - \tilde{\boldsymbol{\beta}}_K \| \leq \delta\}$. Consider the Taylor expansion of $-M_{n,k}(\hat{\boldsymbol{\beta}}_N)$ at intermediary estimator $\check{\boldsymbol{\beta}}_{n,k}$:

$$-M_{n,k}(\hat{\boldsymbol{\beta}}_N) = -M_{n,k}(\check{\boldsymbol{\beta}}_{n,k}) + [\mathbf{A}_{n,k}(\check{\boldsymbol{\beta}}_{n,k})](\hat{\boldsymbol{\beta}}_N - \check{\boldsymbol{\beta}}_{n,k}) + \check{\mathbf{r}}_{n,k},$$

where $\check{\mathbf{r}}_{n,k}$ is the remainder term with j^{th} element $\frac{1}{2}(\hat{\boldsymbol{\beta}}_N - \check{\boldsymbol{\beta}}_{n,k})' \sum_{i=1}^{n} \frac{-\partial^2 \psi_j(\mathbf{z}_{ki}, \boldsymbol{\beta}^*_k)}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'} (\hat{\boldsymbol{\beta}}_N - \check{\boldsymbol{\beta}}_{n,k})$ for some $\boldsymbol{\beta}^*_k$ between $\hat{\boldsymbol{\beta}}_N$ and $\check{\boldsymbol{\beta}}_{n,k}$.

Summing over k,

$$0 = -\sum_{k=1}^{K} M_{n,k}(\hat{\boldsymbol{\beta}}_N) = -\sum_{k=1}^{K} M_{n,k}(\check{\boldsymbol{\beta}}_{n,k}) + \sum_{k=1}^{K} \mathbf{A}_{n,k}(\check{\boldsymbol{\beta}}_{n,k})(\hat{\boldsymbol{\beta}}_N - \check{\boldsymbol{\beta}}_{n,k}) + \sum_{k=1}^{K} \check{\mathbf{r}}_{n,k}.$$

Rearranging terms and recalling that $\mathbf{A}_{n,k}(\check{\boldsymbol{\beta}}_{n,k}) = \tilde{\mathbf{A}}_{n,k}$, we find

$$-\hat{\boldsymbol{\beta}}_N + (\sum_{k=1}^{K} \tilde{\mathbf{A}}_{n,k})^{-1} (\sum_{k=1}^{K} \tilde{\mathbf{A}}_{n,k} \check{\boldsymbol{\beta}}_{n,k} + \sum_{k=1}^{K} M_{n,k}(\check{\boldsymbol{\beta}}_{n,k})) = (\sum_{k=1}^{K} \tilde{\mathbf{A}}_{n,k})^{-1} \sum_{k=1}^{K} \check{\mathbf{r}}_{n,k}.$$

Using the definition of the CUEE estimator $\tilde{\boldsymbol{\beta}}_K$, the above relation reduces to

$$\tilde{\boldsymbol{\beta}}_K - \hat{\boldsymbol{\beta}}_N = (\sum_{k=1}^{K} \tilde{\mathbf{A}}_{n,k})^{-1} \sum_{k=1}^{K} \check{\mathbf{r}}_{n,k}$$

and

$$\|\tilde{\boldsymbol{\beta}}_K - \hat{\boldsymbol{\beta}}_N\| \leq \left\| \left(\frac{1}{nK} \sum_{k=1}^{K} \tilde{\mathbf{A}}_{n,k} \right)^{-1} \right\| \left\| \frac{1}{nK} \sum_{k=1}^{K} \check{\mathbf{r}}_{n,k} \right\|.$$

For the first term, according to (C4'), $\left\| \left(\frac{1}{nK} \sum_{k=1}^{K} \tilde{\mathbf{A}}_{n,k} \right)^{-1} \right\| \leq \lambda_1^{-1}$ since $\mathbf{A}_{n,k}(\boldsymbol{\beta})$ is a continuous function of $\boldsymbol{\beta}$ (according to (C1)) and $\check{\boldsymbol{\beta}}_{n,k}$ is in the neighborhood of $\boldsymbol{\beta}_0$ for small enough η. For the second term, we introduce set $B_\eta(\boldsymbol{\beta}_0) = \{\boldsymbol{\beta} | \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\| \leq \eta\}$. For all $\omega \in \Gamma_{N,k,n}$, we have $\boldsymbol{\beta}_k^* \in B_\eta(\boldsymbol{\beta}_0)$ since $B_\eta(\boldsymbol{\beta}_0)$ is a convex set and $\hat{\boldsymbol{\beta}}_N, \check{\boldsymbol{\beta}}_{n,k} \in B_\eta(\boldsymbol{\beta}_0)$. According to (C5), for small enough η, $B_\eta(\boldsymbol{\beta}_0)$ satisfies (C5) and thus $\boldsymbol{\beta}_k^*$ satisfies (C5). Hence we have $\|\check{\mathbf{r}}_{n,k}\| \leq C_2 pn \|\hat{\boldsymbol{\beta}}_N - \check{\boldsymbol{\beta}}_{n,k}\|^2$ for all $\omega \in \Gamma_{N,K,\eta}$ when η is small enough. Additionally,

$$\begin{aligned}
\|\check{\mathbf{r}}_{n,k}\| &\leq C_2 pn \|\hat{\boldsymbol{\beta}}_N - \check{\boldsymbol{\beta}}_{n,k}\|^2 \leq C_2 pn (\|\hat{\boldsymbol{\beta}}_N - \boldsymbol{\beta}_0\|^2 + \|\check{\boldsymbol{\beta}}_{n,k} - \boldsymbol{\beta}_0\|^2) \\
&\leq C_2 pn (\frac{\eta^2}{n^{2\alpha}} + \frac{\eta^2}{N^{2\alpha}}) \\
&\leq 2C_2 pn^{1-2\alpha} \eta^2.
\end{aligned}$$

Consequently,

$$\|\tilde{\boldsymbol{\beta}}_K - \hat{\boldsymbol{\beta}}_N\| \leq \frac{1}{\lambda_1} \frac{K}{nK} 2c_2 pn^{1-2\alpha} \eta^2 \leq C \frac{\eta^2}{n^{2\alpha}},$$

where $C = \frac{2C_2 p}{\lambda_1}$.

Therefore, for any $\delta > 0$, there exists $\eta_\delta > 0$ such that $C\eta_\delta^2 < \delta$. Then for any $\omega \in \Gamma_{N,k,\eta_\delta}$ and $K = O(n^\gamma)$, where $\gamma < min\{1-2\alpha, 4\alpha-1\}$, we have $\sqrt{N}\|\tilde{\boldsymbol{\beta}}_K - \hat{\boldsymbol{\beta}}_N\| \leq O(n^{\frac{1+\gamma-4\alpha}{2}})\delta$. Therefore, when n is large enough, $\Gamma_{N,k,\eta} \subseteq \{\omega \in \Omega | \sqrt{N}\|\hat{\boldsymbol{\beta}}_N - \tilde{\boldsymbol{\beta}}_K\| \leq \delta\}$ and thus $P(\sqrt{N}\|\tilde{\boldsymbol{\beta}}_K - \hat{\boldsymbol{\beta}}_N\| > \delta) \leq P(\Gamma^c_{N,k,\eta}) \to 0$ as $n \to \infty$. ■

A.6 Proof of Proposition 3.9

Suppose $\mathbf{X}_k$ does not have full column rank for some accumulation point k. For ease of exposition, write $\bar{\mathbf{W}}_k = \text{Diag}\left(S_{ki}^2 W_{ki}\right)$. Note that for generalized linear models with y_{ki} from an exponential family, $W_{ki} = 1/\text{v}(\mu_{ki})$ where $\text{v}(\mu_{ki})$ is the variance function. The IRLS approach is then implemented as follows. For $t = 1, 2, \ldots,$

$$
\begin{aligned}
\bar{\mathbf{W}}_k^{(t)} &= \text{Diag}\left((S_{ki}^2)^{(t-1)} W_{ki}^{(t-1)}\right) \\
\mathbf{Z}_k^{(t)} &= \boldsymbol{\eta}_k^{(t-1)} + \{\mathbf{S}_k^{(t-1)}\}^{-1}(\mathbf{y}_k - \boldsymbol{\mu}_k^{(t-1)}) \\
\boldsymbol{\beta}^{(t)} &= (\mathbf{X}_k' \bar{\mathbf{W}}_k^{(t-1)} \mathbf{X}_k)^- \mathbf{X}_k' \bar{\mathbf{W}}_k^{(t-1)} \mathbf{Z}_k^{(t-1)} \\
\boldsymbol{\eta}_k^{(t)} &= \mathbf{X}_k \boldsymbol{\beta}^{(t)}.
\end{aligned}
$$

As $\mathbf{X}_k$ is not of full rank, $\boldsymbol{\beta}^{(t)}$ uses a generalized inverse and is not unique. Since $\bar{\mathbf{W}}_k^{(t)}$ is a diagonal positive definite matrix, there exists an invertible matrix $\mathbf{V}$ such that

$\bar{\mathbf{W}}_k^{(t)} = \mathbf{V'V}$, where $\mathbf{V} = \sqrt{\bar{\mathbf{W}}_k^{(t)}}$. We thus have

$$\boldsymbol{\eta}_k^{(t)} = \mathbf{V}^{-1}\mathbf{V}\mathbf{X}_k\{(\mathbf{V}\mathbf{X}_k)'(\mathbf{V}\mathbf{X}_k)\}^{-}(\mathbf{V}\mathbf{X}_k)'\mathbf{V'}^{-1}\bar{\mathbf{W}}_k^{(t-1)}\mathbf{Z}_k^{(t-1)}.$$

Therefore, for $t = 1$, $\boldsymbol{\eta}_k^{(1)}$ is unique no matter what generalized inverse of $\mathbf{X}_k'\bar{\mathbf{W}}_k^{(0)}\mathbf{X}_k$ we use, given the same initial value $\mathbf{W}_k^{(0)}$. Furthermore, since $\bar{\mathbf{W}}_k^{(t)}$ and $\mathbf{Z}_k^{(t)}$ depend on $\boldsymbol{\beta}^{(t-1)}$ only through $\boldsymbol{\eta}^{(t-1)}$, $\bar{\mathbf{W}}_k^{(1)}, \mathbf{Z}_k^{(1)}$ and thus $\boldsymbol{\eta}_k^{(1)}$ are also invariant of the choice of generalized inverse. Similarly, we can show that for each iteration, $\bar{\mathbf{W}}_k^{(t)}, \mathbf{Z}_k^{(t)}$ and $\boldsymbol{\eta}_k^{(t)}$ are unique no matter what generalized inverse of $\mathbf{X}_k'\bar{\mathbf{W}}_k^{(t-1)}\mathbf{X}_k$ we use, given the same initial values.

Now, the only problem left is whether the IRLS algorithm converges. We next show that $\boldsymbol{\beta}^{(t)}$ converges under a special generalized inverse of $\mathbf{X}_k'\bar{\mathbf{W}}_k^{(t-1)}\mathbf{X}_k$. Let $\mathbf{X}_k^*$ denote a $n_k \times p^*$ full rank column submatrix of $\mathbf{X}_k$. Without loss of generality, we assume the p^* columns of $\mathbf{X}_k^*$ are the first p^* columns of $\mathbf{X}_k$. Assume $\mathbf{X}_k^*$ satisfies (C1-C3), and the IRLS estimates converge to $\hat{\boldsymbol{\beta}}_k^* = (\mathbf{X}_k^{*\prime}\bar{\mathbf{W}}_k\mathbf{X}_k^*)^{-1}\mathbf{X}_k^{*\prime}\bar{\mathbf{W}}_k\mathbf{Z}_k$, where $\boldsymbol{\beta}_k^*$ is the $p^* \times 1$ vector of regression coefficients corresponding to $\mathbf{X}_k^*$. Since $\mathbf{X}_k^*$ is a full column rank submatrix of $\mathbf{X}_k$, there exists a $p^* \times p$ matrix $\mathbf{P}$ such that $\mathbf{X}_k = \mathbf{X}_k^*\mathbf{P}$, where the first $p^* \times p^*$

submatrix is an identity matrix. We thus have,

$$\begin{aligned}
\boldsymbol{\beta}^{(t)} &= (\mathbf{P}'\mathbf{X}_k^{*\prime}\bar{\mathbf{W}}_k^{(t-1)}\mathbf{X}_k^{*}\mathbf{P})^{-}\mathbf{P}'\mathbf{X}_k^{*\prime}\bar{\mathbf{W}}_k^{(t-1)}\mathbf{Z}_k^{(t-1)} \\
&= \begin{pmatrix} \mathbf{X}_k^{*\prime}\bar{\mathbf{W}}_k^{(t-1)}\mathbf{X}_k^{*} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}^{-} \mathbf{P}'\mathbf{X}_k^{*\prime}\bar{\mathbf{W}}_k^{(t-1)}\mathbf{Z}_k^{(t-1)} \\
&= \begin{pmatrix} (\mathbf{X}_k^{*\prime}\bar{\mathbf{W}}_k^{(t-1)}\mathbf{X}_k^{*})^{-1} \\ \mathbf{0} \end{pmatrix} \mathbf{X}_k^{*\prime}\bar{\mathbf{W}}_k^{(t-1)}\mathbf{Z}_k^{(t-1)}
\end{aligned}$$

Thus, for that special generalized inverse, $\boldsymbol{\beta}^{(t)}$ converges to $\begin{pmatrix}\hat{\boldsymbol{\beta}}_k^{*} & \mathbf{0}\end{pmatrix}'$. By the uniqueness property given above, $\boldsymbol{\beta}^{(t)}$ converges no matter what generalized inverse we choose.

Upon convergence, $\boldsymbol{\beta}^{(t)} = \hat{\boldsymbol{\beta}}_{n_k,k} = (\mathbf{X}_k'\bar{\mathbf{W}}_k\mathbf{X}_k)^{-}\mathbf{X}_k'\bar{\mathbf{W}}_k\mathbf{Z}_k$ and $\mathbf{A}_{n_k,k} = \mathbf{X}_k'\bar{\mathbf{W}}_k\mathbf{X}_k$. As in the normal linear model case, $\mathbf{A}_{n_k,k}\hat{\boldsymbol{\beta}}_{n_k,k}$ is invariant to the choice of $\mathbf{A}_{n_k,k}^{-}$, as it is always $\mathbf{X}_k'\mathbf{W}_k\mathbf{Z}_k$. Therefore, the combined estimator $\hat{\boldsymbol{\beta}}_{NK}$ is invariant to the choice of generalized inverse $\mathbf{A}_{n_k,k}^{-}$ of $\mathbf{A}_{n_k,k}$. Similar arguments can be used for the online estimator $\tilde{\boldsymbol{\beta}}_K$. ■

A.7 Additional Simulations and Results

Normal Linear Regression: Residual Diagnostic Performance

Figure F.1 is the analogous version of Figure 1 for $n_{k^*} = 100$. Note that the average false discovery rate for the predictive residual test based on BH adjusted p-values was controlled in all cases except when $k^* = 2$ and $n_{k*} = 100$, representing the smallest

sample size considered.

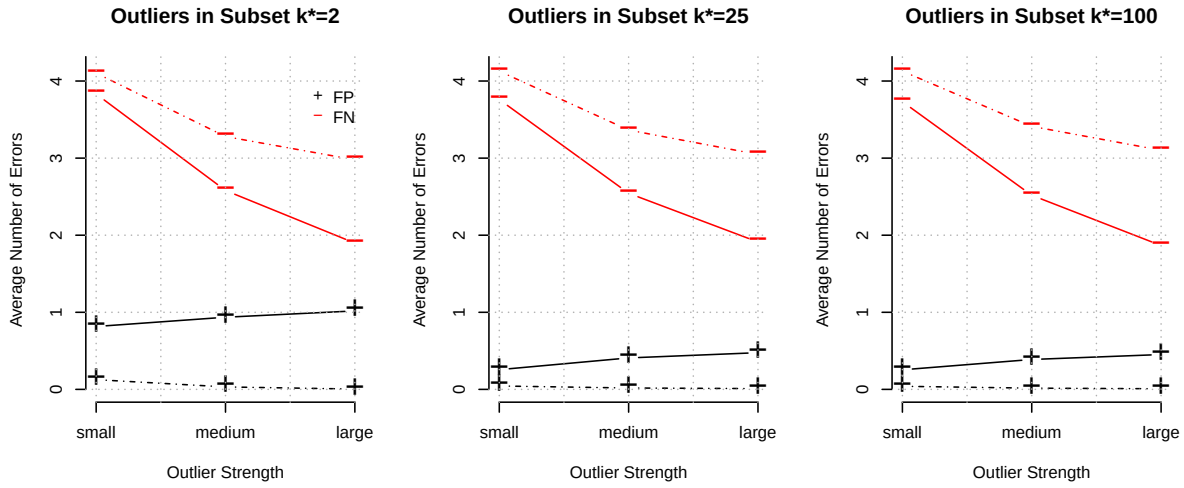

Figure F.1: Average numbers of False Positives and False Negatives for outlier t-tests for $n_{k^*} = 100$. Solid lines correspond to the predictive residual test while dotted lines correspond to the externally studentized residuals test using only data from subset k^*.

Rank Deficiency and Generalized Inverse in EE setting

Consider the CUEE estimator for a given dataset under two choices of generalized inverse, the Moore-Penrose generalized inverse, and a generalized inverse generated according to Theorem 2.1 of Rao and Mitra (1972). For this small-scale, proof-of-concept simulation, we generated $B = 100$ datasets of $y_i \sim \text{Bernoulli}(\mu_i)$, independently for $i = 1, \ldots, 20000$, with $\text{logit}(\mu_i) = \mathbf{x}_i'\boldsymbol{\beta}$ where $\boldsymbol{\beta} = (1, 1, 1, 1, 1)'$, $x_{i[2]} \sim \text{Bernoulli}(0.5)$ independently, $x_{i[3:5]} \sim N(\mathbf{0}, \mathbf{I}_3)$ independently, and $x_{ki[1]} = 1$. We fixed $K = 10$ and $n_k = 2000$. The pairs of $(y_i, \mathbf{x}_i)$ observations were considered in different orders, so that in the first ordering all subsets would result in full rank $\mathbf{A}_{n_k,k}$, $k = 1, \ldots, K$, but in the second ordering all of the subsets would not have full rank $\mathbf{A}_{n_k,k}$ due to the grouping of zeros and ones from the binary covariate. In the first ordering, we used the initially

proposed CUEE estimator $\tilde{\boldsymbol{\beta}}_K$ in (3.18) to estimate $\boldsymbol{\beta}$ and its corresponding variance $\tilde{\mathbf{V}}_K$ in (3.21). In the second ordering, we used two different generalized inverses to compute $\hat{\boldsymbol{\beta}}_{n_k,k}$, denoted by $\text{CUEE}_1^{(-)}$ and $\text{CUEE}_2^{(-)}$ in Table A.7, with variance given by $\tilde{\mathbf{A}}_K^{-1}$. The estimates reported in Table A.7 were averaged over 100 replicates. The corresponding EE estimates, which are computed by fitting all N observations simultaneously, are also provided for comparison. As expected, the values reported for $\text{CUEE}_1^{(-)}$ and $\text{CUEE}_2^{(-)}$ are identical, indicating that the estimator is invariant to the choice of generalized inverse, and these results are quite similar to those of the EE estimator and CUEE estimator with all full-rank matrices $\mathbf{A}_{n_k,k}$, $k = 1, \ldots, K$.

Table F.1: Estimates and standard errors for $\text{CUEE}_1^{(-)}$, $\text{CUEE}_2^{(-)}$, CUEE, and EE estimators. $\text{CUEE}_1^{(-)}$ and $\text{CUEE}_2^{(-)}$ correspond to CUEE estimators using two different generalized inverses for $\mathbf{A}_{n_k,k}$ when $\mathbf{A}_{n_k,k}$ is not invertible.

$\text{CUEE}_1^{(-)}$		$\text{CUEE}_2^{(-)}$		CUEE		EE	
$\tilde{\boldsymbol{\beta}}_{Kj}$	$se(\tilde{\boldsymbol{\beta}}_{Kj})$	$\tilde{\boldsymbol{\beta}}_{Kj}$	$se(\tilde{\boldsymbol{\beta}}_{Kj})$	$\tilde{\boldsymbol{\beta}}_{Kj}$	$se(\tilde{\boldsymbol{\beta}}_{Kj})$	$\hat{\boldsymbol{\beta}}_{Nj}$	$se(\hat{\boldsymbol{\beta}}_{Nj})$
0.9935731	0.02850429	0.9935731	0.02850429	0.9940272	0.02847887	0.9951570	0.02845648
0.8902375	0.03970919	0.8902375	0.03970919	0.8923991	0.03936931	0.8933344	0.03935490
0.9872035	0.02256396	0.9872035	0.02256396	0.9879017	0.02247598	0.9891857	0.02245082
0.9916863	0.02264102	0.9916863	0.02264102	0.9925716	0.02248187	0.9938864	0.02246949
0.9874042	0.02260353	0.9874042	0.02260353	0.9882167	0.02247671	0.9895110	0.02244759

Appendix B

New Variables Supplementation

B.1 Derivations for the Linear Model

Continuing with Section 4.2.3, to solve $\mathsf{Var}(\tilde{\boldsymbol{\beta}}_{K+1})$, we can decompose (4.6) and let

$$W_1 = (\mathsf{Var}^{-1}(\tilde{\boldsymbol{\beta}}_K) + \mathsf{Var}^{-1}(\hat{\boldsymbol{\beta}}_{K+1}))^{-1}\mathsf{Var}^{-1}(\tilde{\boldsymbol{\beta}}_K),$$

$$W_2 = (\mathsf{Var}^{-1}(\tilde{\boldsymbol{\beta}}_K) + \mathsf{Var}^{-1}(\hat{\boldsymbol{\beta}}_{K+1}))^{-1}\mathsf{Var}^{-1}(\hat{\boldsymbol{\beta}}_{K+1}),$$

thus,

$$\begin{aligned}\mathsf{Var}(\tilde{\boldsymbol{\beta}}_{K+1}) =& \mathsf{Var}(W_1\tilde{\boldsymbol{\beta}}_K + W_2\hat{\boldsymbol{\beta}}_{K+1}) \\ =& W_1\mathsf{Var}(\tilde{\boldsymbol{\beta}}_K)W_1^\top + W_1\mathsf{Cov}(\tilde{\boldsymbol{\beta}}_K, \hat{\boldsymbol{\beta}}_{K+1})W_2^\top \\ &+ W_2\mathsf{Cov}(\hat{\boldsymbol{\beta}}_{K+1}, \tilde{\boldsymbol{\beta}}_K)W_1^\top + W_2\mathsf{Var}(\hat{\boldsymbol{\beta}}_{K+1})W_2^\top \end{aligned} \tag{F.1}$$

where the estimate of $\mathsf{Var}(\tilde{\boldsymbol{\beta}}_K)$ is solved in Section 4.2.2 and $\mathsf{Var}(\hat{\boldsymbol{\beta}}_{K+1})$ is estimated at block $K+1$ as $\hat{\mathsf{Var}}(\hat{\boldsymbol{\beta}}_{K+1})$. By decomposing $\tilde{\boldsymbol{\beta}}_K$, the covariance between cumulative and

current estimators of $\boldsymbol{\beta}$ can be decomposed as

$$\mathsf{Cov}(\tilde{\boldsymbol{\beta}}_K, \hat{\boldsymbol{\beta}}_{K+1}) = \mathsf{Var}(\hat{\boldsymbol{\beta}}_{K+1}) - \mathsf{Cov}(\hat{\mathbf{b}}_{K+1}, \hat{\boldsymbol{\beta}}_{K+1}).$$

By the law of total covariance, we have

$$\begin{aligned}\mathsf{Cov}(\hat{\mathbf{b}}_{K+1}, \hat{\boldsymbol{\beta}}_{K+1}) =& E[\mathsf{Cov}(\hat{\mathbf{b}}_{K+1}, \hat{\boldsymbol{\beta}}_{K+1})|\mathbf{X}_{K+1}, \mathbf{Z}_{K+1}] \\ &+ \mathsf{Cov}[E(\hat{\mathbf{b}}_{K+1}|\mathbf{X}_{K+1}, \mathbf{Z}_{K+1}), E(\hat{\boldsymbol{\beta}}_{K+1}|\mathbf{X}_{K+1}, \mathbf{Z}_{K+1})].\end{aligned}$$

According to Clogg et al. (1995), the first component is equal to $\mathsf{Var}(\hat{\mathbf{b}}_{K+1}|H_R)\sigma^2_\nu/\sigma^2_\epsilon$, which can be estimated by $\hat{\mathsf{Var}}(\hat{\mathbf{b}}_{K+1}|H_R)\hat{\sigma}^2_{\nu,K+1}/\tilde{\sigma}^2_{\epsilon,K+1}$ from the sample while the second component is zero. Alternatively, the first component can be estimated directly by using linear algebra which will be demonstrated later.

The other quantity to solve is $\mathsf{Cov}(\tilde{\boldsymbol{\beta}}_{K+1}, \tilde{\boldsymbol{\theta}}_{K+1})$. We can decompose $\tilde{\boldsymbol{\beta}}_{K+1}$ and finally get

$$\mathsf{Cov}(\tilde{\boldsymbol{\beta}}_{K+1}, \tilde{\boldsymbol{\theta}}_{K+1}) = (W_1 + W_2)\mathsf{Cov}(\hat{\boldsymbol{\beta}}_{K+1}, \hat{\boldsymbol{\theta}}_{K+1}) - W_1\mathsf{Cov}(\hat{\mathbf{b}}_{K+1}, \hat{\boldsymbol{\theta}}_{K+1}),$$

where $\mathsf{Cov}(\hat{\boldsymbol{\beta}}_{K+1}, \hat{\boldsymbol{\theta}}_{K+1})$ is estimated as part of the variance-covariance matrix of coefficients at block $K+1$. For $\mathsf{Cov}(\hat{\mathbf{b}}_{K+1}, \hat{\boldsymbol{\theta}}_{K+1})$, we have

$$\begin{aligned}\mathsf{Cov}(\hat{\mathbf{b}}_{K+1}, \hat{\boldsymbol{\theta}}_{K+1}) =& E[\mathsf{Cov}(\hat{\mathbf{b}}_{K+1}, \hat{\boldsymbol{\theta}}_{K+1})|\mathbf{X}_{K+1}, \mathbf{Z}_{K+1}] \\ &+ \mathsf{Cov}[E(\hat{\mathbf{b}}_{K+1}|\mathbf{X}_{K+1}, \mathbf{Z}_{K+1}), E(\hat{\boldsymbol{\theta}}_{K+1}|\mathbf{X}_{K+1}, \mathbf{Z}_{K+1})].\end{aligned}$$

The second component is zero as well, but the first one takes more efforts to solve. From the Linear Model,

$$\hat{\mathbf{b}}_{K+1} = (\mathbf{X}_{K+1}^\top \mathbf{X}_{K+1})^{-1} \mathbf{X}_{K+1}^\top \mathbf{y}_{K+1},$$

and

$$\begin{pmatrix} \hat{\boldsymbol{\beta}}_{K+1} \\ \hat{\boldsymbol{\theta}}_{K+1} \end{pmatrix} = \begin{pmatrix} \mathbf{X}_{K+1}^\top \mathbf{X}_{K+1} & \mathbf{X}_{K+1}^\top \mathbf{Z}_{K+1} \\ \mathbf{Z}_{K+1}^\top \mathbf{X}_{K+1} & \mathbf{Z}_{K+1}^\top \mathbf{Z}_{K+1} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{X}_{K+1}^\top \\ \mathbf{Z}_{K+1} \end{pmatrix} \mathbf{y}_{K+1}.$$

Let

$$\mathbf{B} = \begin{pmatrix} \mathbf{X}_{K+1}^\top \mathbf{X}_{K+1} & \mathbf{X}_{K+1}^\top \mathbf{Z}_{K+1} \\ \mathbf{Z}_{K+1}^\top \mathbf{X}_{K+1} & \mathbf{Z}_{K+1}^\top \mathbf{Z}_{K+1} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{pmatrix},$$

where $\mathbf{B}_{11}$ is $p \times p$, $\mathbf{B}_{12}$ is $p \times q$, $\mathbf{B}_{21}$ is $q \times p$, and $\mathbf{B}_{22}$ is $q \times q$. All of them have closed-forms. So,

$$\begin{pmatrix} \hat{\boldsymbol{\beta}}_{K+1} \\ \hat{\boldsymbol{\theta}}_{K+1} \end{pmatrix} = \begin{pmatrix} \mathbf{B}_{11}\mathbf{X}_{K+1}^\top + \mathbf{B}_{12}\mathbf{Z}_{K+1}^\top \\ \mathbf{B}_{21}\mathbf{X}_{K+1}^\top + \mathbf{B}_{22}\mathbf{Z}_{K+1}^\top \end{pmatrix} \mathbf{y}_{K+1}.$$

Let

$$\mathbf{C} = \mathbf{B}_{11}\mathbf{X}_{K+1}^\top + \mathbf{B}_{12}\mathbf{Z}_{K+1}^\top,$$

$$\mathbf{D} = \mathbf{B}_{21}\mathbf{X}_{K+1}^\top + \mathbf{B}_{22}\mathbf{Z}_{K+1}^\top,$$

$$\mathbf{E} = (\mathbf{X}_{K+1}^\top \mathbf{X}_{K+1})^{-1} \mathbf{X}_{K+1}^\top.$$

The separate least square estimators of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ at block $k+1$ are

$$\hat{\boldsymbol{\beta}}_{K+1} = \mathbf{C}\mathbf{y}_{K+1}, \quad \hat{\boldsymbol{\theta}}_{K+1} = \mathbf{D}\mathbf{y}_{K+1}.$$

Thus, we have

$$E[\mathsf{Cov}(\hat{\mathbf{b}}_{K+1}, \hat{\boldsymbol{\beta}}_{K+1} | \mathbf{X}_{K+1}, \mathbf{Z}_{K+1})] = E[\mathsf{Cov}(\mathbf{E}\mathbf{y}_{K+1}, \mathbf{C}\mathbf{y}_{K+1})] = E[\sigma_\nu^2 \mathbf{E}\mathbf{C}^\top],$$
$$E[\mathsf{Cov}(\hat{\mathbf{b}}_{K+1}, \hat{\boldsymbol{\theta}}_{K+1} | \mathbf{X}_{K+1}, \mathbf{Z}_{K+1})] = E[\mathsf{Cov}(\mathbf{E}\mathbf{y}_{K+1}, \mathbf{D}\mathbf{y}_{K+1})] = E[\sigma_\nu^2 \mathbf{E}\mathbf{D}^\top],$$

where $E[\sigma_\nu^2 \mathbf{E}\mathbf{C}^\top]$ can be estimated by $\hat{\sigma}^2_{\nu,k+1}\mathbf{E}\mathbf{C}^\top$, $E[\sigma_\nu^2 \mathbf{E}\mathbf{D}^\top]$ can be estimated by $\hat{\sigma}^2_{\nu,k+1}\mathbf{E}\mathbf{D}^\top$ from the sample. Finally the variance covariance matrix of $(\tilde{\boldsymbol{\beta}}_{K+1}, \tilde{\boldsymbol{\theta}}_{K+1})$ can be estimated with the formulae above and $\hat{W}_1, \hat{W}_2$, the estimates of W_1 and W_2. Updating with future blocks just follows the updating algorithms in Schifano et al. (2016).

B.2 Proof of Proposition 4.1

In this section, we present the proof for Proposition 4.1, the improvement of new method under the linear regression setting in the three-variable-and-two-block case. For simplicity, both x and z are centered.

Define $S^2_{x_1} = \sum_{i=1}^{n_1} x^2_{1i}/n_1$ and $S^2_{x_2} = \sum_{i=1}^{n_2} x^2_{2i}/n_2$. We assume that variable x has

standard deviation σ_x and the correlation between x and z is ρ_{xz}. Thus, we get

$$\mathsf{Var}(\hat{b}_1|H_R) = \mathsf{E}[\mathsf{Var}(\hat{b}_1|x_1)] + \mathsf{Var}[\mathsf{E}(\hat{b}_1|x_1)] = \mathsf{E}\left[\frac{\sigma_\epsilon^2}{n_1 S_{x_1}^2}\right] + \mathsf{Var}(b) = \frac{\sigma_\epsilon^2}{n_1}\mathsf{E}\left[\frac{1}{S_{x_1}^2}\right].$$

Similarly we get $\mathsf{Var}(\hat{b}_2|H_R) = \mathsf{E}[1/S_{x_2}^2]\sigma_\epsilon^2/n_2$, and $\mathsf{Var}(\hat{\beta}_2) = \mathsf{E}[1/S_{x_2}^2]\sigma_\nu^2/(n_2(1-\rho_{xz}^2))$.

To utilize the correction method by Allison (1995), we need to regress z on x. The variance of $\hat{b}_{zx}$, the coefficient estimate of x in regressing z on x is $\mathsf{Var}(\hat{b}_{zx}) = \mathsf{E}[1/S_{x_2}^2]\sigma_\eta^2/n_2$. Since x has the standard deviation σ_x, $\mathsf{E}[1/S_{x_1}^2]$ and $\mathsf{E}[1/S_{x_2}^2]$ are equal and can be determined by the distribution of x, denoted by $\mathsf{E}[1/S_{x_1}^2] = \mathsf{E}[1/S_{x_2}^2] = 1/\Lambda_x$.

Following the derivations in Appendix B.1, we get their simplified formulae in the three-variable-and-two-block case,

$$\begin{aligned}
\mathsf{Var}(\hat{b}_1|H_F) &= \frac{\sigma_\nu^2}{n_1\Lambda_x} + \theta_2^2\frac{\sigma_\eta^2}{n_2\Lambda_x}, \\
\mathsf{Var}(\hat{\delta}_2) &= \frac{\sigma_\nu^2}{n_2\Lambda_x(1-\rho_{xz}^2)} - \frac{\sigma_\nu^2}{n_2\Lambda_x} + \theta_2^2\frac{\sigma_\eta^2}{n_2\Lambda_x}, \\
\mathsf{Var}(\tilde{\beta}_1) &= \frac{\sigma_\nu^2}{n_1\Lambda_x} + \theta_2^2\frac{\sigma_\eta^2}{n_2\Lambda_x} + \theta_2^2\frac{\sigma_\eta^2}{n_1\Lambda_x} + \frac{\rho_{xz}^2\sigma_\nu^2}{n_2\Lambda_x(1-\rho_{xz}^2)}. \\
&= \mathsf{Var}(\tilde{\beta}_1) = \frac{\sigma_\nu^2(n_2(1-\rho_{xz}^2) + n_1\theta^2\frac{\sigma_\eta^2}{\sigma_\nu^2}(1-\rho_{xz}^2) + n_2\theta^2\frac{\sigma_\eta^2}{\sigma_\nu^2}(1-\rho_{xz}^2) + n_1\rho_{xz}^2)}{n_1 n_2\Lambda_x(1-\rho_{xz}^2)}.
\end{aligned}$$

According to Equation (F.1), we have

$$\mathsf{Var}(\tilde{\beta}_2) = W_1^2\mathsf{Var}(\tilde{\beta}_1) + W_2^2\mathsf{Var}(\hat{\beta}_2) + 2W_1W_2\mathsf{Cov}(\tilde{\beta}_1, \hat{\beta}_2),$$

where

$$W_1 = (\mathsf{Var}^{-1}(\tilde{\beta}_1) + \mathsf{Var}^{-1}(\hat{\beta}_2))^{-1}\mathsf{Var}^{-1}(\tilde{\beta}_1),$$

$$W_2 = (\mathsf{Var}^{-1}(\tilde{\beta}_1) + \mathsf{Var}^{-1}(\hat{\beta}_2))^{-1}\mathsf{Var}^{-1}(\hat{\beta}_2),$$

$$\mathsf{Cov}(\tilde{\beta}_1, \hat{\beta}_2) = \mathsf{Var}(\hat{\beta}_2) - \mathsf{Var}(\hat{b}_2|H_R)\sigma_\nu^2/\sigma_\epsilon^2 = \frac{\rho_{xz}^2\sigma_\nu^2}{n_2\Lambda_x(1-\rho_{xz}^2)}.$$

Leting

$$\Delta = (1-\rho_{xz}^2) + \frac{n_1}{n_2}\theta^2\frac{\sigma_\eta^2}{\sigma_\nu^2}(1-\rho_{xz}^2) + \theta^2\frac{\sigma_\eta^2}{\sigma_\nu^2}(1-\rho_{xz}^2) + \frac{n_1}{n_2}\rho_{xz}^2,$$

we get

$$\mathsf{Var}(\tilde{\beta}_2) = \left(\frac{\frac{n_1}{n_2}}{\frac{n_1}{n_2}+\Delta}\right)^2\frac{\sigma_\nu^2\Delta}{n_1\Lambda_x(1-\rho_{xz}^2)} + \left(\frac{\Delta}{\frac{n_1}{n_2}+\Delta}\right)^2\frac{\sigma_\nu^2}{n_2\Lambda_x(1-\rho_{xz}^2)} + \frac{2\frac{n_1}{n_2}\Delta}{(\frac{n_1}{n_2}+\Delta)^2}\frac{\rho_{xz}^2\sigma_\nu^2}{n_2\Lambda_x(1-\rho_{xz}^2)}.$$

Our goal is find the conditions for

$$\frac{\mathsf{Var}(\hat{\beta}_2)}{\mathsf{Var}(\tilde{\beta}_2)} = \frac{(\frac{n_1}{n_2}+\Delta)^2}{\frac{n_1}{n_2}\Delta + \Delta^2 + 2\frac{n_1}{n_2}\Delta\rho_{xz}^2} \geq 1. \tag{F.2}$$

By expanding Δ and some rearrangements, we discuss the following cases.

When $0 \leq \rho_{xz}^2 < 0.5$, we need

$$\theta^2 \geq \frac{n_2(2\rho_{xz}^2-1) - n_1(2\rho_{xz}^2+1)}{(n_1+n_2)(1-2\rho_{xz}^2)}\frac{\sigma_\nu^2}{\sigma_\eta^2} = -\frac{n_2}{n_1+n_2}\frac{\sigma_\nu^2}{\sigma_\eta^2} - \frac{n_1}{n_1+n_2}\frac{2\rho_{xz}^2+1}{1-2\rho_{xz}^2}\frac{\sigma_\nu^2}{\sigma_\eta^2}.$$

Because

$$-\frac{n_2}{n_1+n_2}\frac{\sigma_\nu^2}{\sigma_\eta^2}-\frac{n_1}{n_1+n_2}\frac{2\rho_{xz}^2+1}{1-2\rho_{xz}^2}\frac{\sigma_\nu^2}{\sigma_\eta^2}<0$$

and $\theta^2 \geq 0$, the inequality always holds.

When $\rho_{xz}^2 = 0.5$, we need

$$(\frac{n_1}{n_2}+\Delta)^2 \geq \frac{n_1}{n_2}\Delta+\Delta^2+2\frac{n_1}{n_2}\Delta\rho_{xz}^2,$$

where $\rho_{xz}^2 = 0.5$. After simplification, that is $n_1/n_2 \geq 0$ which is always true.

When $0.5 < \rho_{xz}^2 < 1$,

$$\theta^2 \leq \frac{n_2(2\rho_{xz}^2-1)-n_1(2\rho_{xz}^2+1)}{(n_1+n_2)(1-2\rho_{xz}^2)}\frac{\sigma_\nu^2}{\sigma_\eta^2}=\frac{n_1}{n_1+n_2}\frac{2\rho_{xz}^2+1}{2\rho_{xz}^2-1}\frac{\sigma_\nu^2}{\sigma_\eta^2}-\frac{n_2}{n_1+n_2}\frac{\sigma_\nu^2}{\sigma_\eta^2}.$$

In this case, θ^2 has an upper limit to ensure that $\mathsf{Var}(\tilde{\beta}_2)$ is less than $\mathsf{Var}(\hat{\beta}_2)$. The limit is strict especially when ρ_{xz}^2 is close to 1 or n_2 is larger than n_1. The equality in Equation (F.2) holds when

$$\theta^2=\frac{n_1}{n_1+n_2}\frac{2\rho_{xz}^2+1}{2\rho_{xz}^2-1}\frac{\sigma_\nu^2}{\sigma_\eta^2}-\frac{n_2}{n_1+n_2}\frac{\sigma_\nu^2}{\sigma_\eta^2},\quad 0.5<\rho_{xz}^2<1.$$

When $\rho_{xz}^2 = 1$, the variances and covariances do not exist.

Bibliography

Adler, D., Glser, C., Nenadic, O., Oehlschlgel, J., and Zucchini, W. (2014), *ff: Memory-efficient Storage of Large Data on Disk and Fast Access Functions*, r package version 2.2-13.

Allison, P. D. (1995), "The Impact of Random Predictors on Comparisons of Coefficients between Models: Comment on Clogg, Petkova, and Haritou," *American Journal of Sociology*, 100, 1294–1305.

Andrieu, C. and Roberts, G. O. (2009), "The Pseudo-Marginal Approach for Efficient Monte Carlo Computations," *The Annals of Statistics*, 697–725.

Bates, D., Francois, R., and Eddelbuettel, D. (2014), *RcppEigen: Rcpp Integration for the Eigen Templated Linear Algebra Library*, r package version 0.3.2.2.0.

Benjamini, Y. and Hochberg, Y. (1995), "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing," *Journal of the Royal Statistical Society. Series B*, 57, 289–300.

Benjamini, Y. and Yekutieli, D. (2001), "The Control of the False Discovery Rate in Multiple Testing under Dependency," *Annals of Statistics*, 29, 1165–1188.

Bickel, P. J., Götze, F., and van Zwet, W. R. (1997), "Resampling Fewer than n Observations: Gains, Losses, and Remedies for Losses," *Statistica Sinica*, 7, 1–31.

Blackford, L. S., Choi, J., Cleary, A., D'Azevedo, E., Demmel, J., Dhillon, I., Dongarra, J., Hammarling, S., Henry, G., Petitet, A., Stanley, K., Walker, D., and Whaley, R. C. (1997), *ScaLAPACK Users' Guide*, Philadelphia, PA: Society for Industrial and Applied Mathematics.

Buckner, J., Seligman, M., and Wilson, J. (2013), *gputools: A few GPU Enabled Functions*, r package version 0.28.

Calderhead, B. (2014), "A General Construction for Parallelizing Metropolis–Hastings Algorithms," *Proceedings of the National Academy of Sciences*, 111, 17408–17413.

Certo, S. T. (2003), "Influencing Initial Public Offering Investors with Prestige: Signaling with Board Structures," *Academy of Management Review*, 28, 432–446.

Chambers, J. (2014), "Interfaces, Efficiency and Big Data," 2014 UseR! International R User Conference.

Chapman, B., Jost, G., and Van Der Pas, R. (2008), *Using OpenMP: Portable Shared Memory Parallel Programming*, Cambridge, MA: MIT Press.

Chen, K., Hu, I., and Ying, Z. (1999), "Strong consistency of maximum quasi-likelihood estimators in generalized linear models with fixed and adaptive designs," *The Annals of Statistics*, 27, 1155–1163.

Chen, X. and Xie, M.-g. (2014), "A Split-and-Conquer Approach for Analysis of Extraordinarily Large Data," *Statistica Sinica*, forthcoming.

Clogg, C. C., Petkova, E., and Haritou, A. (1995), "Statistical Methods for Comparing Regression Coefficients between Models." *American Journal of Sociology*, 100, 1261–1293.

Cohen, R. and Rodriguez, R. (2013), "High Performance Statistical Modeling," Technical Report 401–2013, SAS Global Forum.

Cooper, N. (2014), *bigpca: PCA, Transpose and Multicore Functionality for big.matrix Objects*, r package version 1.0.

Dagum, L. and Enon, R. (1998), "OpenMP: An Industry Standard API for Shared-memory Programming," *Computational Science & Engineering, IEEE*, 5, 46–55.

Davidian, M. (2013), "Aren't We Data Science," *Amstat News*, 433, 3–5.

de Villar, F. and Rubio, A. (2014), *GUIProfiler: Profiler Graphical User Interface*, r package version 0.1.2.

Dean, J. and Ghemawat, S. (2008), "MapReduce: Simplified Data Processing on Large Clusters," *Commun. ACM*, 51, 107–113.

DeGroot, M. and Schevish, M. (2012), *Probability and Statistics*, Boston, MA: Pearson Education.

Desyllas, P. and Sako, M. (2013), "Profiting from Business Model Innovation: Evidence from Pay-As-You-Drive Auto Insurance," *Research Policy*, 42, 101–116.

Diebold, F. X. (2012), "A Personal Perspective on the Origin(s) and Development of "Big Data": The Phenomenon, the Term, and the Discipline, Second Version," Pier working paper archive, Penn Institute for Economic Research, Department of Economics, University of Pennsylvania.

Eddelbuettel, D. (2013), *Seamless R and C++ Integration with Rcpp*, Springer.

— (2014), "CRAN Task View: High-performance and Parallel Computing with R," Https://cran.r-project.org/web/views/HighPerformanceComputing.html.

Eddelbuettel, D. and Francois, R. (2014), *RInside: C++ Classes to Embed R in C++ Applications*, r package version 0.2.11.

Eddelbuettel, D., François, R., Allaire, J., Chambers, J., Bates, D., and Ushey, K. (2011), "Rcpp: Seamless R and C++ Integration," *Journal of Statistical Software*, 40, 1–18.

Eddelbuettel, D. and Sanderson, C. (2014), "RcppArmadillo: Accelerating R with High-performance C++ Linear Algebra," *Computational Statistics and Data Analysis*, 71, 1054–1063.

Efron, B. (1979), "Bootstrap Methods: Another Look at the Jackknife," *The annals of Statistics*, 7, 1–26.

Emerson, J. W. and Kane, M. J. (2012), "Don't Drown in the Data," *Significance*, 9, 38–39.

— (2013a), *biganalytics: A Library of Utilities for big.matrix Objects of Package bigmemory*, r package version 1.1.1.

— (2013b), *bigtabulate: table-, tapply-, and Split-like Functionality for Matrix and big.matrix Objects.*, r package version 1.1.2.

Enea, M. (2014), *speedglm: Fitting Linear and Generalized Linear Models to Large Data Sets.*, r package version 0.2-1.0.

Fan, J., Han, F., and Liu, H. (2014), "Challenges of Big Data Analysis," *National Science Review*, 1, 293–314.

Fan, J. and Lv, J. (2008), "Sure Independence Screening for Ultrahigh Dimensional Feature Space," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70, 849–911.

— (2011), "Non-concace Penalized Likelihood with NP-dimensionality," *IEEE Transactions on Information Theory*, 57, 5467–5484.

Fan, W. and Bifet, A. (2013), "Mining Big Data: Current Status, and Forecast to the Future," *ACM SIGKDD Explorations Newsletter*, 14, 1–5.

Gaujoux, R. (2014), *doRNG: Generic Reproducible Parallel Backend for foreach Loops*, r package version 1.6.

Green, P. (1984), "Iteratively Reweighted Least Squares for Maximum Likelihood Estimation, and some Robust and Resistant Alternatives," *Journal of the Royal Statistical Society Series B*, 46, 149–192.

Greene, W. H. (2003), *Econometric Analysis*, Pearson Education India.

Grothendieck, G. (2014), *sqldf: Perform SQL Selects on R Data Frames*, r package version 0.4-7.1.

Guha, S., Hafen, R., Rounds, J., Xia, J., Li, J., Xi, B., and Cleveland, W. S. (2012), "Large Complex Data: Divide and Recombine (D&R) with RHIPE," *Stat*, 1.

Halekoh, U., Hjsgaard, S., and Yan, J. (2006), "The R Package geepack for Generalized Estimating Equations," *Journal of Statistical Software*, 15, 1–11.

Hood, L., Heath, J. R., Phelps, M. E., and Lin, B. (2004), "Systems Biology and New Technologies Enable Predictive and Preventative Medicine," *Science*, 306, 640–643.

IBM (2014), "Apply SPSS Analytics Technology to Big Data," Http://www.ibm.com/developerworks/library/bd-spss/.

Jonge, E. d., Wijffels, J., and van der Laan, J. (2014), *ffbase: Basic Statistical Functions for Package ff*, r package version 0.11.3.

Jordan, J. M. and Lin, D. K. J. (2014), "Statistics for Big Data: Are Statisticians Ready for Big Data?" *International Chinese Statistical Association Bulletin*, 26, 59–66.

Jordan, M. I. (2013), "On Statistics, Computation and Scalability," *Bernoulli*, 19, 1378–1390.

Kane, M. J., Emerson, J., and Weston, S. (2013), "Scalable Strategies for Computing with Massive Data," *Journal of Statistical Software*, 55, 1–19.

Kane, M. J., Lewis, B., and Emerson, J. W. (2014), *bigalgebra: BLAS Routines for Native R Matrices and big.matrix Objects*, r package version 0.8.4.

Karau, H., Konwinski, A., Wendell, P., and Zaharia, M. (2015), *Learning Spark: Lightning-Fast Big Data Analysis*, O'Reilly Media.

Kleiner, A., Talwalkar, A., Sarkar, P., and Jordan, M. I. (2014), "A Scalable Bootstrap for Massive Data," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76, 795–816.

Knaus, J. (2013), *snowfall: Easier Cluster Computing (Based on snow)*, r package version 1.84-6.

Laney, D. (2001), "3D Data Management: Controlling Data Volume, Velocity, and Variety," Research note, META Group.

L'Ecuyer, P., Simard, R., Chen, E. J., and Kelton, W. D. (2002), "An Object-oriented Random-number Package with Many Long Streams and Substreams," *Operations research*, 50, 1073–1075.

Li, N. (2010), *R Interface to SPRNG (Scalable Parallel Random Number Generators)*, r package version 1.0.

Liang, F., Cheng, Y., Song, Q., Park, J., and Yang, P. (2013), "A Resampling-based Stochastic Approximation Method for Analysis of Large Geostatistical Data," *Journal of the American Statistical Association*, 108, 325–339.

Liang, F. and Kim, J. (2013), "A Bootstrap Metropolis–Hastings Algorithm for Bayesian Analysis of Big Data," Tech. rep., Department of Statistics, Texas A & M University.

Lim, A., Breiman, L., and Cutler, A. (2014), *bigrf: Big Random Forests: Classification and Regression Forests for Large Data Sets*, r package version 0.1-11.

Lin, N. and Xi, R. (2011), "Aggregated Estimating Equation Estimation," *Statistics and Its Interface*, 4, 73–83.

Long, J. (2012), *An R Language Segue into Parallel Processing on Amazon's Web Services*, r package version 0.05.

Lumley, T. (2013), *biglm: Bounded Memory Linear and Generalized Linear Models*, r package version 0.9-1.

Ma, P., Mahoney, M. W., and Yu, B. (2013), "A Statistical Perspective on Algorithmic Leveraging," *arXiv preprint arXiv:1306.5362*.

Ma, P. and Sun, X. (2014), "Leveraging for Big Data Regression," *WIREs Computational Statistics*, 7, 70–76.

Maclaurin, D. and Adams, R. P. (2014), "Firefly Monte Carlo: Exact MCMC with Subsets of Data," *arXiv preprint arXiv:1403.5693*.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., and Byers, A. H. (2011), "Big Data: The Next Frontier for Innovation, Competition, and Productivity," Tech. rep., McKinsey Global Institute.

Mascagni, M. and Srinivasan, A. (2000), "Algorithm 806: SPRNG: A Scalable Library for Pseudorandom Number Generation," *ACM Transactions on Mathematical Software (TOMS)*, 26, 436–461.

Mashey, J. (1998), "Big Data and the Next Wave of InfraStress," Usenix.org.

McCallum, Q. E. and Weston, S. (2011), *Parallel R: Data Analysis in the Distributed World*, O'Reilly Media.

Meinshausen, N. and Buhlmann, P. (2010), "Stability Selection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72, 417–473.

Mersmann, O. (2014), *microbenchmark: Accurate Timing Functions*, r package version 1.4-2.

Miller, A. J. (1992), "Algorithm AS 274: Least Squares Routines to Supplement Those of Gentleman," *Applied Statistics*, 458–478.

Miroshnikov, A. and Conlon, E. M. (2014), "ParallelMCMCcombine: An R Package for Bayesian Methods for Big Data and Analytics," *PloS ONE*, 9, e108425.

Muirhead, R. J. (2009), *Aspects of multivariate statistical theory*, John Wiley & Sons.

Neiswanger, W., Wang, C., and Xing, E. (2013), "Asymptotically Exact, Embarrassingly Parallel MCMC," *arXiv preprint arXiv:1311.4780*.

Nishii, R. (1984), "Asymptotic Properties of Criteria for Selection of Variables in Multiple Regression," *The Annals of Statistics*, 12, 758–765.

Ostrouchov, G., Chen, W.-C., Schmidt, D., and Patel, P. (2012), "Programming with Big Data in R," .

Pacheco, P. S. (1997), *Parallel Programming with MPI*, Morgan Kaufmann.

Peng, R. D. (2006), "Interacting with Data Using the filehash Package," *R News*, 6, 19–24.

Pfeifer, B., Wittelsbuerger, U., Ramos-Onsins, S. E., and Lercher, M. J. (2014), "PopGenome: An Efficient Swiss Army Knife for Population Genomic Analyses in R," *Molecular Biology and Evolution*, 31, 1929–1936.

Politis, D. N., Romano, J. P., and Wolf, M. (1999), *Subsampling*, New York: Springer.

Quiroz, M., Villani, M., and Kohn, R. (2014), "Speeding Up MCMC by Efficient Data Subsampling," *ArXiv e-prints*.

— (2015), "Scalable MCMC for Large Data Problems using Data Subsampling and the Difference Estimator," *ArXiv e-prints*.

R Core Team (2014a), *R: A Language and Environment for Statistical Computing*, Vienna, Austria.

— (2014b), *Writing R Extensions*, Vienna, Austria.

R Special Interest Group on Databases (2014), *DBI: R Database Interface*, r package version 0.3.1.

Rabinovich, M., Angelino, E., and Jordan, M. I. (2015), "Variational Consensus Monte Carlo," *ArXiv e-prints*.

Revolution Analytics (2013), *RevoScaleR 7.0 User's Guide*, Mountain View, CA.

— (2014), "RHadoop," Https://github.com/RevolutionAnalytics/RHadoop/wiki.

Revolution Analytics and Weston, S. (2014), *foreach: foreach Looping Construct for R*, r package version 1.4.2.

Rickert, J. (2013), "Statisticians: An Endangered Species?" Http://blog.revolutionanalytics.com/2013/08/statisticians-contemplate-their-own-extinction.html.

Rodriguez, R. (2012), "Big Data and Better Data," *Amstat News*, 420, 3–4.

Rossini, A. J., Tierney, L., and Li, N. (2007), "Simple Parallel Statistical Computing in R," *Journal of Computational and Graphical Statistics*, 16, 399–420.

Rudin, C., Dunson, D., Irizarry, R., Ji, H., Laber, E., Leek, J., McCormick, T., Rose, S., Schafer, C., van der Laan, M., Wasserman, L., and Xue, L. (2014), "Discovery with Data: Leveraging Statistics with Computer Science to Transform Science and Society," White paper, American Statistical Association.

Rupp, N. G. (2007), "Further Investigations into the Causes of Flight Delays," Tech. rep., Department of Economy, East Carolina University.

Schenker, N., Davidian, M., and Rodriguez, R. (2013), "The ASA and Big Data," *Amstat News*, 432, 3–4.

Schifano, E. D., Wu, J., Wang, C., Yan, J., and Chen, M.-H. (2016), "Online Updating of Statistical Inference in the Big Data Setting," *Technometrics*, In press.

Schmidberger, M., Morgan, M., Eddelbuettel, D., Yu, H., Tierney, L., and Mansmann, U. (2009), "State of the Art in Parallel Computing with R," *Journal of Statistical Software*, 31, 1–27.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *The Annals of Statistics*, 6, 461–464.

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H., George, E., and McCulloch, R. (2013), "Bayes and Big Data: The Consensus Monte Carlo Algorithm," in *EFaBBayes 250 conference*, vol. 16.

Searle, S. (1971), *Linear Models*, New York-London-Sydney-Toronto: John Wiley and Sons, Inc.

Seligman, M., Fraley, C., and Hesterberg, T. (2011), *biglars: Scalable Least-angle Regression and Lasso*, r package version 1.0.2.

Sevcikova, H. and Rossini, A. J. (2012a), *snowFT: Fault Tolerant Simple Network of Workstations*, r package version 1.3-0.

Sevcikova, H. and Rossini, T. (2012b), *rlecuyer: R Interface to RNG with Multiple Streams*, r package version 0.3-3.

Shaw, J. (2014), "Why Big Data is a Big Deal," Http://harvardmagazine.com/2014/03/why-big-data-is-a-big-deal.

Singh, K., Xie, M., and Strawderman, W. (2005), "Combining Information from Independent Sources through Confidence Distributions," *Annals of Statistics*, 159–183.

Sklyar, O., Murdoch, D., Smith, M., Eddelbuettel, D., and Francois, R. (2013), *inline: Inline C, C++, Fortran Function Calls from R*, r package version 0.3.13.

Snijders, C., Matzat, U., and Reips, U.-D. (2012), "Big Data: Big Gaps of Knowledge in the Field of Internet Science," *International Journal of Internet Science*, 7, 1–5.

Song, Q. and Liang, F. (2014), "A Split-and-merge Bayesian Variable Selection Approach for Ultrahigh Dimensional Regression," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*.

Song, Q., Wu, M., and Liang, F. (2014), "Weak Convergence Rates of Population Versus Single-chain Stochastic Approximation MCMC Algorithms," *Advances in Applied Probability*, 46, 1059–1083.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van Der Linde, A. (2002), "Bayesian Measures of Model Complexity and Fit," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64, 583–639.

Stengel, R. F. (2012), *Optimal control and estimation*, Courier Corporation.

Suchard, M. A., Wang, Q., Chan, C., Frelinger, J., Cron, A., and West, M. (2010), "Understanding GPU Programming for Statistical Computation: Studies in Massively Parallel Massive Mixtures," *Journal of Computational and Graphical Statistics*, 19.

The Apache Software Foundation (2014a), "Apache Avro," Https://avro.apache.org/.

— (2014b), "Apache Hadoop," Http://hadoop.apache.org/.

— (2014c), "Apache Spark," Http://spark.apache.org/.

The MathWorks, Inc. (2014), "Big Data with MATLAB," Http://www.mathworks.com/discovery/big-data-matlab.html.

Tierney, L. (2009), "Code Analysis and Parallelizing Vector Operations in R," *Computational Statistics*, 24, 217–223.

Tierney, L. and Jarjour, R. (2013), *proftools: Profile Output Processing Tools for R*, r package version 0.1-0.

Urbanek, S. (2014), *multicore: A Stub Package to Ease Transition to 'parallel'*, r package version 0.2.

van de Geijn, R. A. (1997), *Using PLAPACK*, Cambridge, MA: The MIT Press.

Venkataraman, S. (2013), *SparkR: R Frontend for Spark*, r package version 0.1.

Visser, M. D. (2014), *aprof: Amdahl's Profiler, Directed Optimization Made Easy.*, r package version 0.2.4.

Vitter, J. S. (2001), "External Memory Algorithms and Data Structures: Dealing with Massive Data," *ACM Computing surveys (CsUR)*, 33, 209–271.

Wang, C., Chen, M.-H., Schifano, E., Wu, J., and Yan, J. (2016), "Statistical Methods and Computing for Big Data," *Statistics and Its Interface*, In press.

Wang, X., Guo, F., Heller, K. A., and Dunson, D. B. (2015), "Parallelizing MCMC with Random Partition Trees," *arXiv preprint arXiv:1506.03164*.

White, T. (2011), *Hadoop: The Definitive Guide*, O'Reilly Media, Inc., 2nd ed.

Wickham, H. (2013), *Tools for Visualisation of Big Data Sets*, r package version 0.1.

— (2014a), "Bin-Summarise-Smooth: A Framework for Visualising Large Data," Http://vita.had.co.nz/papers/bigvis.

— (2014b), *profr: An Alternative Display for Profiling Information*, r package version 0.3.1.

— (2014c), *Visualise Line Profiling Results in R*, r package version 0.1.

Wickham, H., James, D. A., and Falcon, S. (2014), *RSQLite: SQLite Interface for R*, r package version 1.0.0.

Woodroofe, M. (1982), "On Model Selection and the Arc Sine Laws," *The Annals of Statistics*, 1182–1194.

Xie, M., Singh, K., and Strawderman, W. (2011), "Confidence Distributions and a Unifying Framework for Meta-analysis," *Journal of the American Statistical Association*, 106, 320–333.

Yan, J., Aseltine, R. H., and Harel, O. (2013), "Comparing Regression Coefficients Between Nested Linear Models for Clustered Data With Generalized Estimating Equations," *Journal of Educational and Behavioral Statistics*, 38, 172–189.

Yan, J., Cowles, M. K., Wang, S., and Armstrong, M. P. (2007), "Parallelizing MCMC for Bayesian Spatiotemporal Geostatistical Models," *Statistics and Computing*, 17, 323–335.

Yu, B. (2014), "Let Us Own Data Science," IMS Bulletin Online, V. 43 (7).

Yu, H. (2002), "Rmpi: Parallel Statistical Computing in R," *R News*, 2, 10–14.

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. (2010), "Spark: Cluster Computing with Working Sets," in *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing*, vol. 10, p. 10.