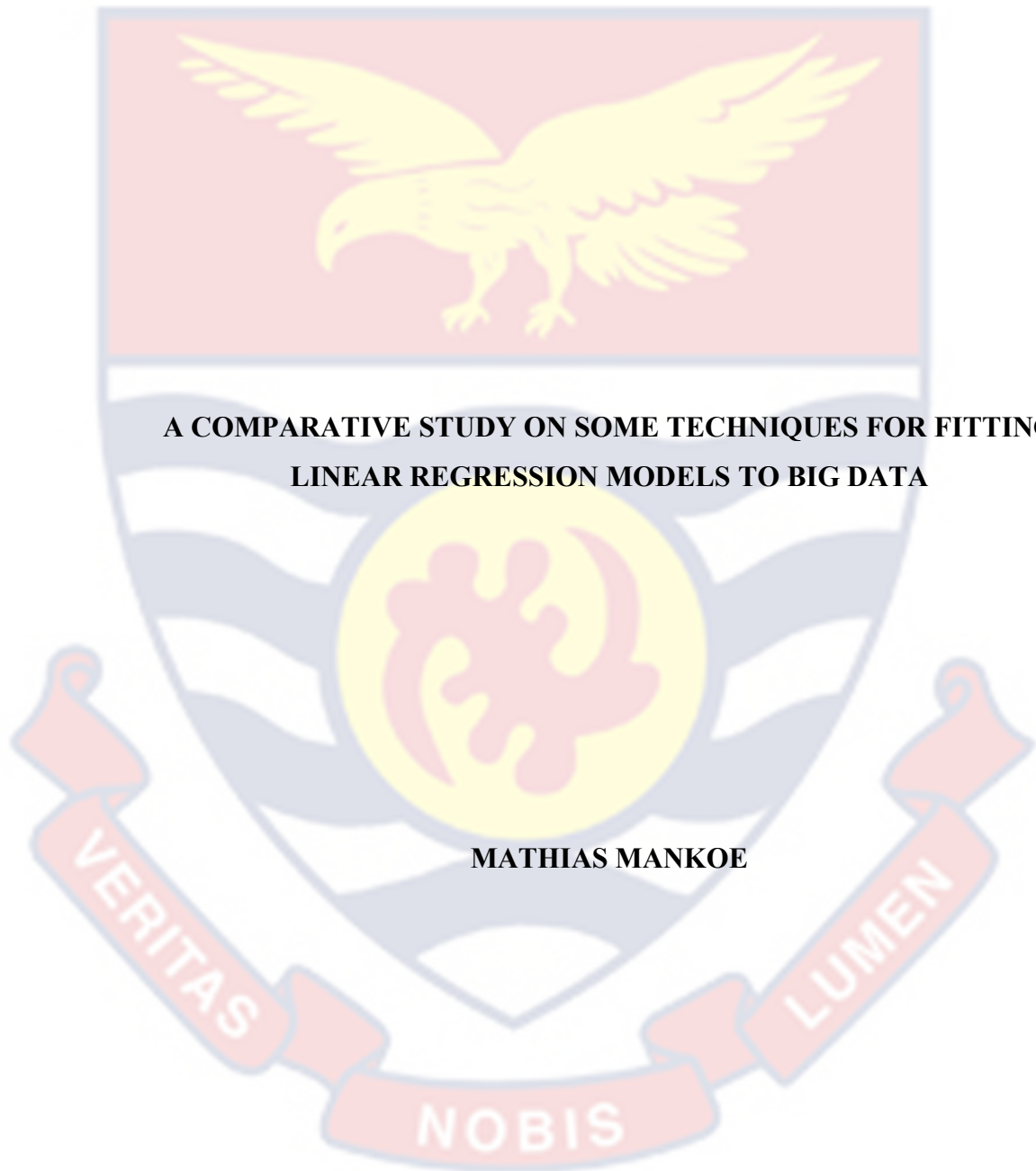


UNIVERSITY OF CAPE COAST



**A COMPARATIVE STUDY ON SOME TECHNIQUES FOR FITTING
LINEAR REGRESSION MODELS TO BIG DATA**

MATHIAS MANKOE

2022



© Mathias Mankoe
University of Cape Coast

UNIVERSITY OF CAPE COAST



**A COMPARATIVE STUDY ON SOME TECHNIQUES FOR FITTING
LINEAR REGRESSION MODELS TO BIG DATA.**

BY

MATHIAS MANKOE

Thesis Submitted to the Department of Statistics of School of Physical Sciences, College of Agriculture and Natural Sciences, the University of Cape Coast in partial fulfilment of the requirements for the award of Master of Philosophy degree in Statistics

OCTOBER, 2022

DECLARATION

Candidate's Declaration

I hereby declare that this thesis is the result of my original research and that no part of it has been presented for another degree at this University or elsewhere.

Candidate's Signature Date

Name: Mathias Mankoe

Supervisors' Declaration

We hereby declare that the preparation and presentation of the thesis were supervised following the guidelines on supervision of thesis laid down by the University of Cape Coast

Principal Supervisor's Signature Date

Name: Dr Francis Eyiah-Bediako

Co-Supervisor's Signature Date

Name: Dr. David Kwamena Mensah

ABSTRACT

This study examines the applicability of two Random Projection and Merge and Reduce methods, widely used in Computer Science, for linear regression analysis of big data in Statistics. The Clarkson-Woodruff, Rademacher Matrix as well as the Merge and Reduce techniques are used as data reduction techniques before performing a linear regression analysis on big data sets. The Classical Merge and Reduce approach uses parameter estimates and standard errors as summary values. In summary statistics, the Bayesian Merge and Reduce approach uses some characteristics of the posterior distribution. The study reveals that the techniques considered in this thesis are good data reduction techniques for fitting linear regression models to big data sets. The Clarkson-Woodruff method provides faster and more reliable reduced data sets for linear regression analysis. The Merge and Reduce models better approximate the true Poisson and linear regression models provided there are enough observations per variable per block (5000 observations per block). However, for data sets with unbalanced factor variables, the Bayesian Merge and Reduce models approximate the true models better than the Classical Merge and Reduce models. The Merge and Reduce models show good approximations of the true models when outliers are evenly distributed among blocks. But the standard errors are overestimated for models without intercept terms. For uneven distribution of outliers, the Random Projection methods provide reliable results. The methods considered in this thesis are largely used in Computers Science, but they can be used for efficient linear regression analysis of big data sets.

KEY WORDS

Big Data

Data Reduction

Data Simulation

Merge and Reduce

Random Projections

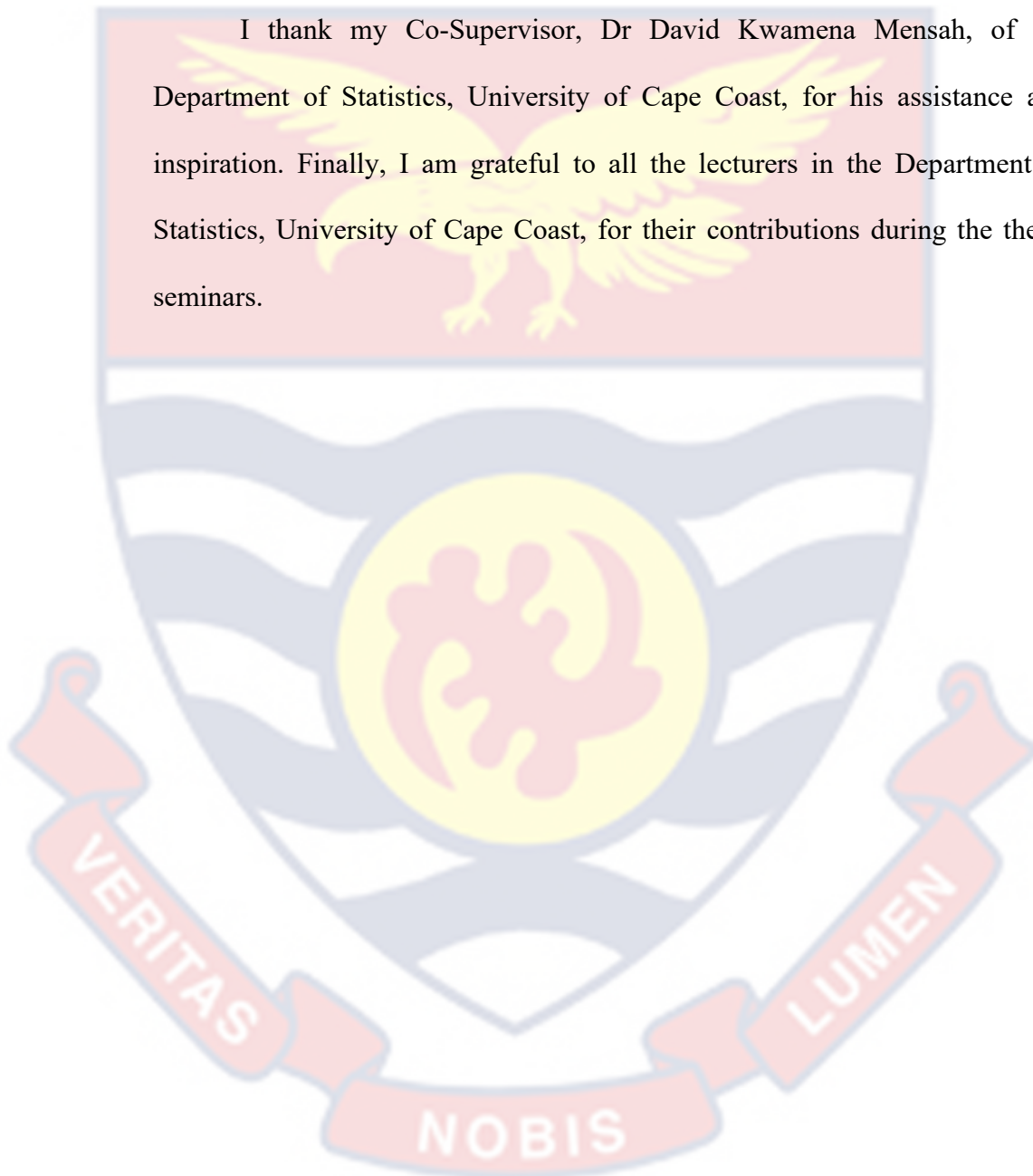
Regression Analysis



ACKNOWLEDGEMENT

I thank my Principal Supervisor, Dr Francis Eyiah-Bediako of the Department of Statistics, University of Cape Coast, for his instructions, guidance and motivation throughout the work.

I thank my Co-Supervisor, Dr David Kwamena Mensah, of the Department of Statistics, University of Cape Coast, for his assistance and inspiration. Finally, I am grateful to all the lecturers in the Department of Statistics, University of Cape Coast, for their contributions during the thesis seminars.



DEDICATION

To My Family



TABLE OF CONTENTS

	Page
DECLARATION	ii
ABSTRACT	iii
KEY WORDS	iv
ACKNOWLEDGEMENT	v
DEDICATION	vi
LIST OF TABLES	xi
LIST OF FIGURES	xii
LIST OF ABBREVIATIONS	xiv
CHAPTER ONE: INTRODUCTION	
Background to the Study	1
Statement of the Problem	8
Purpose of the Study	11
Research Questions	11
Significance of the Study	12
Delimitations of the Study	13
Limitations of the Study	13
Organization of the Study	14
Chapter Summary	14
CHAPTER TWO: LITERATURE REVIEW	
Introduction	16
Random Projections (RP)	17
Bayesian Regression Analysis for Big Data	18
Merge and Reduce (M&R) Method for Big Data	22
Chapter Summary	24

CHAPTER THREE: RESEARCH METHODS

Introduction	25
Linear Regression Analysis	25
Markov Chain Monte Carlo (MCMC) Methods	30
Block-wise Metropolis-Hastings (MH) Method	32
Random Walk Metropolis-Hastings (RWMH) Method	33
Hamiltonian Monte Carlo (HMC) Method	34
Bayesian Linear Regression Analysis	37
Bayesian Inference	38
Parameter Estimation in Bayesian Linear Regression Analysis	39
Random Projections (RP)	41
Epsilon-subspace Projection	42
Some Theoretical Guarantees	42
Rademacher (RAD) Random Projection Technique	43
Clarkson-Woodruff (CW) Random Projection Technique	43
Implementation of Methods	45
The Merge and Reduce (M&R) Principle	46
Merge and Reduce (M&R) Approaches	48
Some Properties of the Merging Technique	50
Generalized Linear Models (GLMs)	51
Application of Methods	52
Data Simulation and Models	52
Software and Packages Used	53
Work Station	54

Chapter Summary	54
CHAPTER FOUR: RESULTS AND DISCUSSION	
Introduction	55
Comparing the Running Times	55
Comparing the Posterior Means	59
Comparing the Fitted Values	61
Comparing the Posterior Distributions	62
Streaming Big Data	63
Analysis of Empirical Big Data Using Random Projection Method	64
Linear Regression Analysis of Simulated Big Data Using the Merge and Reduce Method	68
Results of Linear Regression Analysis for the Classical Merge and Reduce Approach	69
Results of the Linear Regression Analysis for the Bayesian Merge and Reduce Approach	74
Results of Linear Regression Analysis Involving Outliers	78
Results of Linear Regression Analysis Involving Outliers for the Classical Merge and Reduce Approach	79
Results of Poisson Linear Regression Analysis	81
Results of Poisson Linear Regression Analysis for the Classical Merge and Reduce Approach	81
Results of Poisson Linear Regression Analysis for the Bayesian Merge and Reduce Approach	82

Analysis of Empirical Big Data Using the Merge and Reduce Method	84
Chapter Summary	88

CHAPTER FIVE: SUMMARY, CONCLUSIONS AND

RECOMMENDATIONS

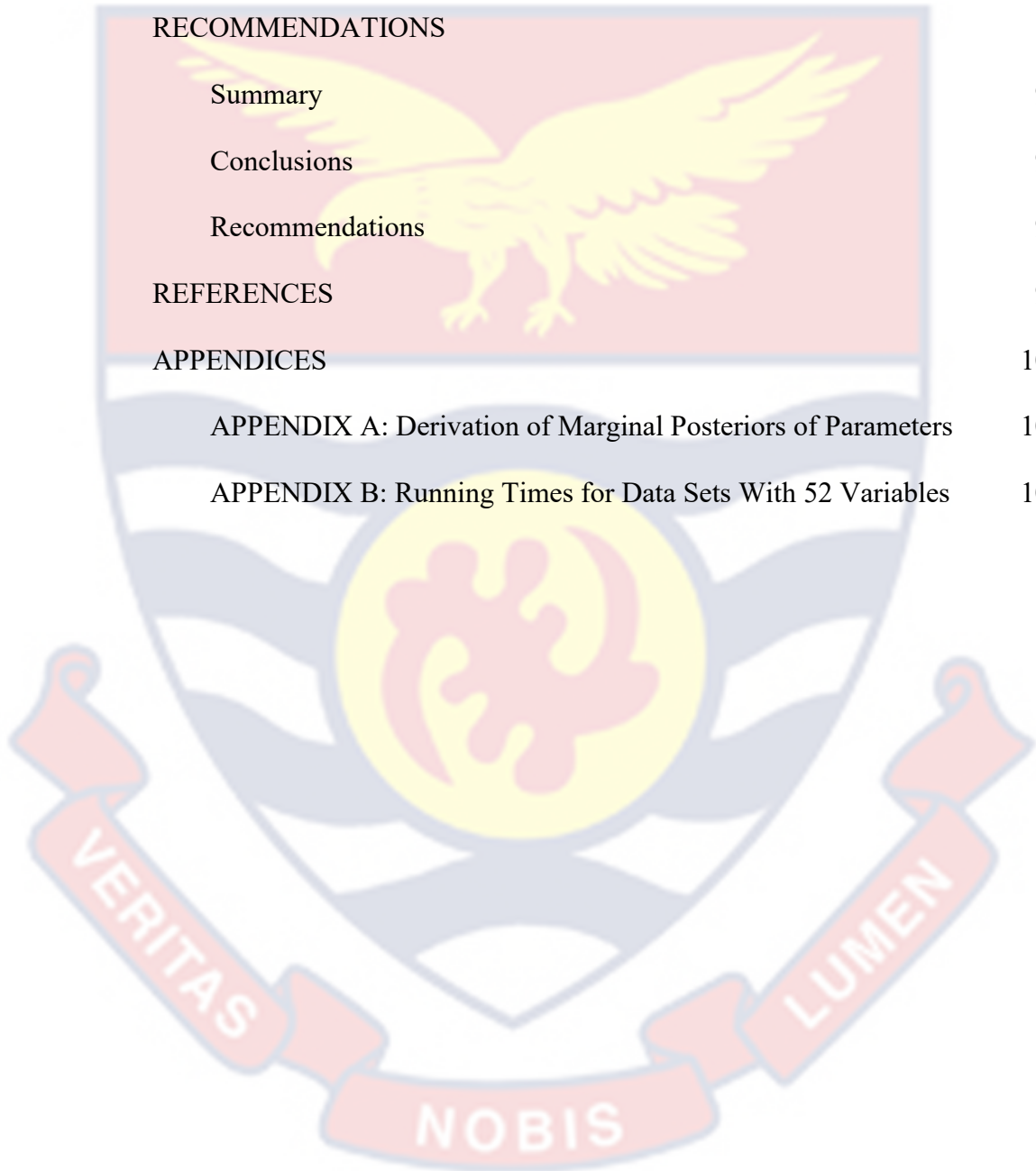
Summary	90
Conclusions	91
Recommendations	93

REFERENCES	94
------------	----

APPENDICES	106
------------	-----

APPENDIX A: Derivation of Marginal Posteriors of Parameters	106
---	-----

APPENDIX B: Running Times for Data Sets With 52 Variables	108
---	-----



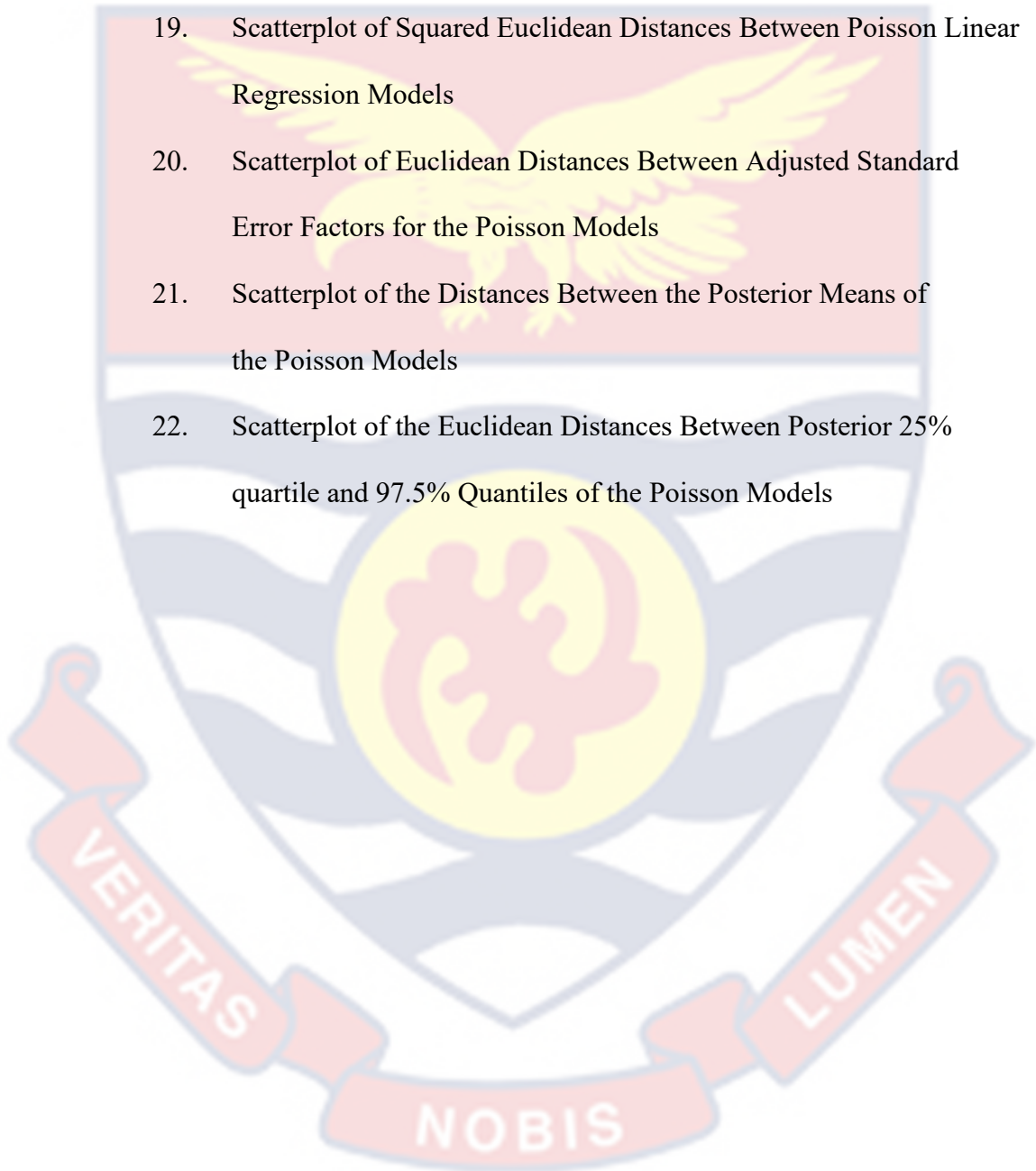
LIST OF TABLES

	Page
1. Comparison of the Two ϵ -subspace Projections	45
2. Overview of Simulated Parameters	52
3. Target Dimensions for the Random Projection Techniques	57
4. Euclidean Distances Between Posterior Means of the Approximated and the Actual Models	59
5. Squared Euclidean Distances Between True Mean Values and Posterior Means of Models Based on the Sketches	60
6. Variables in the Real Data Set	65
7. Sample Sizes of the Sketches	66
8. Sum of Euclidean Distances Between Posterior Means of the Original and Recovered Models	66
9. Quantiles of the Euclidean Distances Between Parameter Estimates	70
10. Results of the Classical Linear Regression Analyses of the Empirical Data	84
11. Results of the Bayesian Linear Regression Analyses on the Empirical Data	85
12. Smallest and Minimal Effective Sample Sizes for Different Block Sizes	87


LIST OF FIGURES

	Page
1. The Principle of the Merge and Reduce Method	46
2. Running Times for Bayesian Linear Regression Models	56
3. Total Running Times in Minutes	58
4. Scatterplot of Fitted Values Using the Original Data Set	61
5. Fitted Values Based on the Original and Approximated Models	62
6. Boxplots of MCMC Samples for β_{11} and β_{22}	63
7. Difference in Fitted Values Based on Original and Reduced Models	67
8. Boxplots of MCMC Samples for the Weather Parameters	68
9. Boxplots of Euclidean Distances Between Simulated Regression Models	69
10. Scatterplot of the Influence of Block Size Per Variable on Euclidean Distances	72
11. Adjusted Standard Error Factors for Simulated Data	73
12. Squared Euclidean Distances Between Posterior Medians	75
13. Relationship Between Block Size Per Variable and Distances Between Posterior Medians	75
14. Boxplots of Adjusted Standard Errors	76
15. Scatterplot of Relationship Between Block Size Per Variable and Distances Between Posterior Lower Quartiles	77
16. Scatterplot of Relationship Block Size Per Variable and Distances Between Posterior 97.5% Quantiles.	78

17. Boxplot of Squared Euclidean Distances Between M&R 1 Models Involving Outliers. 79
18. Boxplots of Euclidean Distances Between Standard Errors of M&R 1 Models Involving Outliers 80
19. Scatterplot of Squared Euclidean Distances Between Poisson Linear Regression Models 82
20. Scatterplot of Euclidean Distances Between Adjusted Standard Error Factors for the Poisson Models 82
21. Scatterplot of the Distances Between the Posterior Means of the Poisson Models 83
22. Scatterplot of the Euclidean Distances Between Posterior 25% quartile and 97.5% Quantiles of the Poisson Models 83



LIST OF ABBREVIATIONS

The background of the page features a large, semi-transparent watermark of the University of Cape Coast crest. The crest is a shield-shaped emblem with a yellow eagle with outstretched wings in the upper half. The lower half is divided into three horizontal bands of blue, white, and blue. In the center of the shield is a yellow circle containing a red and white stylized figure. Below the shield is a red ribbon banner with the Latin motto "VERITAS NOBIS LUMEN" written in white capital letters.

ANN	Artificial Neural Network
CW	Clarkson-Woodruff
GLMs	Generalized Linear Models
GMM	Generalised Method of Moments
HMC	Hamiltonian Monte Carlo
INLA	Integrated Nested Laplace Approximation
JL	Johnson-Lindenstrauss
M&R	Merge and Reduce
MCMC	Markov Chain Monte Carlo
MH	Metropolis-Hastings
MLR	Multiple Linear Regression
NUTS	No-U-Turn sampler
OLS	Ordinary Least Squares
PCA	Principal Component Analysis
RAD	Rademacher
RP	Random Projection
RWMH	Random Walk Metropolis-Hastings
SNA	Social Network Analysis
TSQR	Tall Skinny QR

CHAPTER ONE

INTRODUCTION

Background to the Study

Big data analytics is a study field attracting significant attention from academics and other communities. The volume of data generated in the modern world has increased rapidly. The development of mobile devices, digital sensors, connectivity, computation, and storage has made it possible to gather and store a large amount of data (Yaqoob et al., 2016). Over the past five years, the total amount of data in the world has expanded by more than ninefold and is projected to double biennially (Chen et al., 2014; Naeem et al., 2022). Consequently, the rapid expansion of data has resulted in several problems.

The term big data arose from the demand of corporations such as Google, Facebook and Yahoo to analyze huge amounts of data (Garlasu et al., 2013). There have been different definitions of big data, ranging from 3V to 4V: from the volume, variety, and velocity to volume, velocity, variety, and veracity (Gandomi & Haider, 2015; Hashem et al., 2016; Chen & Zhang, 2014; Rodríguez-Mazahua et al., 2016). Volume describes the quantity of data; velocity relates to the rate of data generation, and variety entails different data types and sources (Chen & Zhang, 2014). Veracity is the fourth feature of big data. Veracity refers to the disorderliness and dependability of data. Value refers to the worth of information concealed within massive data (Chen et al., 2014). Big data refers to huge volumes of data that are difficult for efficient management and processing by current data processing techniques (Chen & Zhang, 2014).

Standard technology used to store and analyze massive amounts of data tend to perform unsatisfactorily (Siddiqa et al., 2016). Massive data could be stored and analyzed using advanced data mining and processing techniques. The rapid pace of data growth, which exceeds the capacity of most current data storage and processing techniques to efficiently process large amounts of data, poses problems for both practitioners and researchers (Begoli & Horey, 2012).

The world's data volume is expected to grow 40% per year and 50 times by 2020 (Yaqoob et al., 2016). About 90% of today's data was generated in the last two years (Naeem et al., 2022). Over the next four years up to 2025, global data creation is projected to grow to more than 180 zettabytes (Naeem et al., 2022). Before the 1990s, data had been growing at around 40 per cent annually. In 1998, it peaked at almost 90 per cent (Odom & Massey, 2003). By the end of 2011, approximately 1.8 zettabytes of data had been created, and it was expected that 2.8 zettabytes would be produced in the subsequent years (Sagiroglu & Sinanc, 2013). Approximately 1.2 zettabytes of electronic data were created annually worldwide (Khan et al., 2014). Sagiroglu and Sinanc (2013) projected business and financial data to exceed 40 zettabytes by 2020. Also, business-to-consumer and online transactions were estimated to exceed 450 billion daily by 2020 (Khan et al., 2014).

Opportunities accompany challenges. The advent of big data has created numerous opportunities for analyzing large data sets. Organizations employ insights from big data to uncover unique hidden behavioural patterns, which tend to be informative for making decisions (Raghupathi & Raghupathi, 2014). Not only does the analysis of big data assist in gaining information on industry trends, but it also facilitates the detection of fraudulent incidents.

For advertising objectives, data analytics assists in the strategic placement of advertisements (Aissi et al., 2002). Moreover, predictive analytics for big data empowers individuals to make informed decisions regarding the knowledge of clients and products. Data analytics helps companies to detect potential threats and opportunities. It assists healthcare organizations in tailoring medication for an individual patient, thus facilitating a quicker and more effective treatment.

The Internet has spawned an explosion of information, including text, audio, photos, and videos. Thousands of transactions frequently occur in online stock trading because of increased human activities and algorithmic high-frequency trading operations. A stock trader may not be able to determine the maximum amount of activity that a particular stock can perform at a given time and circumstance, and thus big data management and analytics are necessary. Big data management and analytics methods that could monitor and provide relevant information to a user are critical in today's world of information.

Another area where big data analytics is of importance is city traffic. Real-time analysis of traffic flow over time and seasons enables planners to alleviate congestion and create alternatives for regular traffic flow. In a broader context, analyzing large data sets from various sources such as mobile phones, GPS devices, and medical equipment could help improve the services offered to the public. Efficient use of these data sets can help retailers enhance their customer experience and manage their various promotions and pricing.

Big data poses some challenges. The main reason why relational database management does not meet the performance requirements of large-scale data is due to its lack of scalability and extensibility (Khare, 2014).

Whereas NoSQL databases have demonstrated some benefits, including scalability, flexibility and cost-effectiveness, they are challenged by large data sets, as they lack maturity and performance reliability. NoSQL databases are not good at handling big data analytics (Khare, 2014). It is imperative to integrate the power of high-performance computing with an efficient technique capable of solving scientific, engineering, and data analysis challenge irrespective of the size of the data. Some high-performance computing techniques enable creativity at any scale. However, the complexity of computational science, as well as engineering codes, presents some obstacles to developing high-performance technologies.

When large data sets are stored in a distributed fashion, it tends to be difficult to retrieve the necessary information promptly. Some novel indexing algorithms and strategies that could expedite the retrieval of essential data are required. Some existing algorithms mainly focus on retrieving data from limited storage sizes; hence, they are incapable of recovering the necessary details promptly in the situation of large data storage (Naeem et al., 2022). Some research studies have addressed this issue, but their efforts appear to be in their infancy (Zhao et al., 2016).

The detection of data patterns could help businesses become more intelligent regarding production and estimations. Nonetheless, given the size, complexity, and dynamism of big data sets, identifying patterns of relevance is difficult. Existing techniques can identify relevant trends, but one of the major issues is that the results are often less accurate (Naeem et al., 2022).

Visualization refers to the representation of information by using graphs. Information abstracted schematically is useful for data processing and offers features for the information units. In most big data applications, executing visualization is tough because of the rapid growth pace and complexity. Existing technologies for big data visualization no longer seem to deliver optimal functionality and fast response time (Wang et al., 2015). Most big data visualization methods have low reaction time, and scalability problems (Naeem et al., 2022). Instead of using old visualization techniques, it is important to reconsider how to visualize large amounts of data.

Corporations are primarily concerned with resolving difficulties in analyzing large data sets while enhancing security. In certain situations where data is generated rapidly, identifying misleading data becomes challenging. Most present data security methods are predicated on a static dataset, though big data sets are constantly changing (Siddiqi et al., 2016; Sookhak et al., 2014). To offer real-time protection, established big data security techniques must embrace the new aspects of big data, including data patterns and variability. Due to the complexity of the data collected and stored in big data, it is hard to implement effective security techniques to protect it without further delay (Naeem et al., 2022).

Many issues about stream computing, cloud computing, parallel computing, grid computing, semantic web computing, granular computing, optical computing and quantum computing and cryptography, as well as edge computing, seem to have not been adequately researched. New technology domains aid in the resolution of big data analytics challenges. Thus, these new

disciplines are insufficiently established to effectively and efficiently manage large data sets.

Big data analytic approaches are required to analyze large data sets in a reasonable time. A handful of methods could be used for big data analysis within a reasonable timeframe. Using data mining techniques, large data sets are summarized into meaningful and useful information. Data mining utilizes statistical and machine learning techniques to retrieve information. Some data mining techniques are regression analysis, cluster analysis, and classification. New strategies for mining massive amounts of data are needed since the data growth rate is accelerating (Chen & Zhang, 2014). Existing techniques for extracting information from large data sets must be modified to employ typical data mining algorithms for big data (Naeem et al., 2022; Zhou & Song, 2017). Hierarchical clustering, k-means and balanced iterative reduction must be enhanced to handle the clustering of big data (Naeem et al., 2022).

Web mining is used to identify patterns from massive web databases (Tracy, 2010). Web mining provides previously undiscovered information about websites and visitors to facilitate data-driven decisions. The method aids in determining the effectiveness of websites. To understand data, web mining visualization techniques generate tables and graphs. However, the complexity of the four Vs makes big data visualization more challenging than conventional small data visualization, even in the web mining setting (Geng et al., 2012). Some data analysts have been using batch mode software to obtain the highest possible parallel data resolution for their Big Data visualization projects (Thompson et al., 2011). Data visualization is thus crucial when dealing with large data sets.

Machine learning enables computers to control their behaviour using empirical data (Chen & Zhang, 2014). Some supervised and unsupervised machine learning algorithms must be scaled up to accommodate large data sets. Map/Reduce is a scalable machine learning framework. Big data machine learning methods are still in their infancy and suffer from scaling issues (Naeem et al., 2022). Utilizing statistical estimations and control theory, artificial neural networks (ANN) are used for adaptive control, pattern recognition and analysis (Liu et al., 2011). Although ANN is frequently employed to satisfy the requirements of large datasets, it is time-intensive (Zhou et al., 2012).

Optimization techniques are applied to tackle quantifiable problems. These procedures are utilized in different disciplines. Different methods, including swarm optimization, genetic algorithms, and quantum and simulated annealing, are applied to solve optimization problems (Sahimi & Hamzhepour, 2010; Li & Yao, 2012; Z. Yang et al., 2008). Optimization methods are particularly effective due to their parallel nature. However, most optimization methods are sophisticated and time-consuming and hence should be scaled up in a real-time context to execute big data operations (Naeem et al., 2022).

Social network analysis (SNA) is a process utilized in the study of social relationships. This type of analysis is commonly used to examine the relationships between various individuals. SNA has grown in importance throughout social and cloud computing. Where the amount of data to be stored is not excessively large, SNA performs well; however, its performance declines when the input data have high dimensionality (Yaqoob et al., 2016).

It is challenging to process high-dimensional data without the appropriate methods. Some methods seek to manage high-dimensional data by reducing its dimension. Linear Discriminant Analysis (LDA), Principal Component Component Analysis (PCA), Multi-dimensional Scaling (MDS), Singular Value Decomposition (SVC), Locally Linear Embedding (LLE), Isometric Mapping (ISOMAP), Laplacian Eigenmap (LE), Independent Component Analysis (ICA), t-distributed Stochastic Neighbour Embedding (t-SNE), and Random Projections are methods that reduce data dimensionality (Anowar et al., 2021).

Statement of the Problem

The term “big data” has become popular in recent years. Available memory and the power of computing have expanded dramatically in recent years. As a result of these developments, it is now possible to store and analyze big data sets using statistical models that are quite complicated (Geppert, 2018). Nevertheless, the time it takes to get results is frequently proportional to the data size (Chen & Zhou, 2020). This indicates that regression techniques frequently do not scale effectively, making analysis of big data sets difficult. Although storing data sets normally may not cause any problem with available memory, it could be imperative to put it into working memory, with far less storage space (Geppert, 2018).

The emergence of big data and available computer processing power is not new, but it presents a problem to statistical techniques and their scalability. Large data sets present a computational and methodological problem when performing regression analysis. Memory requirements scale at least linearly with sample size and the number of variables in the big data (Geppert et al.,

2020). This is a challenge in both Classical and Bayesian statistical analyses, as normal regression analysis approaches require access to the entire set of data. That is, standard statistical methods often have a linear dependence on the sample size and the number of variables in the data set as well. This seems less of an issue in the classical context, as the outcome is obvious after a single run over the data. However, in the Bayesian framework, the majority of the Markov Chain Monte Carlo (MCMC) techniques take several runs through the data, which implies that significant time is usually required to obtain the results. In both situations, linear regression analysis could be impractical for high-dimensional big data, although the challenge is more obvious in a Bayesian context.

To address these issues, some scholars have proposed two requirements for techniques to be viable for Bayesian linear regression analysis on high-dimensional large data sets. First, they assert that every update only reads a part of the data set whose size is independent of the original high-dimensional data set (Welling et al., 2014). This condition allows for the feasibility of techniques for streaming data that could be read as containing indefinitely huge amounts of data, ensuring that the procedure works irrespective of the size of the high-dimensional input data set. Secondly, as per Welling et al. (2014), the technique should be used in parallel settings to avoid potential issues. The second requirement seems more advantageous because large data sets can be redistributed among multiple computers for processing, thereby reducing the workload on each computing system as well as conforming to the design of existing computer systems. Moreover, the second requirement would enable the

linear regression analysis on a parallel system with relatively little computing power.

There are also methodological issues associated with regression analysis of big data to consider. Concerning the concept of statistical significance, large sample size in addition to other variables leads to larger values of test statistics. Although this is a good property in a broad sense, it could reject the hypothesis of interest for extremely tiny discrepancies between the estimated and tested values. It might also result in statistically significant findings for insignificant outcomes in practice.

Scalability is among the primary issues of contemporary data analysis due to the increasingly large amount of information. For some statistical approaches, massively high-dimensional data results in substantial resource utilization (Geppert, 2018). In both Classical and Bayesian linear regression analysis, conducting linear regression analysis on big data sets with data points far greater than the number of variables ($n \gg p$) becomes progressively time-consuming and memory intensive, making the analysis almost impractical. This is particularly critical when big data cannot be stored in the quick internal memory and must be retrieved repeatedly from slow external memory databases, which significantly increases the real elapsed time (Geppert et al., 2020).

Thus, big data poses a problem concerning the memory and the running time required for Classical and MCMC techniques for linear regression analysis, respectively. The problem entails the need for a large computing memory mainly for Classical linear regression analysis and increasing running time mostly for Bayesian linear regression analysis. It is

against the issues of memory and running time that this study attempts to explore the applicability of the principles of Random Projection (RP) and Merge and Reduce (M&R) from Computer Science for linear regression analysis of big data set in Statistics. This thesis focuses on analyzing large data sets using both Bayesian and classical linear regression techniques, in which the number of observations is considerably greater than that of the variables.

Purpose of the Study

The main purpose of this thesis is to examine the applicability and appropriateness of some big data techniques widely used in Computer Science, namely, Random Projection and Merge and Reduce for linear regression models within Statistics. The thesis is tailored towards the following specific objectives.

1. To evaluate the performance of the Rademacher Matrix (RAD) and the Clarkson-Woodruff (CW) random projection techniques on big data sets when applied to linear regression models.
2. To assess the performance of the Merge and Reduce approach when applied to Generalized linear models, particularly Poisson regression models.
3. To examine the performance of the Merge and Reduce approach when applied to linear regression models in the presence of influential observations.

Research Questions

To achieve the objectives of the study, the following research questions would serve as a guide:

1. Which of the two Random Projection techniques: the Rademacher (RAD) and the Clarkson-Woodruff (CW), better approximate linear regression models when applied to big data?
2. Is there any performance difference between the Classical and Bayesian Merge and Reduce methods when applied to linear and Poisson regression analysis of big data?
3. What is the performance of the Merge and Reduce method in approximating the linear regression model when outliers are present in a big dataset?

Significance of the Study

Large data sets present analytical challenges on both computational and methodological levels. Analyzing big data sets needs a significant amount of working computing memory in the classical framework and other modelling frameworks. In the Bayesian framework, in using the Markov Chain Monte Carlo methods, a substantial amount of time is required for developed algorithms to converge to achieve posterior inference for large data sets.

This study presents efficient and less memory-intensive techniques for performing linear regression analysis on big datasets by utilizing Random Projections and Merge and Reduce procedures. This thesis focuses on exploring the applicability and appropriateness of some well-known big data techniques widely used in Computer Sciences for solving linear regression problems in Statistics. As a result, this thesis has the potential to motivate the consideration of some well-known big data techniques within the statistical community. Additionally, this thesis can direct research into other techniques for handling big data sets in different fields within Statistics.

Delimitations of the Study

This study concentrates on conditions where a big data set contains a greater number of observations, with a smaller number of variables. The goal is to lower the sample size of the large data set and efficiently fit a linear regression model to the new data set in recovering the original linear model. The techniques examined in this thesis are for data sets of $n \gg p$. Thus, the methods examined in this thesis are aimed at recovering linear regression models, not the data points upon which the models are built.

Limitations of the Study

Since the study focuses on the case where the sample size (n) is much greater than the number of variables (p), $n \gg p$, the methods examined cannot be applied to the case of $n \ll p$. For the M&R technique, the number of variables in a block is often problematic because each block has its own set of data. This means that the data collected from each block will only be a subset of the total. The subsets are disjointed, causing some challenges as models must be integrated with disjointed parameter sets. Based on the subset of variables, linear models may not adequately fit the data.

Model diagnostics constitute another limitation of this study. While both techniques use observations to approximate the required linear regression model, the data points are not immediately available thereafter. The M&R methods read the observations block-by-block and eliminate them once the model for each block is developed. This requires re-using the data points, which could be impractical. When utilizing the random projection approach, the sketched data are retained; however, they are represented by random linear combinations of the original data. Every projected data point encompasses

many actual data points with varying weights. Generally, sketched data are less useful for model diagnostics.

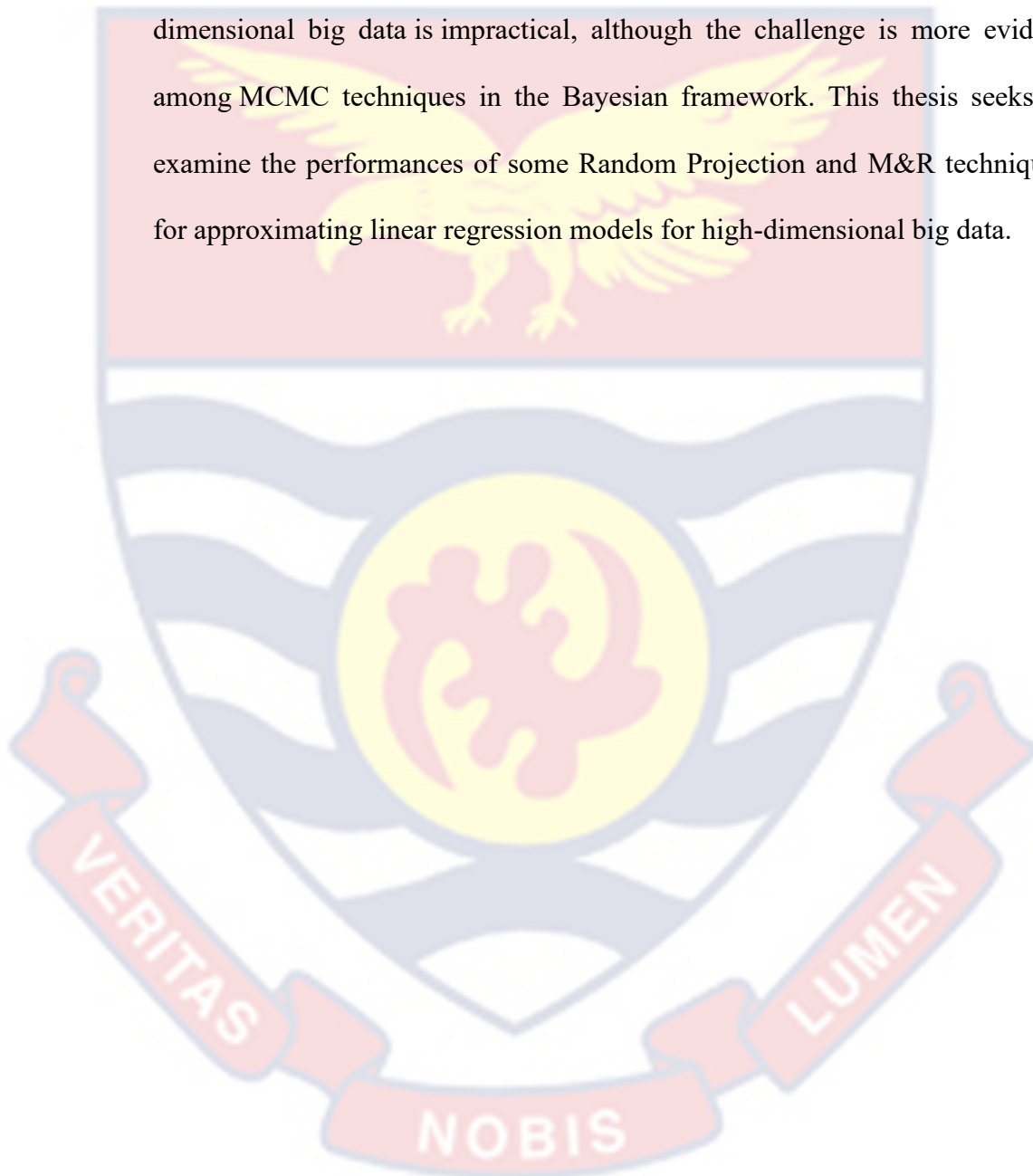
Organization of the Study

This thesis is divided into five chapters. The first chapter presents an introduction to the study. It comprises the study background, the statement of the research problem, the objectives of the study, the research questions and the outline of the thesis. Chapter Two is dedicated to reviewing relevant literature and some theoretical guarantees of the techniques employed in the study. Chapter Three critically discusses the methodologies and techniques employed in the study. The chapter also describes how the simulation study was performed to examine the performances of the Random Projection and the M&R techniques for estimating linear regression models for big data. Chapter Four outlines and discusses the data analysis findings. Chapter Five of the thesis summarises all key findings and highlights them in light of the study's objectives, from which conclusions are drawn.

Chapter Summary

High-dimensional big data sets pose computational and methodological challenges when performing linear regression analysis. Computing memory requirements scale at least proportionally with sample size and the number of variables in a given data set. This provides a challenge in both classical and Bayesian (MCMC) linear regression techniques, as linear regression analysis approaches require access to the entire data set. Likewise, the running time of conventional MCMC techniques is often proportional to the dimension of the data set. This is less of an issue in the classical context, as the outcome is obvious after a single run over the data.

Nevertheless, in a Bayesian framework, many MCMC techniques take several passes through the data, implying that significant time is required to obtain results. In both circumstances, linear regression analysis for high-dimensional big data is impractical, although the challenge is more evident among MCMC techniques in the Bayesian framework. This thesis seeks to examine the performances of some Random Projection and M&R techniques for approximating linear regression models for high-dimensional big data.



CHAPTER TWO

LITERATURE REVIEW

Introduction

This chapter presents a critical review of some literature on linear regression techniques for high-dimensional big data, dimensionality reduction techniques for big data in Classical and Bayesian linear regression analyses, as well as the efficiency of some MCMC techniques, particularly for the Bayesian case.

Reducing the dimension of big data is a fundamental concept widely applied in Statistics and Computer Science. Generally, the objective of dimension reduction is to reduce the number of variables to a reasonably manageable number. Principal Component Analysis is popularly used in Statistics with the primary goal of eliminating multicollinearity issues by substituting orthogonal bases for the original variables. More particularly, PCA is a data reduction technique useful in analyzing multivariate data. The principal component of a multivariate data set is a diminishing proportion of its total variability. With PCA, the dimensionality of multivariate data could be reduced while maintaining the structure of the original data set by using the first principal components.

The purpose of this study includes reducing the number of data points from n to k , where the sample size is much greater than the number of variables, $n \gg p$. This is particularly beneficial when dealing with high-dimensional big data sets since the running time of most statistical procedures is proportional to the sample size n . In this case, approaches such as partial least squares and principal component regression are appropriate (Feldman et al., 2020).

Recent advances on coresets for k-means clustering are based on Principal Component Analysis. A study by Feldman et al. (2020) sought to identify a small subset of high-dimensional data that recovers the original data set with an approximation error of $0 < \epsilon < 0.5$. The goal of Feldman et al. (2020) was to combine the random sampling of multiple observations with the exclusion of identical ones from previously sampled data.

Using the principle of data squashing on large data sets, Geppert (2018) sought to reduce data depending on the likelihood of the observations. The study attempted to retain the statistical information in the original high-dimensional data. The statistical analyses performed on the squashed data set produce findings that are reasonably close to those obtained from the original high-dimensional data. Geppert et al. (2020) partitioned the data set by utilizing clustering-based likelihood. Pseudo-observations of the clusters are then constructed and used in subsequent statistical analysis.

Random Projections (RP)

Several studies have utilized Random Projection techniques in Computer Science to obtain a low-rank approximation for least squares regression, Gaussian process regression, clustering problems and classification tasks, as well as for compressed sensing (Banerjee et al., 2013; Cohen et al., 2015; Kerber & Raghvendra, 2014; Paul et al., 2014). The random Projection procedure has been used to approximate a subspace group using sparse vectors only (Baraniuk et al., 2008).

The statistical elements of Random Projections, as well as some techniques based on randomized linear algebra, have been researched by many scholars to study leverage score-based subsampling techniques and their

statistical characteristics (Ma et al., 2014; Raskutti & Mahoney, 2015). Dimension reduction and subsampling techniques have been used to efficiently prepare the data set before effectively estimating the least squares estimator (J. Yang et al., 2016). The researchers examined parallel and distributed procedures and substantiated their findings with comprehensive empirical assessments of high-dimensional large data sets. This thesis extends the investigation of the statistical aspect of Random Projections by applying it to Classical and Bayesian MLR models.

Bayesian Regression Analysis for Big Data

Bayesian linear regression analysis for big data has attracted the attention of many researchers. In 2015, Guhaniyogi and Dunson recommended using Random Projections as a data reduction tool for a larger number of variables and a smaller number of observations (Guhaniyogi & Dunson, 2015). They demonstrate that the approximations converge to the expected posterior distribution. However, Boutsidis et al. (2014) had already claimed that it is not practical in general, as dimension reduction performed in the absence of the target variable could result in additive errors in the worst scenario.

In contrast, some scholars have used Tall Skinny QR (TSQR) in situations where the number of observations is much larger than the number of variables in the high-dimensional big data set (Benson et al., 2013; Demmel et al., 2012). TSQR makes many runs through the data. Thus, TSQR can be used to precondition big data for Bayesian inference utilizing MCMC techniques. According to Geppert (2018), the TSQR results in a steady decomposition with a high degree of precision. However, he notes that the TSQR is limited to regression models with Gaussian distributed likelihood and takes row-by-row

data. As per Demmel et al. (2012), the running time associated with TSQR is bounded low due to the QR-decomposition.

Without loss of generality, when no mathematically solvable conjugate model is obtainable, MCMC techniques are the gold standard for Bayesian analysis. MCMC techniques provide insight into potential posterior distribution approximation difficulties. The amount of time it takes to run the analysis depends on its size and might be fairly long, resulting in inefficiency (Balakrishnan & Madigan, 2006).

As Geppert et al. (2020) and Geppert (2018) notes, there is a fascinating field of research seeking to remedy this problem, and one such area of research is the enhancement of MCMC techniques, such as the Hamiltonian Monte Carlo (HMC) approach. The benefit of simply determining whether the outcome is a reasonable approximation of the expected posterior distribution is retained. Balakrishnan and Madigan (2006) suggest other options like tweaking the MCMC algorithm. A major reason for this limitation is that the likelihood is evaluated repeatedly, which often requires all the observations.

Balakrishnan and Madigan (2006) developed a method for reading high-dimensional data block by block and performing a series of MCMC steps on each block. Each successive block retains certain data points while discarding others based on respective weights. Although the approach developed by Balakrishnan and Madigan (2006) makes only one pass over the data, it presents a complete MCMC sample. Also, although the selection rule used is experimentally justifiable, only the univariate scenario has theoretical validity, which is predicated on the central limit theorem for Sequential Monte Carlo techniques.

Some research studies have proposed subsampling the high-dimensional data to estimate the acceptance or rejection region at each stage of the Metropolis-Hastings (MH) technique (Bardenet et al., 2014). They demonstrate that the estimated decision is highly close to the original decision in each iteration. According to Bardenet et al. (2014), the proposed subsampling algorithm's iteration count is not fixed in advance. Still, the number of iterations that a stopping criterion takes is determined by the variability of its likelihood ratios.

Others have also proposed that instead of the MH or MCMC algorithms, a variant ensures the proposed point is accepted or rejected depending on the subsample of the data set (Quiroz et al., 2018). This minimizes the algorithm's processing cost, resulting in a more efficient method for finding the posterior distribution (Quiroz et al., 2018). Moreover, Quiroz et al. (2018) offered proportionate inclusion probability values to the contributions of the data points to the likelihood estimated by a Gaussian Process. Nonetheless, the proposed scheme for MH adds a delayed acceptance method that evaluates the likelihood only when a significant chance of acceptance exists (Quiroz et al., 2018a). While both scenarios may bear considerable resemblance to the aforementioned data-squashing, Quiroz et al. (2018b) sought to provide standard MCMC techniques. Although there seem to be no theoretical guarantees of approximations, Quiroz et al. (2018a, b) demonstrated useful empirical applications.

Additionally, some alternatives replace the MCMC algorithm. The Integrated Nested Laplace Approximation (INLA) is a good alternative to MCMC techniques (Rue et al., 2009). Others have stated that the Laplace

approximation is a more accurate method of estimating posterior distributions than Monte Carlo (Rue et al., 2009). They emphasize that the Laplace approximation technique is only applicable to latent Gaussian models. However, the group of latent Gaussian models contains many models with which the INLA is considerably faster than the Monte Carlo Markov Chains (Martins et al., 2013). Rue et al. (2009), however, posit that it could be more challenging to check the posterior distribution approximation accuracy.

There are instances where it may not be possible to compute the likelihood function, and in such a situation, Approximate Bayesian Computation provides a way around the Bayesian analysis (Csilléry et al., 2010). Alternatively, Csilléry et al. (2010) suggest that simulations could be used to approximate the likelihood. They reveal that the Approximate Bayesian Computation is not entirely comparable to the others because its primary goal is not to perform more efficient analyses but to perform Bayesian analyses in situations where it might be difficult otherwise. As a general rule, Csilléry et al. (2010) state the Approximate Bayesian Computation can produce summary statistics for small dimensions, where the number of variables is less than 10.

Some research studies have also examined the divide and conquer strategy for some statistical challenges associated with big data, with an emphasis on different regressions models, hypothesis testing, partially linear models, and estimating equation frameworks with differentiable estimating functions (Chen & Xie, 2014; Lee et al., 2017; Li et al., 2012; Zhang et al., 2015; Zhao et al., 2016). Zhang et al. (2015) compressed each subsample into statistical measures and uniquely combined the respective statistics to generate the final model estimates employing one of the most frequent techniques for the

divide and conquer strategy. The estimates approximated the model with a close efficiency to estimates from the whole data set. A computationally and memory-efficient technique for the estimating equation method with differentiable estimating functions has also been designed (Lin & Xi, 2011). Nevertheless, although quantile regression's parameters could be estimated using the estimating equation approach, this technique is difficult to use.

The Rao-CD and Wald-CD methods have been used to perform quantile regression on big datasets (Zhou & Song, 2017). However, Rao-estimation CD's process requires minimization of a quadratic form provided as a generalized method of moments (GMM) estimator, which is problematic for non-smooth estimating functions. Additionally, when analyzing a data stream, the Rao-CD approach needs re-running the minimization step when new data is received, which is extremely time-demanding. Zhou and Song (2017) did not demonstrate how to compute the weight matrix for the Wald-CD technique. It is critical to recognize that the estimation technique may be somewhat difficult due to the unknown density function.

Merge and Reduce (M&R) Method for Big Data

The M&R is a basic strategy for handling static data systems to accommodate dynamic entries. The technique has been used to process coresets in a data streaming setting (Agarwal et al., 2004; Har-Peled & Mazumdar, 2004). According to Geppert et al. (2020), the M&R concept is mostly used to develop effective and fast streaming and parallel techniques for the processing of high-dimensional big data. Although the M&R is implicitly used, it is regarded as a standard method in coresets. Coresets have been explored widely as a data reduction and aggregation technique for addressing scalability

concerns associated with a variety of challenges in Computational Statistics. Coresets have been applied to address issues with shape fitting (Feldman et al., 2020). Several studies have employed coresets for clustering, classification, 2-regression, 1-regression, p -regression, M-estimators and generalized linear models (GLMs) (Baykal et al., 2018; Huggins et al., 2016; Munteanu et al., 2018; Tolochinsky & Feldman, 2018). An interesting study applied the M&R concept to the relational database physical design of big data (Bruno & Chaudhuri, 2007).

In the Bayesian framework, a one-pass-through-the-data Merge and Reduce streaming technique could be used to facilitate the conduct of Bayesian linear regression involving big data. Balakrishnan and Madigan (2006) applied the M&R principle where the data was read blockwise and run through an MCMC sampler in several stages. The data was read block by block; the technique either retained or replaced some samples. The selection criteria were weighted to reflect the data's significance. A pass-through-the-data-once streaming algorithm for Bayesian linear regression has been developed using Random Projections-based dimension reduction techniques (Geppert et al., 2017). They employed a linear map data sketching structure, which made it simple to add and modify data interactively.

A study by Law and Wilkinson examined composable Bayesian models for streaming high-dimensional data analysis (Law & Wilkinson, 2018). They addressed the issue of sample frequency imbalance in practical streaming and parallel situations and hence had a distinct scope.

Only a few research studies have applied the Rademacher and Clarkson-Woodruff random projection techniques directly to linear regression analysis of big data, despite its power, simplicity and low error rate compared to the other dimension reduction methods. Also, since the M&R streaming technique is popularly used in data structures in Computer Science, it has not received any serious attention in Statistics, particularly in solving MLR analysis of big data in a streaming setting. In the extant literature, hierarchical modelling, constraints such as data storage and accumulation, data streaming and parallel computations exist. Therefore, some statistical models tractable in M&R could be applied, in principle, to linear regression analysis of big data.

Chapter Summary

The literature has revealed how big data pose computational and methodological challenges in data analysis. It is established that big data consumes significant time when performing Bayesian regression analysis on them, particularly when MCMC methods are employed. In classical regression analysis, the literature reveals that big data poses a memory challenge. Records further indicate that data will continue to grow. There are studies on how to perform regression analysis on big data in both Bayesian and classical settings. Standard statistical techniques have been used extensively to study the effect of high-dimensional big data on linear regression models. Some results show that the impacts of big data on linear regression models depend on the type of data. However, big data has an important and consistent influence on the predictive performance of linear regression models.

CHAPTER THREE

RESEARCH METHODS

Introduction

This chapter presents the theories of the primary methods that are important to the subject under study. The methods reviewed in this chapter include regression models, Random Projection methods and M&R techniques for approximating regression models for big data sets.

Linear Regression Analysis

Regression analysis is a well-established statistical concept that models and estimates the effects of p predictors x_1, x_2, \dots, x_p , on typically one response variable, Y . The dependent variable Y depends on the values of independent variables x_1, x_2, \dots, x_p . The latter is usually represented as column vectors in a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. The row vectors of \mathbf{X} are commonly denoted by $\mathbf{X}_i \in \mathbb{R}^p$, where $i = 1, 2, \dots, n$, representing the i th data point. A linear model is the most fundamental method of connecting the two variables. The matrix \mathbf{X} and the vector \mathbf{Y} are connected by a vector, $\boldsymbol{\beta}$, of weights and an additive error term $\boldsymbol{\eta}$, which is unobservable, culminating in a model of the form;

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim N(0, \sigma^2 I_n) \quad (1)$$

Some Assumptions of Linear Regression Analysis

Some assumptions underline linear regression analysis. First, when the response is normally distributed, linear regression is typically used. Multiple linear regression analysis assumes that the residuals, which are the predicted minus observed values, are also normally distributed. Although most tests are relatively robust against violation of the normality assumption, it is usually

prudent to check the probability distributions of important variables in the study before reaching definitive conclusions. Generally, a normal probability plot is used to examine the probability distribution of the residuals.

When at least two independent variables in a linear regression model are correlated, it is said that multicollinearity exists. Without loss of generality, multicollinearity is defined as a condition in which a significantly high degree of correlation lies between predictor variables, such that their effects cannot be differentiated. It is expected, in a linear regression analysis, that the explanatory variables are not significantly correlated.

Another important assumption in multiple linear regression analysis is the assumption of homoscedasticity. Homoscedasticity refers to a condition where the response variable has a similar proportion of the variation as the predictor variables over a given range of values. Generally, homoscedasticity or the assumption of constant variance is checked by plotting the residuals against the fitted values. The form of the spread about the zero line indicates whether or not the assumption of constant variance is met.

Estimation of Parameters in Classical Linear Regression Analysis

The design matrix, \mathbf{X} , may contain data variables as well as other elements like transformed and standardized variables, interactions among variables, and an intercept term. In classical and Bayesian frameworks, GLMs can be used. In a classical context, the goal is to estimate every element of $\boldsymbol{\beta}$, often regarded as unknown constant values. The estimation of the $\boldsymbol{\beta}$ is done in such a way that the error function is minimized. The ordinary least squares (OLS) estimator, which is the optimal solution, $\hat{\boldsymbol{\beta}}$, of equation 1, is estimated using the following;

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \quad (2)$$

In standard texts, the OLS estimate is an unbiased linear estimator which has $E(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$ and $\text{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$. The variance-covariance matrix of the OLS estimator is used in the estimation of the standard errors given as follows;

$$se_{\hat{\beta}_j} = \sqrt{\sigma^2 \mathbf{e}_j' (\mathbf{X}'\mathbf{X})^{-1} \mathbf{e}_j} \quad (3)$$

where \mathbf{e}_j denotes the j^{th} unit vector, $j = 1, 2, \dots, p$ and $se_{\hat{\beta}_j}$ is the standard deviation.

Linear Regression Model Diagnostics

Following a linear regression analysis, model diagnostics are performed to assess the goodness of fit of the model. The residuals are used to assess the regression model's goodness of fit. Residuals signal some significant effects that are not captured by the model and show when certain linear regression assumptions have been violated. Since the error term, η , cannot be observed, the residual, $\mathbf{r} = \mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}$, is considered as an estimated error term that serves as a substitute for the non-observed error term, the raw residuals, r , represents the difference between the observed and the estimated values of \mathbf{Y} . The OLS estimator minimizes the sum of squared residuals when compared to all choices of $\hat{\boldsymbol{\beta}}$ (McCullagh & Nelder, 2019).

Although examining the raw residuals in regression analysis is critical and useful, it is important to highlight the difference between the raw residuals and the error term. Unlike raw residuals, the vector of unknown errors, η is presumed to be uncorrelated and homoscedastic. That is $\mathbf{r} \sim (\mathbf{0}, \sigma^2(\mathbf{I}_n - \mathbf{H}))$, $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ is known as the hat matrix. The variances, $\text{Var}(r_i)$, are dependent on the value of i , which is not optimal for diagnostics. Given the

foregoing, two additional forms of residuals are presented. The standardized residuals (r) are defined as follows;

$$\tilde{r}_i = \frac{r_i}{\hat{\sigma}\sqrt{(1 - h_{ii})}} \quad (4)$$

h_{ii} is the i^{th} diagonal element of the hat matrix, \mathbf{H} . $i = 1, 2, \dots, n$. When the estimated $\hat{\sigma}$ is substituted for the true unknown σ , it is obvious that $E(\tilde{r}_i) = 0$ and $\text{Var}(\tilde{r}_i) = 1$, $i = 1, 2, \dots, n$. The studentized residuals and the standardized residuals are interrelated. With the studentized residuals, rather than estimating the variance, σ^2 , the variance for each observation is estimated by taking into account the difference between the observed and the model. The studentized residuals (r_i^*) are defined as

$$r_i^* = \frac{r_i}{\hat{\sigma}_{-i}^2 \sqrt{(1 - h_{ii})}} \quad (5)$$

where $\hat{\sigma}_{-i}^2$ is the estimated variance without the i^{th} observation. $i = 1, 2, \dots, n$.

The standardized and studentized residuals are analyzed using a residual plot containing the fitted values and the standardized residuals. If the linear model fits the data well with the underlying assumptions met, no clear pattern will be seen in the plot. That means that the residuals have constant variance and are placed around the zero line. However, If most of the low residuals, $r_i < 0$, are clustered in a small area of the domain, the variability increases as \hat{Y} increases, indicating that the linear model has some problems. The studentized residuals could be utilized in addition to standardized residuals.

The connection between standardized and studentized is as follows;

$$r_i^* = \tilde{r}_i \sqrt{\frac{n-p-1}{n-p-r_i^2}} \quad (6)$$

This indicates that whenever the residuals lie in the interval $[-1, 1]$, the standardized and studentized residuals seem to be approximately equal. Where the residuals have a greater absolute value, the variance increases as studentized residuals have higher absolute values. Accordingly, when the studentized residual is used, exceptionally high or low residuals become more apparent. Additionally, the studentized residual is used to create statistical tests, like the r_i^* have a t_{n-p-1} distribution (Chatterjee & Hadi, 1988). Apart from residual diagnostics, which analyze the difference between the observed and estimated values as per the linear model. The extent to which the observations affect the parameter estimates and their variability is examined.

In simple linear regression, $Y = \beta_0 \cdot \mathbf{1}_n + \beta_1 X_1 + \eta$, variations in the values of y_i have a greater influence on the linear model when the data points are closer to the high values of y and a lower effect on the linear model when the data points are in the centre of the range (Geppert, 2018).

Euclidean Distance

The Euclidean distance may give a more precise definition of open sets. If p is a point of \mathbb{R}^3 and $\varepsilon > 0$ is a number, the ε neighbourhood N_ε of p in \mathbb{R}^3 is the set of all points q of \mathbb{R}^3 such that $d(p, q) < \varepsilon$. Then a subset S of \mathbb{R}^3 is open, provided that each point of S has an ε neighbourhood entirely contained in S . In short, all points near enough to the point of an open set are also in the

set. This definition is valid with \mathbb{R}^3 replaced by \mathbb{R}^n . If p and q are points of \mathbb{R}^3 .

The Euclidean distance from p to q is the number;

$$d(p, q) = \left(\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2} \right) \quad (7)$$

A Euclidean distance matrix is a representation of a set of points in a given space that is spaced by square distances. Given a set of points $p_1, p_2, p_3, \dots, p_n$ in k -dimensional space, \mathbb{R}^k , the elements of its distance matrix P are given by the squares of distances between them. That is, $P = (p_{ij})$;

$$p_{ij} = d_{ij}^2 = \|p_i - p_j\|^2 \quad (8)$$

Where $\|\cdot\|$ denote the Euclidean norm on \mathbb{R}^k .

$$P = \begin{bmatrix} 0 & d_{12}^2 & d_{13}^2 & \dots & d_{1n}^2 \\ d_{21}^2 & 0 & d_{23}^2 & \dots & d_{2n}^2 \\ d_{31}^2 & d_{32}^2 & 0 & \dots & d_{3n}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1}^2 & d_{n2}^2 & d_{n3}^2 & \dots & 0 \end{bmatrix}$$

In the case of distance matrixes, the entries are usually defined as distances, not squares. In the case of the Euclidean scheme, squares are used to simplify the various algorithms and theorems related to distance.

Markov Chain Monte Carlo (MCMC) Methods

MCMC methods are exact for posterior inference when the posterior distribution is intractable (Givens & Hoeting, 2013). For situations where a Bayesian linear regression model cannot be analytically fitted, an approximation is often used. For such approximations, MCMC methods are regarded as the gold standard since they tend to be more effective and applicable to several model types. It is also possible to test for convergence. This allows

us to determine if the approximation is good and if more iterations are required. A downside of MCMC approaches is that they are inefficient since the needed running time before convergence could be large. In the following lines, a quick overview of common MCMC algorithms is provided in the following paragraphs. This thesis employs the No-U-turn technique.

As a numerical integration approach, Monte Carlo integration draws observations at random and evaluates them based on a function of interest. Monte Carlo integration techniques include rejection sampling. The techniques used in this process are designed to analyze a candidate's value from a proposal distribution and determine whether or not it should be accepted or rejected. Accepted observations are then added to the sample to represent the posterior distribution. Where a candidate value is rejected, the sample remains intact. In both circumstances, the procedure is repeated until the required sample size is attained.

In addition to rejection sampling, there are alternative Monte Carlo techniques that sample observations directly from the posterior distribution. Nevertheless, the majority of such approaches are ineffective, especially when the prior distribution is uninformative relative to the probability (Bolstad, 2009). As a result, rejection sampling has been superseded in many cases by more effective methods, some of which are described below.

The use of MCMC techniques is an efficient technique for sampling the posterior distribution. But MCMC algorithms could be quite slow based on the model and data size. The MCMC techniques begin with initial values for the parameters and draw a new candidate value from a proposed distribution at each iteration. According to the Markov property, the new observation can

solely depend on the current observation only. The goal of the algorithm is to find a way to match the new value with the posterior distribution of the current observation, allowing the candidate value to be retained or discarded.

The MH method provides one option for constructing an appropriate Markov chain. Given θ_1 , the initial observation of the Markov chain and $\theta_t, t = 2, \dots, T$, additional Markov chain observations, where T is the given chain length. The elements θ_t of the Markov chain are iteratively drawn. The MH technique uses knowledge of the unnormalized posterior distribution. Only the current state and the new candidate value of the Markov Chain are considered. The proposed distribution, q , does not affect the stationary distribution, but it can affect the acceptance rate as well as convergence speed (Givens & Hoeting, 2013).

According to Geppert (2018), the desirable acceptance rates for a given proposal range from around 0.3 to 0.7. Higher acceptance rates may indicate that the proposal's variation is less than that of the posterior distribution. This could lead to an extended exploration of the high-probability mass regions. A low acceptance rate suggests that the proposal's variation is greater than the posterior distribution. This is because the sample includes multiple observations. This method lowers the effective size of the sample.

Block-wise Metropolis-Hastings (MH) Method

The block-wise MH algorithm is suitable for the multi-parameter problem. A candidate vector is drawn and then rejected or accepted using the MH algorithm. The block-wise variant partitions the parameters into J blocks. If variables are found to be associated, it is possible to place them in one block, although not required. When the values of the parameters are partitioned, they

are replaced with a single value, θ_c , which is then used to sample the candidate values for each block, $\theta_{cj} = 1, \dots, J$. The proposal distribution for a given block is dependent on the current values of the various elements of the data chain. For instance, the value of the block j parameter in block 1 is important.

The candidate values for entry t of block j are computed iteratively. The candidate values for entry t of block j are then dependent on the t^{th} value for blocks $J^* < J$ and $(t - 1)^{th}$ value for blocks $J' > J$.

The Gibbs sampling algorithm is a special case of block-wise MH. It constructs a Gibbs sampler with the proposal distribution chosen as the full conditional distribution in block j . The probability of a candidate's value being accepted is 1 in every proposal distribution. This is due to the strong correlation between the proposal's posterior distribution and the probability of the candidate's value being accepted. Unfortunately, this is not the case with Gibbs sampling.

Random Walk Metropolis-Hastings (RWMH) Method

The proposal distribution of the block-wise or basic MH algorithm can be chosen symmetrically depending on the number of parameters involved, such that the proposal distribution $q(\theta_{t-1} | \theta_c) = q(\theta_c | \theta_{t-1})$. In such cases, the acceptance probability is given by,

$$\alpha = \min \left(1, \frac{p(\theta_c | X, Y)}{p(\theta_{t-1} | X, Y)} \right) \quad (9)$$

where;

θ_c is a candidate sampled from the proposal distribution, θ_{t-1} refers to the next elements of the Markov Chain, $t = 2, 3, \dots, T$. T is the length of the Markov Chain.

With the Rake-Hundred-Mandel-Hinds (RWMH) algorithm, the next candidate doesn't have to take into account the current position of the chain. If the candidate's posterior distribution is better than the current value, then it is accepted, but if it is further away from the centre, it is rejected. On the other hand, if the candidate's probability mass is lower than the centre, it can still be accepted. The proposed distribution of the MH algorithm should always move in the direction of the high probability mass to ensure the exploration of the posterior distribution. The variance of q is also important in the characterization of the distribution. For instance, if the variance of q is greater than the average step size, the algorithm's efficiency will be affected.

Hamiltonian Monte Carlo (HMC) Method

In most MCMC algorithms, the initial value, θ_1 is drawn to represent the probability close to a region of high probability mass. However, given a higher dimensional situation, θ_1 is likely to be placed elsewhere. The algorithm's implementation ensures that the constant step sizes are maintained. In most cases, the initial value of the algorithm is not ideal in certain situations, such as when the target region has a high probability of mass. However, in these cases, the adaptive step sizes provide a remedy.

The HMC algorithm is a Monte Carlo method that provides a solution to the increasing step sizes in a system. It is a hybrid approach that combines MCMC techniques and Hamiltonian dynamics. Neal (2011) compared the two components to a hockey stick that moves over ice. Every component of the

algorithm has its momentum variable, φ_j . The goal-making mechanism of a hockey stick relies on its position, θ and momentum, φ_j . When the plane section moves up a slope, the momentum will remain constant, while the position will be slowed down. This makes the HMC algorithm ideal for determining the overall direction of the movement.

The HMC algorithm updates the value of the momentum using a new distribution, $N(0, I_p)$. Next, the momentum and position are updated. This method is performed by drawing a p -dimensional unity matrix. The HMC algorithm uses the L steps of the leapfrog method to perform Hamiltonian dynamics simulations. It updates the dynamics continuously using a series of steps. The first two updates are followed by an update of the position. The last step is a new value of momentum. The HMC algorithm updates the momentum's value by an addition to the current value, an l -fraction of its product. Regarding half-updates, the algorithm takes into account the product of the gradient, Δ_θ , and $\ln p(\theta|Y, X)$ of the current parameter value. Following L of the step size, l , u is sampled, $u \sim \text{Unif}(0,1)$ and accept the new observation if

$$u < \alpha = \min \left(1, \frac{\exp \left(\ln p(\theta^*|Y, X) - \frac{1}{2} \varphi^* \varphi^* \right)}{\exp \left(\ln p(\theta_{t-1}|Y, X) - \frac{1}{2} \varphi_{t-1} \varphi_{t-1} \right)} \right) \quad (10)$$

When the candidates are rejected, $\theta_t = \theta_{t-1}$ and $\varphi_t = \varphi_{t-1}$. When the candidates are accepted, $\theta_t = \theta^*$ and $\varphi_t = -\varphi^* \cdot \varphi^*$ is negated. It is not possible to have a symmetrical proposal distribution. It is “time-reversible,” but since the HMC algorithm can preserve volumes, it is considered a valid proposal.

The HMC approach requires careful attention to two key parameters: the step size, l , and number, L . Doing so will allow us to perform better simulations and minimize the effects of Hamiltonian dynamics (Hoffman & Gelman, 2014). If too many steps are performed in a sequence, the algorithm may not be able to follow the steps of the previous ones properly. To avoid this, Gelman and Hoffman (2014) proposed a no-u-turn sampler (NUTS) that can be used without tuning any parameters. This method achieves results similar to an HMC algorithm. In 2014, Gelman and Hoffman proposed a method for optimizing step size, l , with vanishing adaptation. They used a doubling algorithm to tune L automatically. The NUTS algorithm takes into account the candidate's various values and generates a set of suitable values

The first step in the process of adding candidates involves adding the current position and momentum. In the second step, two candidates are added, and in the third step, a total of three candidates are added. After a j th step, the number of candidates is 2^j . Every single step of the process involves a 2^{j-1} candidate moving backwards or forward from the outermost candidate. The process halts when the new observations are re-traced, which suggests that the candidate may already be explored. It also suggests that the candidate could be characterized by a lower probability of happening.

The program shows the latest position as a random sampling of one candidate from the set. To maintain the current state of the program, NUTS suggests that one of the two new candidates should be considered a candidate for a new position in step j . This method is associated with longer jumps; it has a low likelihood of finding a new one in the later stages. Although small jumps

are not ideal for every situation, they allow long jumps even though they are not always feasible.

Bayesian Linear Regression Analysis

The parameter β is considered a random variable in the Bayesian framework. It is modelled using a probability density. The assumed distribution is called the prior distribution, $p(\beta)$, which means that the knowledge about it before its existence is related to the current knowledge. The knowledge about the prior distribution is taken into account when choosing $p(\beta)$. This is shown in equation 11.

$$p(\beta, \sigma^2 | X, Y) \propto L(Y | \beta, \sigma^2) p(\beta) p(\sigma^2) \quad (11)$$

where,

$p(\beta)$ – Prior distribution for β

$p(\sigma^2)$ – Prior distribution for σ^2

$L(Y | \beta, \sigma^2) \sim N(X\beta, \sigma^2 I_n)$ – Likelihood

The posterior distribution is used to infer the linear relationship existing between X and Y . The posterior distribution is a function that takes into account the information that was previously known. It is computed as a function of the prior distribution and the likelihood. The distribution in equation 11 is a representation of the product of the probability and prior distribution.

The posterior can be obtained as a closed-form expression in a variety of ways, such as through a simple Bayesian analysis. Conjugate models are well-known for their use in estimating the likelihood and prior distributions. Conjugate models are commonly used for analysis, but they are not available for every model. This is because the normalization constant is not an ideal solution, and it can prevent an analytical solution from being obtained. Usually,

methods such as MCMC or Laplace approximation are used to solve these problems.

In line with a critical review of computational methods for approximating the posterior distribution by Bolstad (2010), this thesis employs the HMC technique. After successfully obtaining a sample of this type, various methods are used to analyze it. The methods include the mean, median, variance, and quantiles. A good boxplot or kernel density estimate can provide a good visual representation of the data. In a regression context, the significance of a variable can be evaluated using statistics. For instance, if a variable is important, a significance test can be performed.

Bayesian Inference

Bayesian inference is about the posterior distribution of the parameter of interest. It combines data information termed the likelihood function with the prior probability model assumed for the parameter. Let Y follow the probability model $P(Y|\theta)$, where θ is a parameter. Treating θ as random, let the uncertainty associated with θ be modelled using $P(\theta)$. Then, the posterior inference about θ can be made using the posterior distribution.

$$P(\theta|Y) = \frac{P(Y|\theta)P(\theta)}{P(Y)} \quad (12)$$

$$P(Y) = \int P(Y|\theta)P(\theta)d\theta \quad (13)$$

From equation (12), the following statement can be made;

$$P(\theta|Y) \propto P(Y|\theta)P(\theta)$$

and

$$P(\theta|Y) \propto \frac{1}{P(Y)}$$

It can be observed that the integral in equation (13) can introduce intractability in the posterior for complex models in which there are many parameters. This shows that some alternative methods should be considered for inference in cases where $P(Y)$ cannot be obtained in closed form. That is, the posterior distribution will not be available in closed form. An alternative method that avoids the use of the marginal likelihood $P(Y)$ is the MCMC method. MCMC methods are based on the full conditionals of the parameters instead of the posterior distributions.

Parameter Estimation in Bayesian Linear Regression Analysis

For the linear model in Equation (11) to make posterior inference about the parameters of interest, the joint posterior distribution must be obtained first. Given the following models for β and σ^2 .

$$P(\beta) = N(\mu_\beta, \Sigma_\beta) \text{ and}$$

$$P(\sigma^2) = IG(\alpha, \gamma)$$

$IG(.)$ denotes an inverse gamma distribution with parameters α and γ . The joint posterior for β and σ^2 are derived as follows.

$$P(\beta, \sigma^2 | Y) = \frac{P(Y | \beta, \sigma^2) P(\beta) P(\sigma^2)}{P(Y)} \quad (14)$$

$$P(Y) = \int P(Y | \beta, \sigma^2) P(\beta) P(\sigma^2) d\beta d\sigma^2$$

Using the numerator of equation (14),

$$P(\beta, \sigma^2 | Y) \propto P(Y | \beta, \sigma^2) P(\beta) P(\sigma^2)$$

$$P(\beta, \sigma^2 | Y) \propto N(y; X\beta, \sigma^2 I_n) \times N(\beta; \mu, \Sigma_\beta) \times IG(\sigma^2; \alpha, \gamma)$$

$$\begin{aligned}
P(\beta, \sigma^2 | Y) &\propto (2\pi\sigma^2)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma^2}[(Y-X\beta)'(Y-\beta)]} \\
&\times (2\pi)^{-\frac{r}{2}} |\Sigma_\beta|^{-\frac{1}{2}} e^{-\frac{1}{2}[(\beta-\mu_\beta)'\Sigma_\beta^{-1}(\beta-\mu_\beta)]} \\
&\times \frac{\gamma^\alpha}{\Gamma(\alpha)} (\sigma^2)^{-\alpha-1} e^{-\frac{\gamma}{\sigma^2}}
\end{aligned} \tag{15}$$

Then, in Equation (15), the marginal posteriors for β and σ^2 are obtained respectively. It can be shown that the marginals are of the form.

$$P(\beta | Y) = \int P(\beta, \sigma^2 | Y) d\sigma^2 \tag{16}$$

$$= N(\mu_\beta^*, \Sigma_\beta^*) \tag{17}$$

where

$$\Sigma_\beta^* = \frac{\sigma^2 \Sigma_\beta}{\Sigma_\beta X'X + \sigma^2 I_r}$$

$$\mu_\beta^* = \frac{Y' \Sigma_\beta + \sigma^2 \mu_\beta}{\Sigma_\beta X'X + \sigma^2 I_r}$$

r is the dimension of β .

$$P(\sigma^2 | Y) = \int P(\beta, \sigma^2 | Y) d\beta \tag{18}$$

$$= IG(a^*, b^*) \tag{19}$$

where,

$$a^* = \left(\frac{n}{2} + \alpha\right), \quad b^* = \frac{1}{2} [(Y - X\beta)'(Y - X\beta)] + \gamma$$

Point estimates for β and σ^2 can be obtained from the corresponding marginal posteriors. Usually, the posterior means are used. For β , the posterior estimate will be the μ_β^* and for σ^2 . The corresponding estimate will be the posterior mean of the inverse gamma given by

$$E[\sigma^2 | Y] = \frac{b^*}{a^* - 1}, a^* > 1 \tag{20}$$

The details of the derivations are provided in Appendix A.

Random Projections (RP)

Random projections are mathematical techniques that show how a given vector's dimension can be reduced by a random matrix. For instance, if a vector has a dimension of n , $\mathbf{v} \in \mathbb{R}^n$, then $\exists \boldsymbol{\pi} \in \mathbb{R}^{k \times n}$ for which the following inequality is true:

$$(1 - \varepsilon)\|\mathbf{v}\| \leq \|\boldsymbol{\pi}\mathbf{v}\| \leq (1 + \varepsilon)\|\mathbf{v}\| \quad (21)$$

The epsilon parameter, which is an estimate of the projection distance, is used to determine the close relationship between the original vector and the projection. This concept was first introduced by the Johnson-Lindenstrauss theorem in 1984 (William & Lindenstrauss, 1984). There has been a lot of research in this area, and one of the main goals is to find a way to create random matrixes suitable for different applications.

To reduce the size of the data, random projections are used. In the case of $n \gg p$, the goal is to reduce the n to k , where $n > k > p$ and p is the number of variables. However, this method is not ideal for every case and is prone to error. For instance, if a reduction below the rank of \mathbf{X} is performed, it can lead to a catastrophic loss. The term random projections are used to describe the process of sketching. In this thesis, the Johnson-Lindenstrauss (JL) theorem is applied in the random projections. In a Classical case, the loss function for $\boldsymbol{\beta}$ could be regarded as a vector.

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^n (y_i - X_i \boldsymbol{\beta})^2 = \arg \min_{\boldsymbol{\beta}} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad (22)$$

This allows the values of the function to be recovered when multiple options are considered. A property that is desirable for various kinds of random

projections is that the computation of the goodness-of-approximation can be controlled by the parameter ϵ .

Epsilon-subspace Projection

Consider a matrix $U \in \mathbb{R}^{n \times p}$ with orthonormal columns, some integer, $k \leq n$. Given an approximation parameter, $0 < \epsilon \leq 0.5$, the epsilon-subspace projection for U is a function $\pi: \mathbb{R}^n \rightarrow \mathbb{R}^k$ such that $(1 - \epsilon)\|U_X\|^2 \leq \|\pi U_X\|^2 \leq (1 + \epsilon)\|U_X\|^2, \forall U_X \in \mathbb{R}^p$. If $p(\beta|X, Y)$ is tractable, then the epsilon-based posterior $p(\beta|\pi X, \pi Y)$ will also be tractable. First, the big data set is projected onto a lower-dimensional subspace and then an appropriate statistical analysis is performed on the reduced data set. This helps to reduce the running time problem associated with MCMC techniques as well as the memory problem in the Classical framework in big data analytics.

Some Theoretical Guarantees

This thesis presents two methods that are commonly used for obtaining random projections in linear regression. The first is the Rademacher matrix (RAD), while the second is the Clarkson-Woodruff (CW) approach. The resulting sketches are linear. Instead of sketching the entire data set at a time, one can easily create subsets of the data set and then aggregate them into a single output similar to the original. This is especially useful when performing a quick analysis of several data sets in parallel.

Most RP fails to provide the expected good-of-approximation. This occurs when the probability of failure (δ) increases. This is because the target number of observations (k) is influenced by the failure probability. The good-of-approximation is not held by the random projection method. It is merely a statistical approximation that can be slightly worse than the expected result.

Although catastrophic failures can happen, such as the failure of a computer, the likelihood of a meteor hitting the device is still higher if it hits the embedded data set. This study compares two different classical linear regression models and sketches the original data set. If the results of the analysis (β) are not similar, a failure might occur.

Rademacher (RAD) Random Projection Technique

The Rademacher matrix is a simple form of sketching that takes into account the probability of every single cell sampled from $\{-1, 1\}$. Then, the matrix is multiplied by $\frac{1}{\sqrt{k}}$ to rescale it, which gives the sketching matrix, π .

When choosing $k = \mathbf{O}\left(\frac{p \log(p/\delta)}{\epsilon^2}\right)$, the failure probability is δ , and the error of projection is ϵ (Sarlos, 2006). This technique was first used to decrease the number of observations needed to get the lower bound. However, Clarkson and Woodruff (2009) proposed $k = \mathbf{O}\left(\frac{p + \log(1/\delta)}{\epsilon^2}\right)$.

Clarkson-Woodruff (CW) Random Projection Technique

For the CW method, the matrix is scaled by multiplying it with $\frac{1}{\sqrt{k}}$ to get $\pi = \frac{1}{\sqrt{k}} \varphi D$. $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix. With equal probability, its elements are drawn from $\{-1, 1\}$. $\varphi \in \mathbb{R}^{k \times n}$ is also a matrix of 0s and 1s. A random map, h , is used for the positions of the 1-entries. That is $h: \{1, \dots, n\} \rightarrow \{1, \dots, k\}$. For each $i \in \{1, \dots, n\}$. The image, $h(i)$, is drawn from $\{1, \dots, k\}$ with equal chance, $\frac{1}{k}$. The elements, $\varphi_{h(i), i} = 1, \dots, n$, are turned to 1. Any other member of φ is turned to 0.

The CW sketch could be used to apply any matrix in $\mathbf{O}(np)$ time. This is ideal for analyzing and sketching complex data sets. It also provides a small multiplicative factor when sketching data sets (Geppert, 2018). The target dimension of the sketch is $k = \Omega(p^2)$. Nevertheless, with a probability of $1 - \delta$ and $k = \mathbf{O}\left(\frac{p^2}{\varepsilon^2\delta}\right)$, a good ε -subspace embedding is guaranteed (Nelson & Nguyen, 2013).

The two main methods used to sketch the data set were RAD and CW. The former could be faster than the latter, as it takes into account the number of entries in the set. With the latter on the other hand, the number of variables affects the size of a sketch. This is also why the number of variables in a given sketch affects the overall size. Although CW could generally be faster than RAD when it comes to creating sketches, it can also be affected by the size of the target dimension.

The selection of a sketching method is influenced by the size of the data and desired time to obtain the sketch. This is relevant for frequentist linear regression. For instance, if the data set is large, the MCMC algorithm's running time can be affected by the size of the data set. If a quick analysis is a focus, then a small target dimension is required. These are ideal for applications in distributed systems. For instance, a k dimension is not dependent on n , and can be easily used in a distributed environment.

Although the sketches can handle certain types of cases where $k > n$ is an important difference, they can also be affected by situations where the data set is being distributed in parallel or batches. For instance, in a streaming situation, the data set may be larger than the sketch since k is independent of n . The results of both methods are controlled by the approximation error

parameter. For instance, if the ϵ values are larger than the original results, then the smaller the data set is the larger error. On the other hand, if the values of ϵ are lower than the original results, then the larger the data set is. Table 1 provides a comparison of the two random projection techniques.

Table 1: Comparison of the Two ϵ -subspace Projections

Technique	Target Dimension	Running Time	Handles $k > n$
RAD	$\mathcal{O}\left(\frac{p+\ln(1/\delta)}{\epsilon^2}\right)$	$\mathcal{O}(npk)$	Yes
<i>CW</i>	$\mathcal{O}\left(\frac{p^2}{\epsilon^2\delta}\right)$	$\mathcal{O}(\text{nnz}(X)) = \mathcal{O}(np)$	Yes

Source: Geppert (2018)

As shown in Table 1, the target dimension k , or the ϵ -subspace embedding function, depends on the number of 0s in \mathbf{X} and the time it takes to get it. The failure probability is expressed in terms of the non-zero entries in \mathbf{X} .

Implementation of Methods

The R package *RaProR* was used. The *RaProR* makes it easy to set the target dimension, k , of a random projection without having to specify a projection error value, ϵ . This is a significant trade-off in the Bayesian case. The lower the target value, the larger the approximation error. If the goal is to get a result as quickly as possible, a larger value of the projection error can lead to a higher reduction in the accuracy of the approximation, while a lower value of the projection error can provide a better approximation.

The Merge and Reduce (M&R) Principle

The M&R approach consists of the following steps that are carried out repeatedly:

1. Batching data using the appropriate procedure
2. Reading in blocks of data
3. Conducting statistical analysis of the current block of data
4. Sufficient statistics that summarize the analysis are stored
5. Merging models following a tree structure while ensuring their complexity does not increase.

Figure 1 illustrates how the principle of Merge and Reduce was applied.

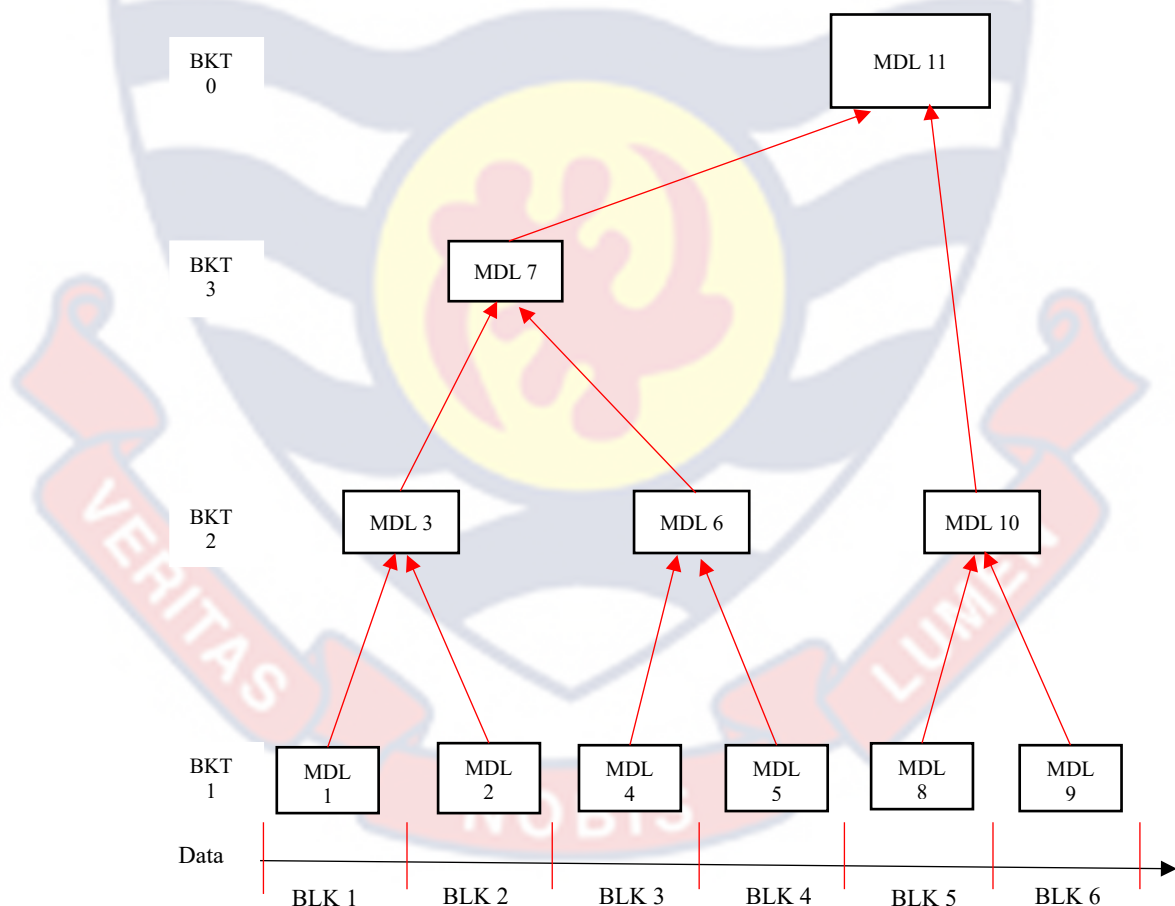


Figure 1: The Principle of the Merge and Reduce Method

Before implementing the Merge and Reduce technique, it is important to specify the number of observations that will be made per block, n_b . In streaming setups, the total number of observations and blocks may not be known beforehand. However, since memory requirements are also taken into account, $n \leq n_b B$. Every model in our implementation summarizes the results of statistical analysis. It also contains some meta-information: the model level and the sample size it takes to make a model work.

The data is read block-wise until it reaches the desired number of observations. A model is then built on that block of data, and once the model has been created, the data is deleted. This happens whenever large sample sizes are recorded in streaming contexts. Models are then merged with another model after the relevant information has been passed to the next level. This ensures that the space for storing observations is always maintained. On the other hand, to keep the models, the memory required is only computed based on the number of blocks to be analyzed, $\mathcal{O}\left(\ln\left(\frac{n}{n_b}\right)\right)$. Two models with the same level are merged when two of them become available. For instance, if M_2 is stored, then M_1 and M_2 will immediately merge to form M_3 . However, since they are on different levels, M_4 is not merged with M_3 . When M_5 is calculated, then M_6 becomes M_7 . The process of merging two different models is not considered problematic if the appropriate weight is used. After the two models are merged, the complexity of the new model is reduced to prevent it from increasing.

Merge and Reduce (M&R) Approaches

In this section, the Classical and Bayesian Merge and Reduce techniques are discussed. The merge step involves merging the models, while the reduce step ensures that the complexity of the model does not increase.

Classical Merge and Reduce Approach (M&R 1)

In the Classical Merge and Reduce approach, the parameter estimates and standard errors were used as summary statistics. A Classical regression model is typically characterized by the use of parameter estimates and standard errors. In most cases, the standard error could be used for both the linear and generalized models. The estimate of the change in the j th variable is very important for predicting the effect of the variable. The standard error is used to determine whether the variable impact on a given set of values is distinguishable from zero. This helps to retrieve important details about the original model. Our first step in the Merge and Reduce approach 1 is to take the weighted mean of the various summary values. Given the vectors, S_{i-1} and S_i , of summaries for models $i - 1$ and i respectively, the merged vector, $S_{m\&r}$ is thus obtained using Equation (18).

$$S_{m\&r} = w_{i-1}S_{i-1} + w_i S_i \quad (23)$$

The weights, w_{i-1} and w_i , are given by $w_{i-1} = n_{b,i-1}/(n_{b,i} + n_{b,i-1})$ and $w_i = n_{b,i}/(n_{b,i-1} + n_{b,i})$ with $n_{b,i-1}$ and $n_{b,i}$ as the number of observations upon which the two models are built respectively. The merge step ensures that the model does not get too complex. The reduce step automatically adds value to the final model after merging. The merge step ensures that the model becomes overly complex, and the reduce step automatically adds value to the final model after the merge.

Bayesian Merge and Reduce Approach (M&R 2)

The Bayesian Merge and Reduce models used some characteristics of the posterior distribution as summary statistics. The M&R 2 was used to estimate the posterior distribution of Bayesian linear regression models. The statistics were chosen carefully so that they provide an accurate representation of the distribution. Several useful and reasonable solutions can be found to the problem of estimating the posterior distribution of a given model. In this study, the mean, median, 2.5% and 97.5% quantiles, as well as the MCMC sample are used as summary values.

$$S = (\bar{x}_1, \dots, \bar{x}_d, \tilde{x}_{u,1}, \dots, \tilde{x}_{u,d}, s_1, \dots, s_d) \quad (24)$$

where $u \in \{0.025, 0.25, 0.5, 0.75, 0.975\}$ denotes the posterior quantiles considered.

The number of observations that the models make is used as the weights for the final model. In the case of the frequentist case, every observation is taken into account to ensure that the final model is equal to its importance. In the Bayesian situation, some correction factors are needed. The standard deviation is used for the posterior distribution. The same procedure is used for the standard error. The posterior standard deviation is then divided by $\sqrt{\left[\frac{n}{n_b}\right]}$. Thus, standardization is required, as shown in Equation (21).

$$S_{u,j}^{corrected} = \frac{S_{u,j} - \bar{x}_j}{\sqrt{\left[\frac{n}{n_b}\right]}} + \bar{x}_j \quad (25)$$

For quantiles considered to be measures of dispersion, the posterior mean is subtracted from the equation. The correct quantile is then computed by introducing the correction factor.

Some Properties of the Merging Technique

In this section, some properties of the merging approach are reviewed. Given a non-stochastic matrix, \mathbf{X} with rank p , the values of \mathbf{X} are realizations from some arbitrary random distribution with expected value $-\infty < \mu_X < \infty$ and variance $\sigma_X^2 > 0$. For linear regression models, as the standard least squares estimator $\hat{\boldsymbol{\beta}}$ is unbiased for every block $b = 1, 2, \dots, B$. That is $E(\hat{\boldsymbol{\beta}}^b) = \boldsymbol{\beta}$. This thesis provided an unbiased estimate of the true values of various parameters $\boldsymbol{\beta}$ using a merging approach.

The variance-covariance matrix of the estimator is computed by taking into account the different values of \mathbf{X} . The principal diagonal matrix shows the variances for every estimate, $\beta_j, j = 1, 2, \dots, p$. The j th component has a variance, $\text{Var}(\beta_j) = \sigma^2 (\mathbf{X}'_j \mathbf{X}_j)^{-1}$. Although σ^2 is an unbiased estimator, its block-wise counterpart, $\text{Var}(\beta_j^b)$, may not be as accurate as its counterpart, $\text{Var}(\beta_j)$. This is because the difference between the two is due to the difference in the respective version of $(\mathbf{X}'_j \mathbf{X}_j)$. The \mathbf{X} is non-stochastic, and its entries are derived from the distribution, (μ_X, σ_X^2) . The n and $\mathbf{X}'\mathbf{X}$ are positively related. On the other hand, the elements of the secondary diagonal are slower than those of the primary diagonal. Hence, the primary diagonal of $(\mathbf{X}'\mathbf{X})^{-1}$ and n are negatively related.

The estimation of the variance of a block is often biased if the previous summary statistics are used to merge it. For most blocks, the estimated variance is around $\frac{n_b}{n}$. However, for the last block, the expected variance is higher due to the presence of fewer observations. This block then has a lower weight, which

makes the variance overestimated by $\left[\frac{n_b}{n}\right]$. To avoid overestimation, the summary statistics that represent standard deviations should be divided by the overestimated factor, as shown in Equation (21).

$$S_{m\&r}^{*corrected} = \frac{S_{m\&r}^*}{\sqrt{\left[\frac{n_b}{n}\right]}} \quad (26)$$

where $S_{m\&r}^*$ denotes standard errors. If they are not corrected, the variance would be significantly higher, which would lead to a conservative approach to the selection of variables. The procedure can be used to estimate the results of a given model based on the estimates of its standard errors and its regression coefficients.

Generalized Linear Models (GLMs)

The ability of Y to follow different distributions is a key feature of the GLM framework. The functions of the design matrix \mathbf{X} and vector $\boldsymbol{\beta}$ remain the same. The assumptions concerning them are also the same. The link between $\mathbf{X}\boldsymbol{\beta}$ and Y is established using a function, g . The $E(Y)$ is computed by taking into account $\mathbf{X}\boldsymbol{\beta}$. The result of this function is a general formulation of a GLM, as shown in Equation (22).

$$g(E(Y)) = \mathbf{X}\boldsymbol{\beta} \quad (27)$$

A GLM is a type of model that takes into account the elements of \mathbf{Y} and then produces a linear regression model, a Poisson regression, or a logistic regression. In these cases, the link between the elements of \mathbf{Y} and the $\mathbf{X}\boldsymbol{\beta}$ is an identity function. In the simulation study, a generalized linear model (Poisson model) was examined. When performing the regression analysis on the real data, the link function was applied.

Application of Methods

This section presents the application of the selected techniques on both simulated and real data.

Data Simulation and Models

A simulation study was performed to analyze the applicability and appropriateness of the different methods in linear regression analysis. Table 2 shows the various parameters used in the simulation study.

Table 2: Overview of Simulated Parameters

Sample Size (n)	Number of Variables (p)	Error Term Standard Deviation (σ)
50,000	50	1
100,000	100	2
500,000	500	5
1,000,000	1,000	10

Source: Researcher's Construct (2021)

The data set's dimensions play different roles in the selection of optimal settings. The number of variables, p , can significantly affect the number of observations, k . But, the number of data points, n , is not dependent on variables, p . These two parameters are used to confirm the reduction of observations. The model's fit to the data set is also considered when it comes to the estimation of the good approximation. This parameter is included as a parameter in the model's design. All the simulated datasets were appropriate since they were based on the true model. However, it could be observed that there are some differences among the patterns exhibited by the datasets. This is due to the settings considered for the variance parameter, σ^2 . The values of β were set to zero with a probability of 0.5. They were sampled from a Poisson distribution having a rate of 3. These values were chosen to ensure that the Poisson

distribution is not inflated. To examine the effects of negative influences on the model, all components were multiplied by -1. The values in every column were then drawn from a demeaned normal distribution with a standard deviation of 25. A column of 1s was then used to model the intercept term. The η values were sampled from $N(0, \sigma^2)$. Y was computed as $Y = X\beta + \eta$.

A Bayesian model is a statistical method that uses a standard likelihood to estimate the likelihood of a given outcome. Y is assumed to follow a normal distribution,

$$Y \sim N(X\beta, \sigma^2 I_n) \quad (28)$$

In the prior distribution, an improper uniform distribution was used for both β and σ^2 . This led to a posterior distribution, which is proportional to its original distribution.

$$p(Y|X\beta) \propto L(\beta|X, Y) \cdot 1_p \quad (29)$$

The proper posterior distribution is provided by $L(\beta|X, Y)$, which ensures that the information in the data is sufficiently large to minimize potential challenges associated with a large data set.

Software and Packages Used

All the simulations and real data analyses in this thesis were conducted using the R statistical software. Some standard R software packages were used. The Bayesian regression analyses were performed by utilizing the R package *rstan*. The sketches were calculated using the R package *RaProR*. The Merge and Reduce principle was implemented using the *mrregression* R package. All the R packages used in this thesis were downloaded from the Comprehensive R Archive Network (CRAN) website.

Work Station

The simulations were conducted on an Apple MacBook Pro with Intel(R) Core(TM) i5-2435M CPU running at 2.40GHz using 4 GB RAM on a Windows 10 Home Operating System. The stan function from the rstan package was used to fit the Bayesian linear regression models. The stan function takes into account the various constraints of a Bayesian model and returns a set of models that are based on the No-U-Turn Sampler. The default settings were then used to run parallel chains.

Chapter Summary

The chapter has reviewed some techniques that are considered suitable for the regression analysis of big data. The techniques include the Merge and Reduce, a technique mainly used in data structures in Computer Science and some Random Projection methods, as well as their underlying assumptions and theoretical guarantees. The random projections are used to perform a pre-processing step before the analysis is performed by randomly projecting high-dimensional input data onto a low-dimensional sub-space with the pairwise distance almost preserved. The M&R technique enabled us to perform regression analysis on the whole big data by dividing the big data into blocks and performing the analysis on each block, after which the respective models are merged to obtain one model for the whole data. The chapter has also reviewed some computational methods, including MCMC techniques and Bayesian estimation of linear regression parameters, as well as the Euclidean distance. It is expected that the two techniques would make MLR analysis of big data more efficient in both the Classical and Bayesian settings.

CHAPTER FOUR

RESULTS AND DISCUSSION

Introduction

In all, 32 data sets were generated in our simulation study, one for each of the possibilities listed in Table 2 in Chapter Three. On each set of data points, a Bayesian linear regression analysis was performed to get the results for the original data set. Nevertheless, because of resource constraints, most of the analyses succeeded for most of the simulated data sets. Reduced data sets were created for each original data set using the Rademacher (RAD) and CW Random Projection techniques. Random projection error, ϵ , values of 0.1 and 0.2 were used which resulted in four reduced data sets for each simulated data set. The Bayesian linear regression analysis results for the reduced data sets were compared to the actual values of β . The results of the Bayesian linear regression analysis were compared with those of the original data sets.

Comparing the Running Times

First, a basic simulation experiment was performed to determine how long a Bayesian linear regression model takes to converge as the number of observations, n , increases. The running times used for various numbers of observations, $n \in [50, 50,000]$ with the number of variables, p , set at 52 as shown in Figure 2.

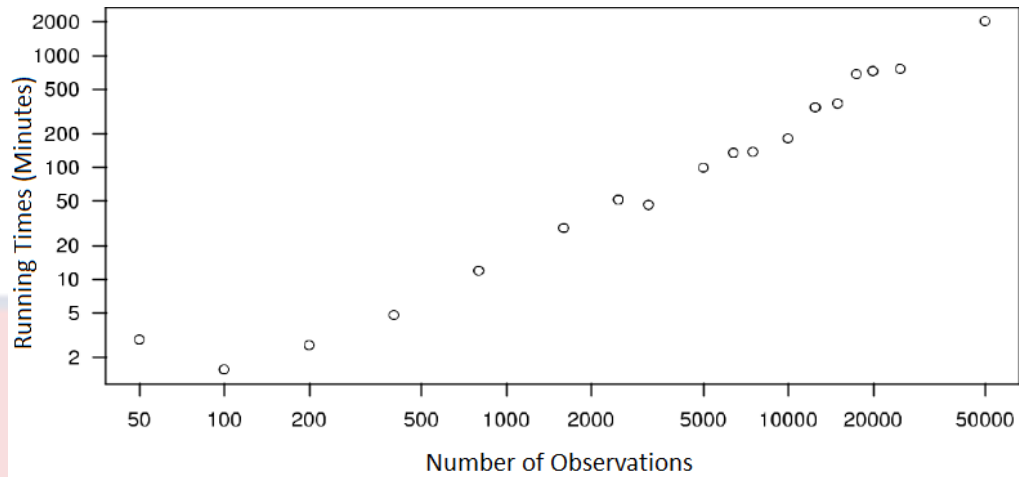


Figure 2: Running Times for Bayesian Linear Regression Models

From Table 2, the analysis running time appears to be linearly proportional to the number of observations, with rare leaps. The running time for 50 observations appears to be an outlying observation, which could be that the number of variables exceeded the number of data points. The dependence on the number of observations seems unproblematic for medium and small data sets. But in big data situations, only approaches with a sublinear dependence on the rising dimensions scale adequately and remain practicable (Cormode & Muthukrishnan, 2005). This shows the importance of projecting big data on a lower-dimensional subspace. Theoretically, doubling the number of data points should affect just the required time to obtain the sketched data set, but not the time needed to perform linear regression analysis on the sketched data set. Both embedded data sets are identical in size. Considering the condition at hand, when the linear regression analysis takes most of the total running time, doubling the number of data points has a negligible impact.

Table 3 shows the target dimensions, k , of the RP methods for various numbers of variables and projection errors. The required dimensions are functions of the number of variables, p , and the desired level of accuracy of the approximations, ϵ .

Table 3: Target Dimensions for the Random Projection Techniques

P	ϵ	RAD	CW
52	0.1	20547	16384
52	0.2	5137	4096
102	0.1	47175	65536
102	0.2	11794	16384

Source: Researcher's Computation (2021)

Appendix B summarises the running times by grouping them into “Preprocessing” and “Analysis” times. The “Analysis” running time shows the amount of time used to approximate a Bayesian linear regression model which converged, given the data sets. The time used for “Preprocessing” varies across the original and the sketched data sets. For original sets of data, the number represents the time used in reading and loading the data set into the working memory. For the sketched data, the number indicates the time used in constructing the projected data sets. The running time needed for the Bayesian linear regression analysis was computed by summing the times spent reading and embedding the data sets. It then comes up with a convergent model that takes into account all of the reduced data sets.

Figure 3 shows the running time for various sample sizes and the number of observations. The figure also reflects the consequence of doubling the sample size on the running time.

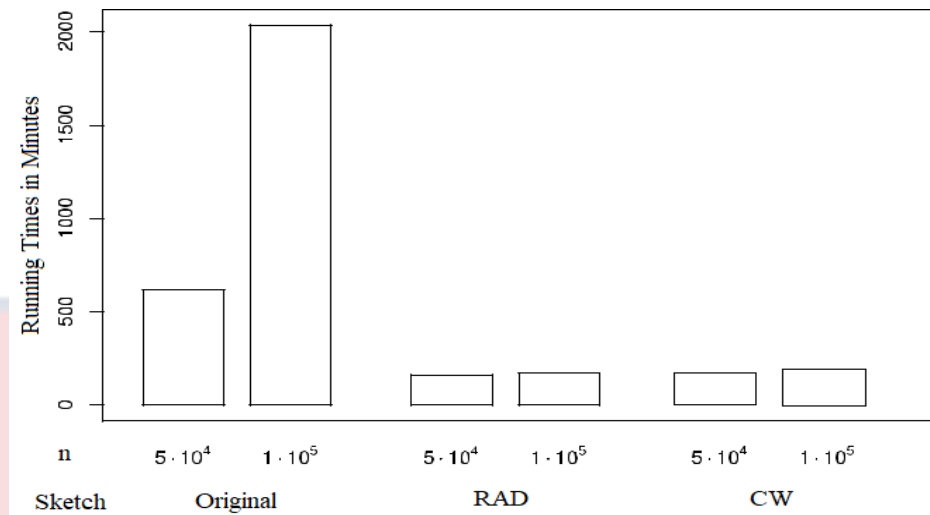


Figure 3: Total Running Times in Minutes

The running times presented for the original data set contain time spent reading and analyzing the data. In both the RAD and CW sketching techniques, the sketching time is added to the running time. Once the sample size is increased from 50,000 to 100,000, the overall running time more than doubles from over 600 minutes to over 2,000 minutes. The running time of the RAD and CW sketches exhibits no discernible pattern. Whereas the running time needed for embedding doubles for both techniques, that of the Bayesian linear regression analysis exhibit only a few minor deviations that appear to be random. The running time for the Bayesian linear analysis is higher than that for reading and sketching. Hence there seems to be no clear systematic impact. The overall running time for the embedded data sets includes the time spent reading, sketching, and approximating the linear regression model. The time for sketching for the original data is 0 because this step is not applicable in this condition.

Comparing the Posterior Means

Table 4 shows the Euclidean distances between the posterior means of the true data sets and the sketches.

Table 4: Euclidean Distances Between Posterior Means of the Approximated and the Actual Models

n	Sketch	ε	$\sigma = 1$	$\sigma = 2$	$\sigma = 5$	$\sigma = 10$
Squared Euclidean Distances						
5×10^4	RAD	0.1	0.052	0.025	0.021	0.834
5×10^4	RAD	0.2	0.014	0.781	0.892	1.512
5×10^4	CW	0.1	0.025	0.004	0.021	0.195
5×10^4	CW	0.2	0.016	0.040	0.156	0.915
1×10^5	RAD	0.1			0.836	0.958
1×10^5	RAD	0.2			0.061	0.777
1×10^5	CW	0.1			0.056	3.844
1×10^5	CW	0.2			2.624	2.937

Source: Researcher's Computation (2021)

The mean values of the posterior distributions on both embedded and actual data sets were compared. It was found that the values of β differed in some cases. The sum-of-squares Euclidean distances increase proportionally to the error term's standard deviations. There appear to be no consistent performance differences between the two Random Projection techniques. With greater ε , some distances increase, although this is not at all times. Some numbers are missing from the linear regression models due to the failure of the models to converge in a reasonable time. Additionally, the posterior mean values were compared to the true mean values.

Table 5 presents the squared Euclidean distances between the actual mean and the standard deviations in 52 variables.

Table 5: Squared Euclidean Distances Between True Mean Values and Posterior Means of Models Based on the Sketches

n	sketch	ε	$\sigma = 1$	$\sigma = 2$	$\sigma = 5$	$\sigma = 10$
Squared Euclidean Distances						
5×10^4	none		0.000	0.003	0.065	4.614
5×10^4	RAD	0.100	0.048	0.016	0.124	1.718
5×10^4	RAD	0.200	0.012	0.710	0.506	10.845
5×10^4	CW	0.100	0.022	0.011	0.046	6.474
5×10^4	CW	0.200	0.014	0.056	0.089	1.870
1×10^5	none				0.065	0.035
1×10^5	RAD	0.100	0.007	0.031	1.354	0.679
1×10^5	RAD	0.200	0.033	0.009	0.117	0.579
1×10^5	CW	0.100	0.004	0.232	0.022	4.496
1×10^5	CW	0.200	0.011	0.072	3.484	3.473
5×10^5	RAD	0.100	0.009	0.223	0.563	12.920
5×10^5	RAD	0.200	0.045	0.322	1.729	0.658
5×10^5	CW	0.100	0.027	0.097	1.305	0.153
5×10^5	CW	0.200	0.050	0.009	0.135	3.579
1×10^6	RAD	0.100	0.001	0.016	0.126	3.967
1×10^6	RAD	0.200	0.080	0.011	0.072	1.357
1×10^6	CW	0.100	0.002	0.289	1.202	4.445
1×10^6	CW	0.200	0.003	0.047	0.100	0.395

Source: Researcher's Computation (2021)

The overall result appears to be consistent with the findings in Table 5. While the original model consistently has the lowest squared Euclidean distances, the linear regression models based on sketched data sets are occasionally more

accurate. Thus, no consistent difference appears to exist between the projection techniques. With some assuming lower values as the sample size increases, the squared Euclidean distances seem unaffected by the value of the sample size.

Comparing the Fitted Values

Having made some parameter-level comparisons, where the number of parameters is unaffected by projecting, the linear regression models concerning the number of observations where the sketched data sets contain a proportion of the original data points were compared. The mean vector of β , which is derived from the data set or the sketch, is used to multiply X to determine the accuracy of the approximation in the response space. The scatterplot in Figure 4 shows two-dimensional estimates of the kernel densities derived from the data sets.

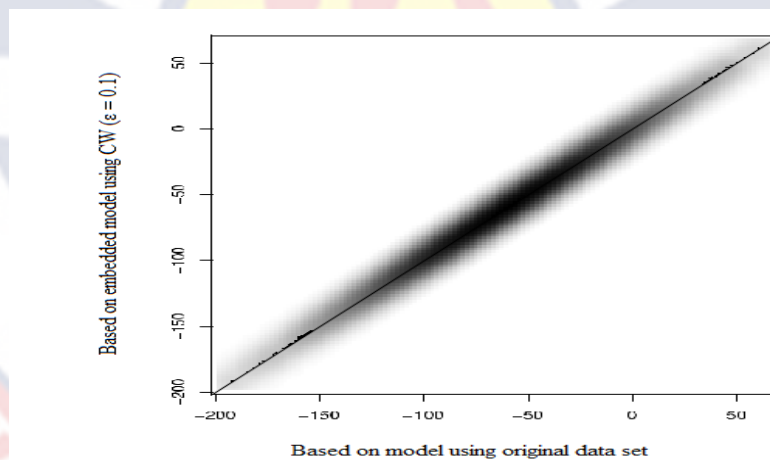


Figure 4: Scatterplot of Fitted Values Using the Original Data Set

Although the data sets used for the fitted values are prone to errors, all of them are close to the dividing line, which shows that the models are similar in terms of their fitted values. The darker shades of black represent the presence of more observations. Figure 5 presents the distances between the fitted values to provide a better overview of both RP techniques.

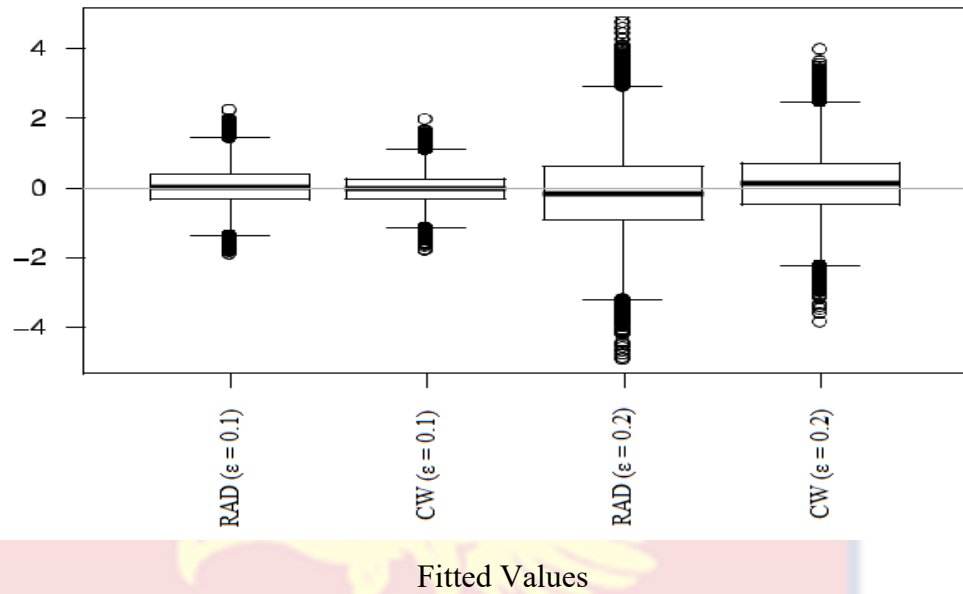


Figure 5: Fitted Values Based on the Original and Approximated Models

Each of the four sets of distances is clustered around the zero line. The boxplot reveals the effect of the goodness of approximation value; the variability is greater for $\varepsilon = 0.2$, independent of the random projection method used. Given a fixed ε , both sketching techniques produce almost the same results. The RAD technique is more variable than the CW method when it comes to sketching. This is because the former uses more variability while the latter uses more control.

The results of our study indicate that the RP techniques can sufficiently recover the posterior means. The study also shows that the observed variations in the data set are not significant. The posterior predictive distribution is appropriate for updated information. Compared to the posterior distribution, the posterior predictive distribution incorporates additional variability to account for the uncertainties in predicting unknown data.

Comparing the Posterior Distributions

A Bayesian model is a statistical model that takes into account the whole posterior distribution of a given parameter. In Figure 6, two boxplots show the

distributions of these models for two Parameters β_{11} and β_{22} . The number of observations is 50,000. 52 variables and a standard deviation of 5, as well as their respective sketches, were used.

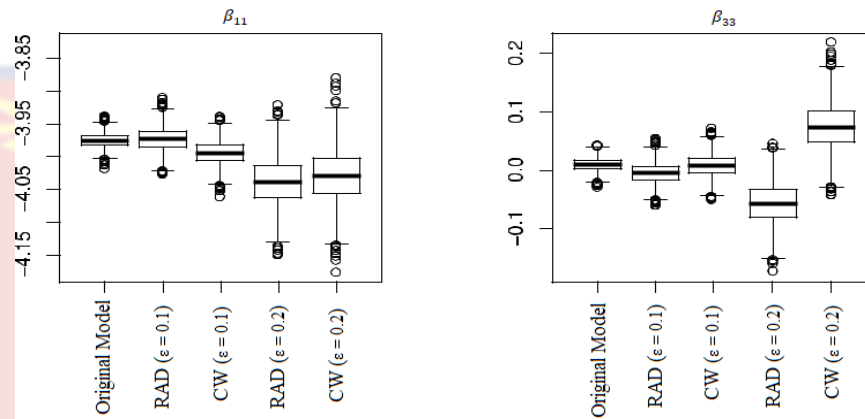


Figure 6: Boxplots of MCMC Samples for β_{11} and β_{22}

The medians of the MCMC samples are well represented by the sketches. No systematic biases were found. The introduction of additional variation does not seem to affect the sketching techniques.

One of the most common tasks in a regression analysis is identifying important variables. This can be done through variable selection, which can be performed in a Bayesian setting. The results of this study indicate that the accuracy of this process can be achieved through the use of the methods under study. One should take into account the additional variation that can be found in the variable selection process.

Streaming Big Data

Most of the simulations in the study are focused on the size of the data sets. This allows them to perform a quick analysis of the original data set, but it precludes the analysis of large data sets. To solve this issue, a big data simulation was conducted. In this simulation, some data sets were created based on the rules described in Chapter Three. The data set's dimensionality is

$10^9 \times 100$, and every entry has a double-precision value. This means that it requires at least 750 GB of RAM to perform the analysis.

In a format of comma-separated values CSV, the sketched data set has a dimension of $65,536 \times 100$. It can easily fit into the working memory of a computer. It took 2,781 minutes to perform a Bayesian regression analysis on the data. Although the result of the analysis is not the same as the one shown in the sketch, the distances between the β values and the posterior means are small. The random projection and reading of the data set only add a small multiplicative element to the running time of the analysis. The interval [1.01, 1.04] was the optimal range for small data sets. On the other hand, lower factors were typically observed for large data sets.

Analysis of Empirical Big Data Using Random Projection Method

In addition to a simulation study, some linear regression analyses were performed on a real data set that contains the number of bikes rented in the United States of America (USA), Washington. The data was sourced from a repository created by Fanaee-T and Gama in 2014. It consists of 17379 hours of observations. The variables used to determine the hourly number of rented bike users are put into registered and casual bike users. In our model, the total number of bike users is regarded as a count variable. A square root transform reveals that this variable is bi-modal. However, it also fits linearly well to a normal distribution.

Table 6 contains some information about the real data. The data set used for this analysis includes three standardized variables: apparent temperature, wind speed, and humidity.

Table 6: Variables in the Real Data Set

Variable	Description	Remark
<i>cnt.</i>	Number of rental bikes	Response variable
<i>season</i>	Seasons in the year	4-level factor
<i>yr.</i>	Year 2011 or 2012	2-level factor
<i>hour</i>	Hours from 0 to 23	24-level factor
<i>holiday</i>	Public holiday	2-level factor
<i>weekday</i>	Week days	7-level factor
<i>weathersit.</i>	Weather situations	3-level factor
<i>atemp.</i>	Apparent temperature	Standardized
<i>hum.</i>	Humidity	Standardized
<i>windspeed</i>	Windspeed	Standardized

Source: UCI Machine Learning Repository (Geppert, 2018)

The other variable that was excluded when it comes to the *weathersit* is the apparent temperature. This is because the data set had a lot of these factors, which are highly correlated with the model. The variable had 4 levels, namely *heavy rain*, *light rain*, *clear* and *cloudy*. The final variable that was excluded when it comes to the *weathersit* is the apparent temperature. It appears thrice in the data set. The third and fourth levels were joined to get *rainy weather* as the new level 3.

To create the correct design matrix for the various factor variables, intercept and dummy variables were introduced. The design matrix has a dimension of 17379×40 . Small sample sizes cause the embedded matrix to be larger than the original design, given an approximation error of 0.1. For the RAD and the CW sketches, approximation errors of 0.15 and 0.20 were chosen respectively. This resulted in the 6767 and 3807 observations for the reduced

data. For the various CW-sketched data, k is used to represent the target dimensions.

Table 7 shows the sketch sizes for the real data given the sketching method and the approximation error.

Table 7: Sample Sizes of the Sketches

P	ε	RAD	CW
40	0.15	6767	8192
40	0.20	3807	4096

Source: Researcher's Computation (2021)

In Chapter 3, how to check the accuracy of the results of the projected data sets against those of the true data set was discussed. This method is similar to the one used in Table 8. The distance between the sketched models and that of the true models is shown in Table 8.

Table 8: Sum of Euclidean Distances Between Posterior Means of the Original and Recovered Models

ε	RAD	CW
0.15	1.790	0.907
0.2	6.511	1.657

Source: Researcher's Computation (2021)

The difference between the projection errors of 0.15 and 0.20 for the RAD sketch is high, while the other sketching method does not increase as much when the projection error increases. The CW method offers the lowest square distances, which may be why the number of observations made using the CW technique is higher when compared to the RAD method.

The values in Figure 7 are shown in terms of their original and approximated models. They are then fitted using the data set X .

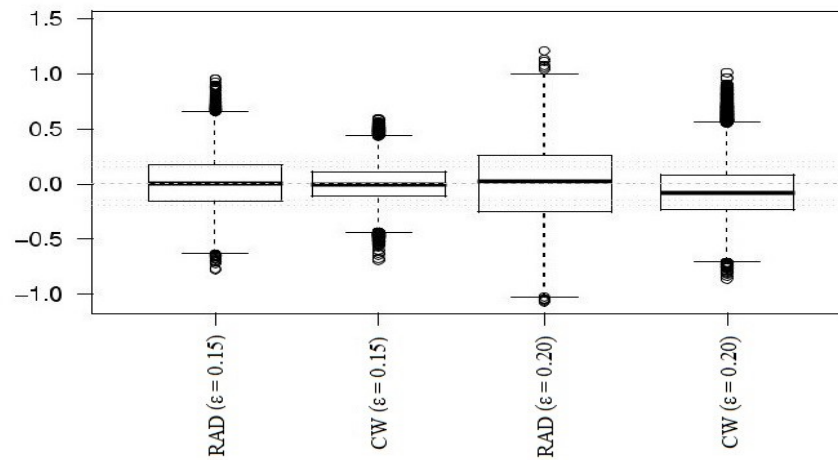


Figure 7: Difference in Fitted Values Based on Original and Reduced Models

The boxes are the smallest sizes for the CW sketches when a constant projection error is kept. The values for $\epsilon = 0.15$ and $\epsilon = 0.20$ are close to zero, while those for RAD sketches are different. For instance, some fitted values for $\epsilon = 0.15$ are more than 1. The effect becomes clearer as the data set's target dimension grows. The highest values could be observed when the number of bikes increases. Hence, the differences between the values for the RAD sketches are low.

The boxplots of the MCMC sample show the distributions of the variable *weathersit*. They are based on the original and sketched data sets. The reference category for this variable is clear weather: partly cloudy or sunny.

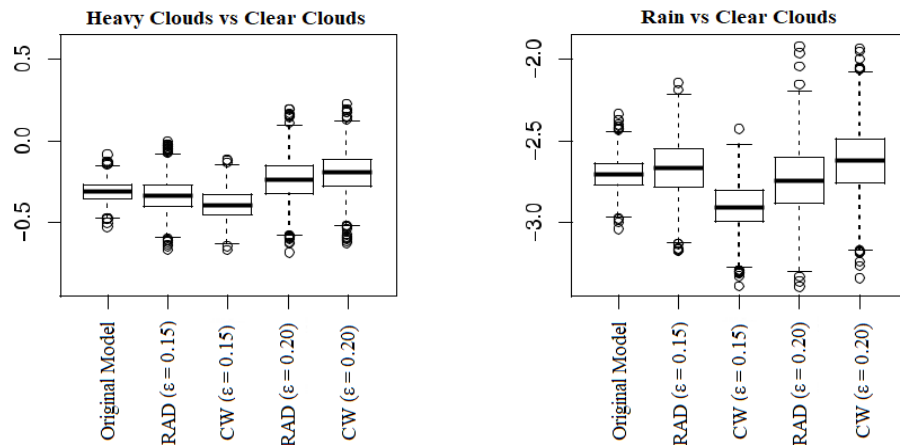


Figure 8: Boxplots of MCMC Samples for the Weather Parameters

According to the data set, about 66% of the hours feature a weather condition in category 1. 25% of the time, it is cloudy weather, while 6% are rain and cloudy weather. On the other hand, the number of rental bikes is not significantly different between clear and cloudy weather. The 95% credible interval of both the original and approximated linear model does not include 0, which is a concern when sketching. This means that a variable's credible interval could be zero in a model when the data set's original interval is close to zero.

The negative effect of the weather situation is shown in Figure 8. It can be seen that the results of the analysis are close to those of the original data set when the weather gets worse and there are thunderstorms. The influence of the parameter ε is also important. It introduces additional variability in the distribution. On the other hand, the sketching method seems to have a noticeable effect.

Linear Regression Analysis of Simulated Big Data Using the Merge and Reduce Method

The results of the simulation study are presented for the Merge and Reduce method. They show that the models performed well in linear and Poisson regression models. Two different measures are then used to evaluate

the performance of the approximated linear models. The first measure is the square of the distance between the estimates of the models, which are based on the M&R process. The second measure is used to evaluate the standard deviations of the models. It takes into account the recovery of the standard deviation through the Merge and Reduce techniques.

Results of Linear Regression Analysis for the Classical Merge and Reduce Approach

In Figure 9, the boxplots display the Euclidean distances between the original models and the approximated models used in the simulation using the Merge and Reduce technique 1.

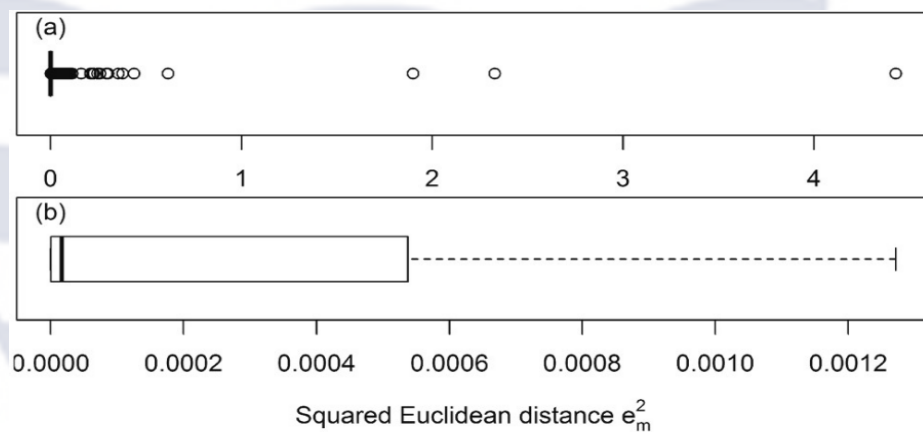


Figure 9: Boxplots of Euclidean Distances Between Simulated Regression Models

Subfigure (a) has all the squared Euclidean distances. For subfigure (b), the outliers are excluded. The Merge and Reduce approach is useful in performing frequentist linear regression or analyzing the posterior distribution in a Bayesian case. Figure 9 shows how well the method recovers the original estimate of the original variables.

Subfigure (a) shows the values of the squared Euclidean distances, which were computed in the simulation study that included various parameter settings. The box on the left shows the outliers. Subfigure (b) displays only the

box and whiskers of the distances. Out of the total values obtained, 75% of them are between 0 and 0.0006. On the other hand, the median squared Euclidean distance is very low, at around 0.00002. Observations that are outside the box's interquartile range are considered to be outlying.

A further look at quantiles shows that the original linear model does not account for the significant differences between the observed and proposed square-shaped distances. Table 9 shows some of the selected quantiles of squared Euclidean distances that are obtained from the Classical Merge and Reduce models.

Table 9: Quantiles of the Euclidean Distances Between Parameter Estimates

Quantiles	Min.	50%	75%	90%	95%	97.5%	99%	Max
e^2	0.0000	0.0000	0.0006	0.0100	0.0355	0.0870	0.2956	4.4287

Source: Researcher's Computation (2021)

For most simulation settings, the Merge and Reduce technique accurately approximates the results of the true linear model. The squared Euclidean distance is also lower in many settings.

In Figure 9, subfigure (a), it can be seen that there are three different cases where Euclidean distances are more than 1: error variance of 100, 400 variables, and block size of 200 data points. In most cases, the associated distance is not high enough to make these settings necessary. Other factors such as high error variance and a low number of observations are also taken into account to determine the optimal settings. If a large data set is divided into smaller blocks, there would be a shift from the model's true design. This is because the variance of the error term can be high. An observations-per-variable ratio between 10 and 20 depending on continuous response variables is

recommended. Harrell (2001) notes that models having lower than normal ratios of observations per variable are unreliable. This is because their predictive performance for a new observation is low. The simulation study conducted on this issue revealed that the use of the Merge and Reduce approach improves the reliability of linear models.

In Figure 10, it is shown that the decision regarding the optimal ratio between block size per variable and the squared Euclidean distance is the determinant. It can be seen that the values of the squared Euclidean distances are higher if the ratio of blocksize to a variable is 2, while the distances decrease for a blocksize to a variable ratio of 4. The results of the study suggest that for blocksize to the variable ratio between 10 and 25 inclusive, the squared distances may be high than 0.1, but for a ratio greater than 25, no deviations were observed above 0.1. The reason why the error term variance is important is that it increases the likelihood that there would be higher squared distances between the two models. Even with the highest error term variance of 100, low squared distances are observed. The data set did not suggest that the number of observations affects the distance between the two models.

In most cases, the distance between the estimated intercept and the actual intercept is the determinant of the squared Euclidean distances. This is, in fact, the case in which the square Euclidean distance is the highest. Surprisingly, the influence of the intercept on squared Euclidean distance is not related to the number of variables. Considering the artificial data set with 200 variables, the proportion of Euclidean distances influenced by the intercept term is greater than 0.8. Utilizing only the estimates of each variable generates lower distances, of which 0.107 is the maximum value. In Figure 10, the effect of

block size per variable on the squared Euclidean distance is shown for Merge and Reduce approach 1

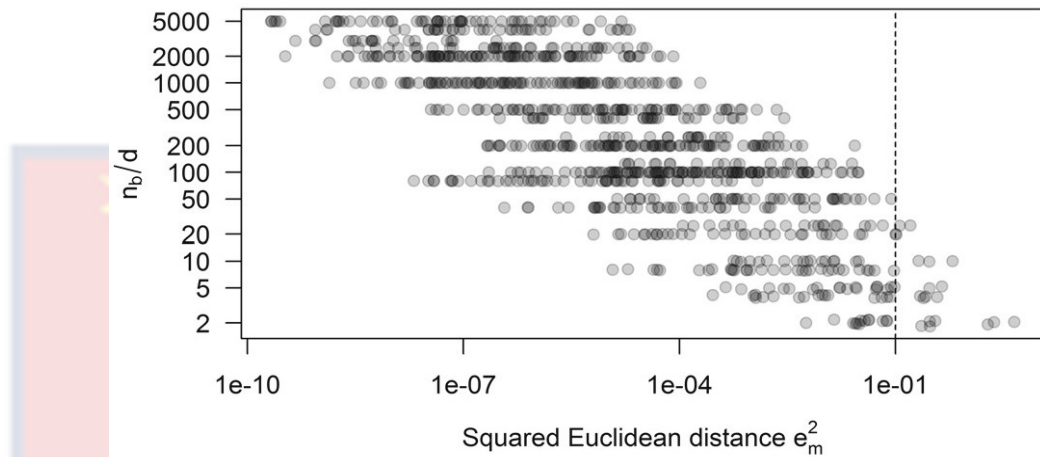


Figure 10: Scatterplot of the Influence of Block Size Per Variable on Euclidean Distances

After examining the difference between the two estimators, the estimated standard errors are considered. The data are drawn as transparent points. The black area represents many data points at one location, while the grey points represent single observations. To make the correct standard error factor, all of the variables are considered. The standard errors tend to behave uniformly for the intercepts and variables.

Subfigure (a) displays the corrected error values for all the simulation settings. The peak value of the kernel density estimate for the errors peaked at 1. However, the corrected standard error values for some settings are higher than the estimated standard error. This is because the expected standard error is higher in these settings. To gain a better understanding of this issue, the split between the values of the corrected standard errors and blocksize per variable is used to estimate the true value of the error.

The corrected error factors, f_{se}^m , are shown in Figure 11. They are used in all simulated settings of the Merge and Reduce approach 1.

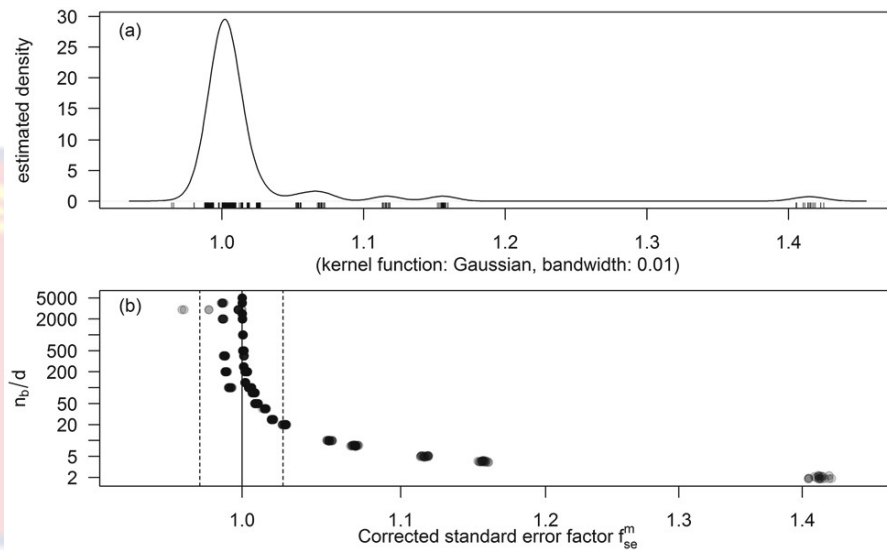


Figure 11: Adjusted Standard Error Factors for Simulated Data

Concerning the number of variables, subfigure (b) reveals the influence of the block size. There is a negative relationship between the values of the block size per variable and the corrected standard error factor, as the latter tends to be close to 1 for a high block size for each variable. As the block's size decreases, the value of the corrected standard error factor increases. Some values of the corrected standard error factor below 1 are not integer-valued. For instance, if the ratio of the sample size to block size is not an integer-valued number, then dividing by such ratio is a slight overcorrection. As a result, the significance of the variables is more important than the p-values of the associated t-tests. The values of the adjusted standard error factor are close to 1. This means that for a corrected standard error value of 0.9876, the estimated errors are 1.24% lower than the original model.

The squared distance between the estimate and its approximate estimate is analyzed using the corrected standard error factor. This shows that the quality of the estimate depends on the block size per variable. The estimated standard

errors for block size per variable of fewer than 10 observations are typically inflated by around 5% to 40%. If the inflation is less than 2.5%, then the value is considered acceptable, which is 20 observations per block for each variable. In the case of the corrected standard errors, the various variables and intercept behave similarly concerning every entry. On the other hand, the error term variance does not seem to have any effect on the computation.

The standard error factor used in the model reconstruction helped to recover the estimated effects of the true models. However, if the number of observations is high enough, the estimated errors increase. This is because the intercept is distant from the original model's estimate. A minimum block size of 20 observations per variable must be maintained.

Results of the Linear Regression Analysis for the Bayesian Merge and Reduce Approach

In the Bayesian framework, the artificial data used in the Classical framework was employed. In the MCMC sample, the results of the measure of location returned by the method were examined. The difference between the median and the posterior quantiles is also corrected. The distance between the posterior medians and the mean values of the simulations is shown in Figure 12. The distance between these values is computed using the Merge and Reduce technique 2 as well as the true Bayesian models.

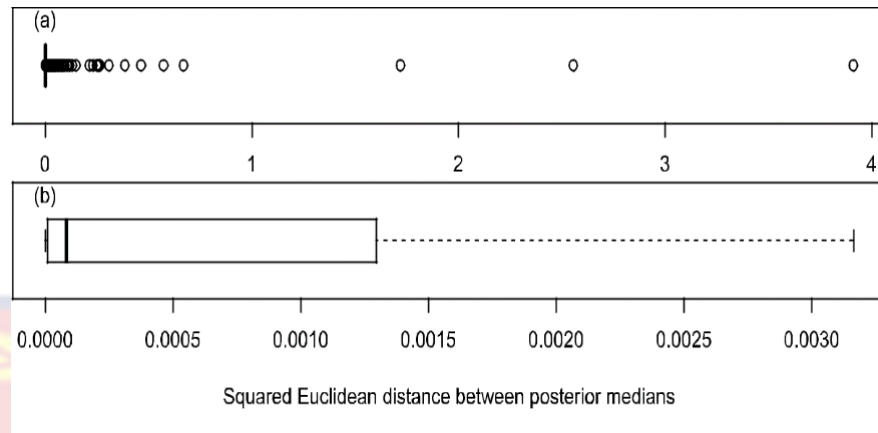


Figure 12: Squared Euclidean Distances Between Posterior Medians

The distance between the Euclidean distances and the posterior medians is studied in the simulation study that uses the Bayesian model and the Merge and Reduce approach 2. In the frequentist case, the values of Euclidean distances are found to be below 4. Subfigure (b) shows that the difference between the Merge and Reduce model, and the original model is not as large as it would appear in the frequentist case. The 95% quantile of squared Euclidean distances is also very small. In most of the simulations, the differences between the two models are very small.

The plot in Figure 13 shows the distance between the posterior median and the squared Euclidean distances of the simulated data sets. This is done using the second Merge and Reduce approach and the true Bayesian models.

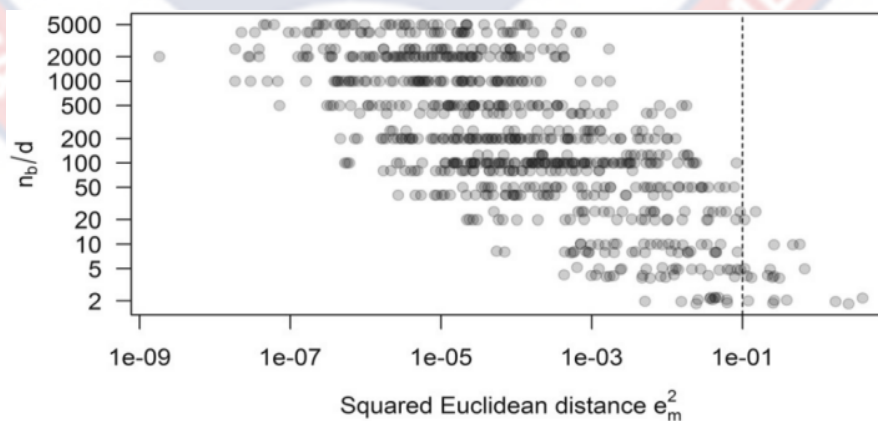


Figure 13: Relationship Between Block Size Per Variable and Distances Between Posterior Medians

The data are drawn as transparent points, with the grey ones representing single data points. The black area denotes many data points at one location. The results of the first merge and reduce approach are similar to those of the second approach. The median distance is well recovered by the second approach but is unreliable for smaller block sizes per variable. The minimum block size per variable must be at least 25 observations

The characteristics and standard deviations of the various models based on the Bayesian linear model and Merge and Reduce models were examined. The corrected standard error factors values are shown in Figure 14. They are grouped by block size and the number of variables. Subfigure (a) displays the connection between the block size and the corrected standard error factor. Subfigure (b) presents the influence on the corrected standard error factor by the number of variables.

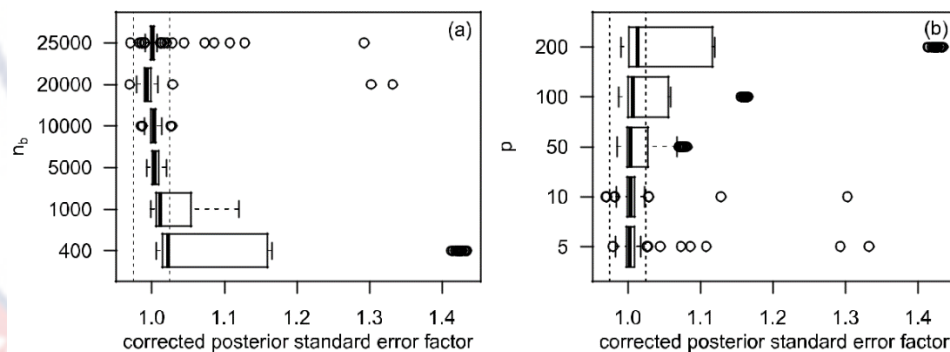


Figure 14: Boxplots of Adjusted Standard Errors

Most parameter settings of the number of variables and block sizes except for block size of 5000 observations exhibit an unacceptably high or low value of corrected standard error factor. Where the standard deviations are computed according to the number of observations, in the case of frequentist regression, the results are highly inflated even when the number of variables is low. The standard deviation of the posterior quartiles does not appear to be a

reliable and useful summarizer for the Merge and Reduce approaches. For the correction, the distance between the mean and median quantiles seems to be growing.

The results show that the square distances between the lower and higher quantiles are similar to those in the posterior median and mean. The variables and block sizes are significant in the overall results. The values of the low block size per variable values may become too high if the distance between the M&R result and the original model gets too high. The results of the study revealed that the Merge and Reduce models performed well to approximate the original distributions of the Bayesian linear models, including those of the 97.5% and 2.5% quantiles.

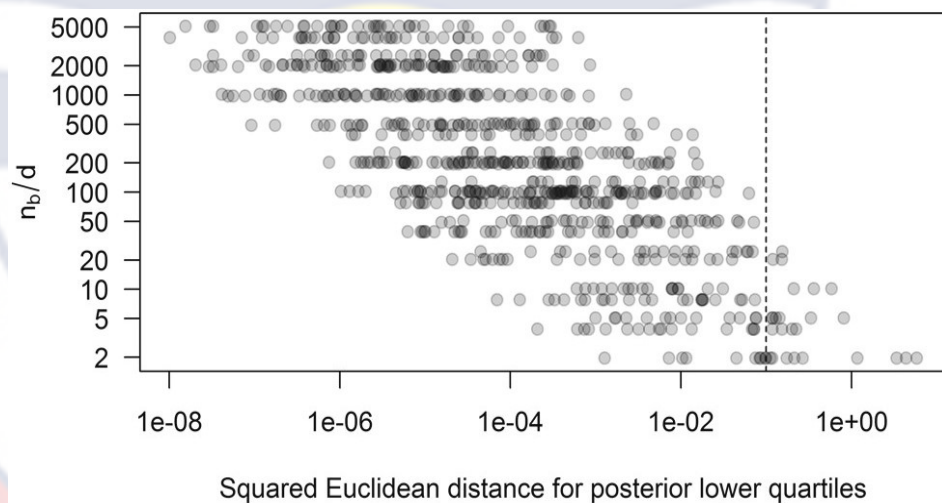


Figure 15: Scatterplot of Relationship Between Block Size Per Variable and Distances Between Posterior Lower Quartiles

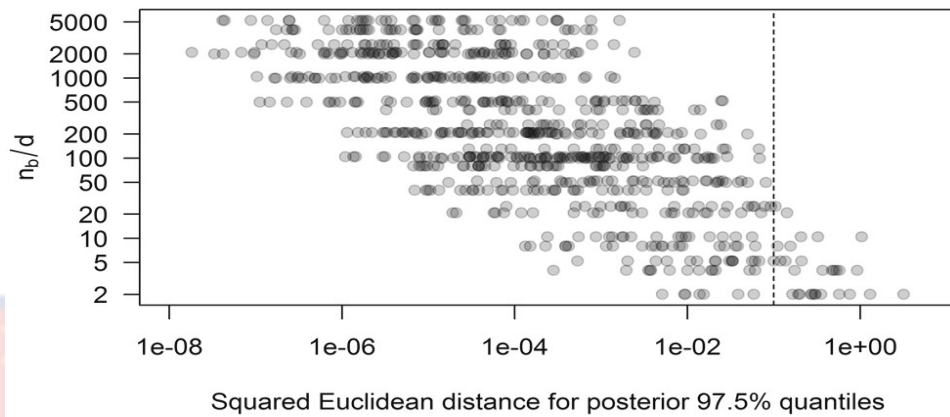


Figure 16: Scatterplot of Relationship Block Size Per Variable and Distances Between Posterior 97.5% Quantiles.

In figures 15 and 16, the observations are drawn as transparent points. The black area represents several data points at the same location. The grey area represents single observations.

Results of Linear Regression Analysis Involving Outliers

This section presents the results of the linear regression analysis of big data containing some outliers. The data sets are simulated with different values and are bound by the same linear model. However, they also contain outlying data points that are different from the values **X** and **Y** to minimize the effects of these outliers.

Four different ways of merging the data sets were created. The merge and reduce principle ensures that the blocks' order does not influence the results of a given operation. Changing the position of one of the mixture components does not correspond to the order of blocks. The simulated sets may not be identical to the observations made in the real world. Also, since the changes are not performed according to the same principles, they are not equal to the changes in the block order. The values of the "pos" variable affect the degree of homogeneity in the blocks. For instance, the random arrangement of the blocks results in some of them having a small number of outliers, while the other blocks

have a high relative frequency of these outliers. For instance, if the setting is first and last, then the behaviour of the data sets will be similar to that of the previous setting. On the other hand, if the setting is random, then the behaviour of the data sets will vary.

Results of Linear Regression Analysis Involving Outliers for the Classical Merge and Reduce Approach

The square-wise Euclidean distances shown in Figure 17 are the simulated data sets' absolute values. They are broken down into the outliers' positions.

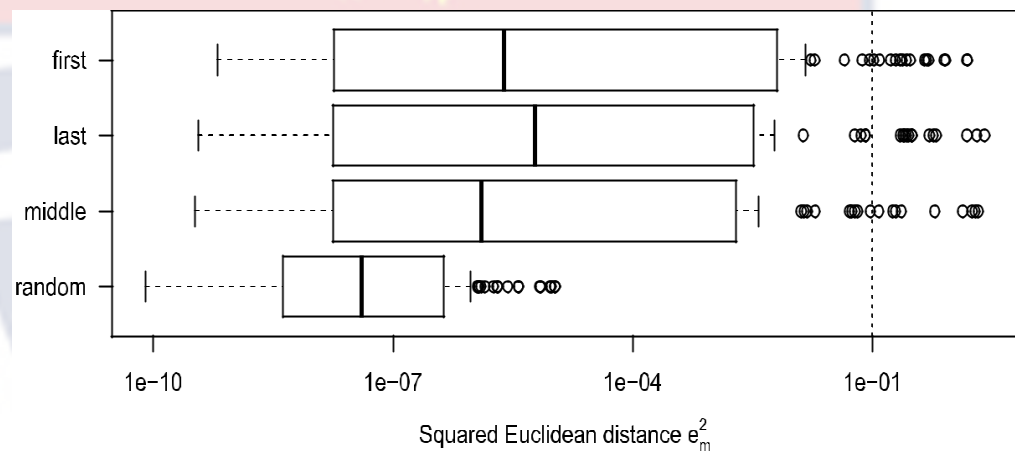


Figure 17: Boxplot of Squared Euclidean Distances Between M&R 1 Models Involving Outliers

Although the position of the outliers can affect the values of the Euclidean distance, they do not have a significant impact on the estimates. For instance, the largest square distance of 0.112 is only slightly above our boundary of 0.1. The M&R 1 seems to be more effective when extracting the original model results from the data sets that contain a random distribution of outliers. Data sets with the first or last outlying data points display the maximum values of Euclidean distance and hence are less likely to benefit from the M&R method.

Figure 18 shows the standard error factors for the simulated linear models with different orders of the outliers.

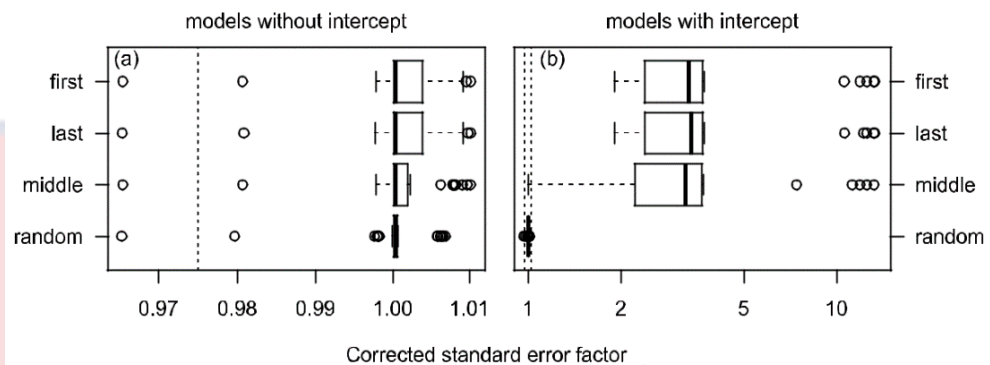


Figure 18: Boxplots of Euclidean Distances Between Standard Errors of M&R 1 Models Involving Outliers

Figures 18a and 18b show all models that contain and models that do not have an intercept term, respectively. Where a model does not contain any intercept term, its consequences are significant. The smallest values of f_{se}^m are found in the centre of the data set. The outliers in the study lead to the smallest values. For all simulations without any intercepts, the standard error factors are below 1.025. The standard errors are generally recovered by the M&R 1 technique when the data sets have an unusual arrangement of outliers. For instance, in data sets where all the outliers are grouped, the standard errors are overestimated by the M&R method.

The M&R 1 technique can recover the estimated errors of the original models in all cases where the data sets have an unusual arrangement of outliers. Where a model has no intercept term, the standard errors can be overestimated. The outlier position also affects the recovery of the model's results. The results of the study indicate that the complexity of merging heterogeneous blocks is increased due to the data set's homogeneous composition. The results of the blocks are not affected by their order. The high similarity between the simulated

data sets and observations from distant regions shows that the results of these observations are similar to those of the simulated data sets.

Results of Poisson Linear Regression Analysis

In this section, a simulation model that considers the data sets that are dependent on the Y variable is presented. Poisson regression analysis is performed to find out the optimal values for these sets. Compared to previous studies, this simulation study is relatively small, with six sets having 50000 and 100000 numbers of observations. The simulation also employs 5, 10 and 20 numbers of variables. The block sizes are 400, 1000 and 5000 observations.

The standard deviation of the results of a Poisson distribution is not chosen because it is equal to the mean for that model. To evaluate the effectiveness of the Merge and Reduce approaches in recovering the outcomes of the original models, the Euclidean distances between the parameter estimates as well as standard errors are used. The standard error factor for the M&R methods is used to evaluate their effectiveness.

Results of Poisson Linear Regression Analysis for the Classical Merge and Reduce Approach

Figure 19 shows the various squared distances taken for different types of Poisson models in the simulations. The values of these distances are taken into account in the Classical Merge and Reduce approach. Even with the low block sizes, the estimates of these models are fairly accurate.

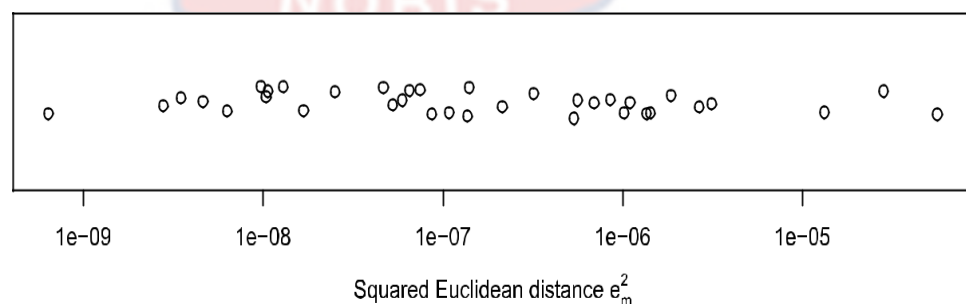


Figure 19: Scatterplot of Squared Euclidean Distances Between Poisson Linear Regression Models

Figure 20 displays the adjusted standard error factor for the Poisson models using the Merge and Reduce approach 1.

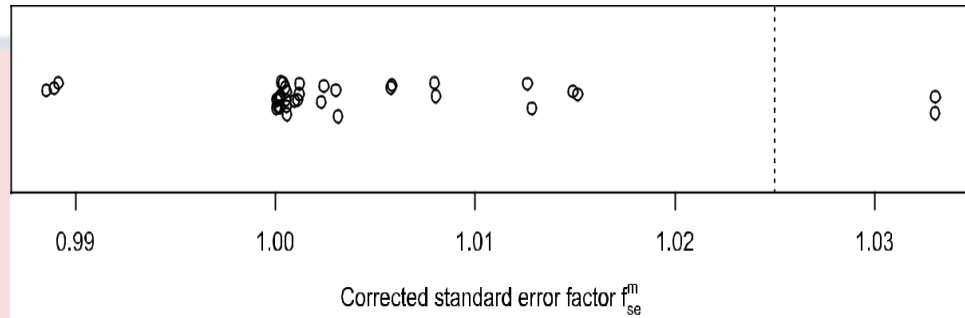


Figure 20: Scatterplot of Euclidean Distances Between Adjusted Standard Error Factors for the Poisson Models

Figure 20 reveals that all but two of the values in the range of 0.975 to 1.025 are within the acceptable range. However, the two outlying observations are from artificial data sets that have $p = 20$ and $n_b = 400$, which indicates that the standard error factor is more dependent on the value of $\frac{n_b}{p}$ than on the parameter estimate. The values of f_{se}^m which are closest to the acceptable range, are around 1.015, and this is within our interval of normal values.

Results of Poisson Linear Regression Analysis for the Bayesian Merge and Reduce Approach

In this section, the performance of the Bayesian Merge and Reduce approach on Poisson regression models is examined. The first step is to look at the distance between the original and the approximated posterior means. A scatterplot of the Euclidean distances is shown in Figure 21.

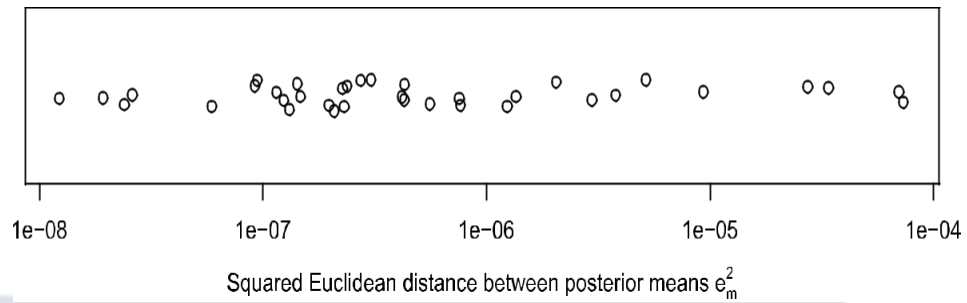


Figure 21: Scatterplot of the Distances Between the Posterior Means of the Poisson Models

The values in figure 21 are very small, and the largest one is below 0.0001. Due to the reduction of the posterior location, the values are well estimated by the Merge and Reduce approach 2. Figure 22 shows the distances between the Bayesian regression models and the 25% and 97.5% quantiles of the original models.

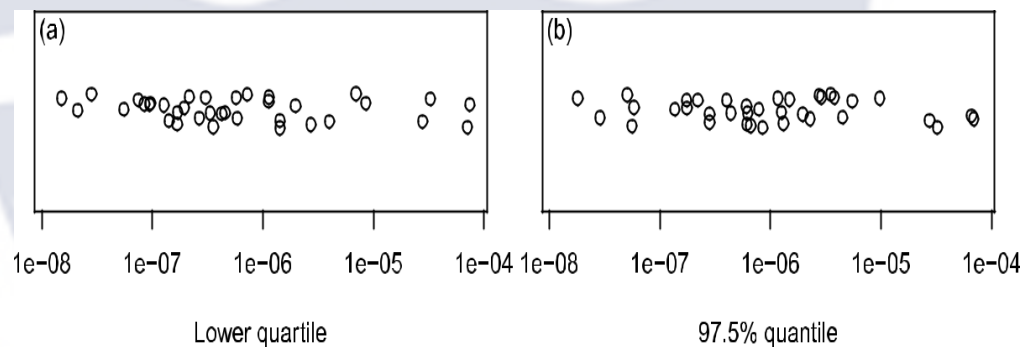


Figure 22: Scatterplot of the Euclidean Distances Between Posterior 25% quartile and 97.5% Quantiles of the Poisson Models

Apart from the location measures, the study also assessed the standard deviation of the quantiles for both the Classical and Bayesian models, which were poorly performed with the Bayesian Merge and Reduce approach. Because of the correction, all of the squared distances get values less than 0.0001. This ensures that the Poisson regression models are recovered well in the Classical and Bayesian frameworks.

Analysis of Empirical Big Data Using the Merge and Reduce Method

A bicycle rental data set was used to evaluate the various factors that affect the quality of life in a community. To do so, the variables that appear in the data set, such as the *year* were considered. This is because the variables appear systematically, and they indicate whether the observation is a first or second year. The variable that affects the number of bicycle rentals in a community stays constant throughout the data set. This makes it difficult to estimate the effect of this variable on the number of bike rentals in a given block. Also, since the data set is not always complete, it can be hard to identify the variable in the data. Since the variables that appear in the data set are not in a systematic manner, the various factors that affect the quality of life in a community were modelled using two different models. One of these is a linear model, while the other is a more similar model without the variables *atemp*, *hum*, and *wind speed*. Table 10 shows the results of the M&R analysis that are closest to the Classical linear regression model.

Table 10: Results of the Classical Linear Regression Analyses of the Empirical Data

n_b	Quantitative Variables				Addition of Factor Variables			
	Logarithmic		Poisson		Logarithmic		Poisson	
	e^2	f_{se}	e^2	f_{se}	e^2	f_{se}	e^2	f_{se}
10,000	0.0676	1.0078	0.0443	1.0184	0.1595	0.9412	0.0988	1.0139
5,000	0.0796	1.0279	0.0316	1.0574	0.0898	0.9193	0.0288	1.0182
1,000	2.6690	1.4216	0.3666	1.5174	1.9845	0.9399	0.7687	1.1529
400	5.9530	1.5078	0.9500	1.6528	10.2661	1.0118	7.7288	1.2610

Source: Researcher's Computation (2021)

The column headed e^2 displays the Euclidean distances between the model and the block size of the original model. Similarly, the column headed f_{se} displays the adjusted standard error factors. This exercise is performed on four different models. The two models that use only quantitative variables are compared with two other models that use factor variables and have independent variables. Linear regression analyses are performed on the four models. The data set used for the analysis consists of 17379 observations. Block sizes of 400, 1000, 5000 and 10000 observations were used to get the correct block sizes. The table below shows the results of the M&R analysis of the linear regression model. They are compared with the original model.

Table 11: Results of the Bayesian Linear Regression Analyses on the Empirical Data

Variable	Model	n_b	$\tilde{x}_{0.025}$	$\tilde{x}_{0.25}$	\bar{x}	$\tilde{x}_{0.5}$	$\tilde{x}_{0.75}$	$\tilde{x}_{0.975}$	s
Quan.	Lg.	10000	0.0683	0.0683	0.0678	0.0676	0.0672	0.0665	1.0016
Quan.	Lg.	5000	0.0781	0.0795	0.0817	0.0803	0.0836	0.0870	1.0146
Quan.	Lg.	1000	2.4054	2.5725	2.6727	2.6763	2.7802	2.9835	1.3294
Quan.	Lg.	400	5.4859	5.7775	5.9470	5.9439	6.1245	6.4713	1.3971
Quan.	Poi.	10000	0.0447	0.0444	0.0442	0.0445	0.0441	0.0441	1.0267
Quan.	Poi.	5000	0.0319	0.0319	0.0314	0.0314	0.0311	0.0310	1.0633
Quan.	Poi.	1000	0.3594	0.3640	0.3665	0.3665	0.3690	0.3744	1.5346
Quan.	Poi.	400	0.9366	0.9457	0.9504	0.9504	0.9545	0.9637	1.6700
Facto.	Lg.	10000	0.1673	0.1637	0.1617	0.1625	0.1605	0.1570	0.9293
Facto.	Lg.	5000	0.0991	0.0950	0.0939	0.0932	0.0918	0.0873	0.9126
Facto.	Poi.	10000	0.0985	0.0985	0.0981	0.0983	0.0986	0.0990	1.0205
Facto.	Poi.	5000	0.0291	0.0289	0.0288	0.0288	0.0287	0.0286	1.0141

Source: Researcher's Computation (2021)

Four models are analyzed, with two of these being only quantitative-variable models and two being factor models. Linear regression analysis is performed for each of these models, while a Poisson regression analysis is also conducted. The Euclidean distance between the two models is shown in the rows. The distance between the model that was originally studied and the one that was obtained using the Merge and Reduce approach 2 is also shown. The variance and posterior mean values are also given for each model. The two-factor models that failed to converge resulted in significant deviations from the original models. Because of this, they are presented not here.

The results of both the classical and Bayesian models are similar when it comes to Poisson and linear regression. For instance, the model based on M&R is generally good for the smaller block sizes (from 5000 to 10000) but not for the larger block sizes. On the other hand, the results of the model that includes factor variables are different when it comes to the three quantitative variables. The results of the model with factors are significantly different when compared to the original model when it comes to the block sizes. This issue is caused by the variable *holiday* not being present in all blocks. This leads to different models that deal with the conflicting aspects of the posterior distribution of β . Despite the presence of two variables that were excluded due to their inappropriateness, the results of the model were still good. This demonstrates the importance of carefully selecting the model and ensuring that it has enough information. Another important factor that should be considered is the number of observations the model should make per variable. Steyerberg et al. (2001) recommend a minimum of 20 observations per variable.

The number of observations that the model makes per variable is known as the effective number. For models that only contain quantitative variables, this number can be used. However, if the model includes factor variables, this number of observations changes. For instance, given n_i sample size and k factor levels, the effective sample size for a binary variable is $m = \min(n_1, n_2)$ while in the case of a factor variable, it is $m = n - \frac{1}{n^2} \sum_{i=1}^k n_i^3$ $k > 2$ (Steyerberg et al., 2001). Table 12 below shows the smallest effective number of observations, m , that can be obtained from various blocks of observations with the values of observations per variable and block.

Table 12: Smallest and Minimal Effective Sample Sizes for Different Block Sizes

Block Size n_b	Quantitative Variables Only		Including Factor Variables	
	$\min n_{eff}$	$\frac{\min n_{eff}}{p}$	$\min n_{eff}$	$\frac{\min n_{eff}}{p}$
400	179	44.75	0	0
1000	379	94.75	0	0
5000	2379	594.75	95	2.64
10000	7379	1844.75	191	5.31

Source: Researcher's Computation (2021)

For each block, the effective number of observations for all the block sizes is calculated. The model that takes into account only quantitative variables has a constant effective number of observations. On the other hand, the model with factors varies depending on the frequency of the variable levels per block.

Table 12 shows that the block sizes 400 and 1000 appear inadequate for the real data analysis. For instance, when the number of observations per block is 400, 50% of the blocks lack the variable *holiday*. On the other hand, when it

is 1000, only one block has a holiday. Even the blocks with a holiday typically only have 24 observations. This suggests that the model should be taken into account when there are potentially unbalanced binary variables in it. Although the results of the smaller models containing continuous variables are not as extreme, they show the same pattern for block sizes, 5000 and 10,000 observations. Some high deviations between the acceptable approximation and the actual sizes are revealed for the 400 and 1000 block sizes.

The number of observations required for each variable to perform the analysis is critical to obtain the best results when a complex model is involved. The other important factor that can help in recovering the results is the M&R technique. It can recover the more complex models even when the ratios of the variables are lower than those of the quantitative models. The number of observations also plays a role in the recovery of the results. In addition, the model's good-of-fit is also important in helping in recovering the results.

The goal of the Merge and Reduce method was to perform a comprehensive analysis of large data sets that are not feasible to analyze on a single basis. In this case, choosing the appropriate number of observations for a block is not a problem. On the other hand, choosing a small block size would be impractical.

Chapter Summary

In this chapter, it has been found that the running times of the Rademacher (RAD) and Clarkson-Woodruff (CW) random projection techniques exhibit no clear pattern. An inconsistent performance difference between the two random projection techniques was observed. Nevertheless, closer observation shows that the CW technique tends to produce larger

sketches and works faster than the RAD technique does. In most instances, the CW method produces the lowest square distances between posterior means and medians.

Concerning the merge and reduce techniques, the Bayesian merge and reduce models are found to recover the posterior means and medians well. Similarly, the Classical merge and reduce models well recovered the parameter estimates and their estimated standard errors. For random distribution of outliers, the Classical merge and reduce models recorded the lowest Euclidean distance values. However, for first or last order of outliers, the models showed high Euclidean distance values.

Empirically, the results of both the Classical and Bayesian merge and reduce models are found to be similar for linear and Poisson regression models. They showed good approximation for block sizes of 5000 and 10000 observations, particularly for quantitative variables. However, they showed high deviations for blocks of sizes of 400 and 1000 observations. The Bayesian merge and reduce models are found to better recover more complex model including factor variables.

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

Summary

This chapter summarizes the findings of the study and draws some conclusions based on the results presented in the fourth chapter of the study. Some recommendations are then provided for further research studies.

This study has sought to compare the performances of Random Projection and Merge and Reduce methods for estimating regression models for big data. The two techniques could be used in performing linear regression analysis on large amounts of data. Both techniques apply to high-dimensional data with a few numbers of variables and a relatively very high number of data points. The JL theorem has been used to construct random projections. In this thesis, a data reduction method useful for linear regression analysis through a random projection is presented. This method is beneficial for preserving data structure.

By including a sketching phase before the analysis, it is possible to efficiently model, using linear regression analysis, big data involving high dimensions. This method aims to improve the efficiency of the linear regression model by reducing the data set. Theoretically, Random projections provide some guarantees for required approximations. The requisite number of data points in the reduced set is determined by the number of variables p and the desired goodness-of-approximation factor. That is, the size of the reduced data set is not affected by the size of the original data set. This makes the Random Projection techniques particularly important for the linear regression analysis of big data sets with a few variables. The technique is comparatively robust to

prior distribution adjustments. In most cases, only highly informative priors or distributions that conflict with the likelihood presents significant concerns as a result of the original high-dimensional data set's projection into the low-dimensional subspace. Such a condition would be unacceptable as well as challenging from a modelling standpoint.

The performance of the Merge and Reduce method was examined on both simulated and empirical data, which ensured an efficient computation-intensive linear regression analysis on big data streams with a large number of observations. The data is partitioned into manageable blocks from which individual summaries are computed employing standard data analysis procedures. The generated summaries are joined in such a style that the working memory requirements of the models stay under an insignificant factor. It is important to select suitable statistical summarizers and perform various reduction actions to minimize the effects of the model.

Conclusions

The conclusions of the study are drawn from the results. The discussion around the findings also leads to various conclusions. The Clarkson-Woodruff (CW) and the Rademacher (RAD) random projection techniques are good for reducing big data sets before performing either Classical or Bayesian multiple linear regression analyses. But the Clarkson-Woodruff method performs faster and provides more reliable reduced data sets. For the Bayesian multiple linear regression analysis, the Clarkson-Woodruff and the Rademacher methods performed better on the simulated data. For high variance of the error term, both random projection techniques produce results similar to those found in the original data set. Therefore, the Clarkson-Woodruff and the Rademacher

techniques can be used to reduce a big data set before conducting multiple linear analyses.

For Poisson linear regression models, both the Classical and Bayesian Merge and Reduce approach perform better provided that there are enough observations per variable per block. Although the Classical Merge and Reduce approach show a good approximation of the true linear regression models, the standard errors are overestimated when there are outliers in the big data set for models without an intercept term.

The Bayesian Merge and Reduce approach improves the accuracy of the linear models by considering enough observations per variable per block. The Bayesian Merge and Reduce results are similar to the original linear model's results. The standard deviations of the various Bayesian Poisson Merge and Reduce models are close to the same level as the original model's posterior standard deviation.

Given that outliers are evenly distributed across the blocks, the Bayesian Merge and Reduce approach approximates the true linear regression models better than the Classical Merge and Reduce approach. Also, in the presence of unbalanced factor variables, the Bayesian Merge and Reduce models approximate the true models better than the Classical Merge and Reduce models.

Thus, both Classical and Bayesian Merge and Reduce approaches better approximate the true linear regression model provided there are enough observations per variable per block. Both approaches show a good approximation of the true linear models for a block size of 5000 observations.

Recommendations

All the linear regression models considered in this thesis employed non-informative prior distributions. Given that the sample size per block could be significantly lesser than the overall number of observations, it is imperative to ensure that the prior distribution for each block is made less informative. Future research can therefore seek to determine the amount of information that the prior distribution must contain and how this goal can be accomplished.

It is difficult to validate the assumptions and run diagnostics on the linear regression models considered in this thesis since each block of data is lost immediately after the appropriate model is generated. Hence, future studies can explore methods to examine the residuals without increasing working memory demand.

Future studies should focus on expanding the tractability of the Merge and Reduce statistical models to classifications and clustering. Since the Merge and Reduce method needs to have full access to the data set for the linear regression analysis, future studies can consider an application of the Merge and Reduce technique, in conjunction with some suitable goodness-of-fit measures.

The thesis focused on a situation where the original model cannot be fitted efficiently. The objective was to reduce the size of the data while achieving a similar model quality. Future studies can examine the performance of the techniques introduced in this study in the case where the big data set contains more variables than observations. Future studies could also examine the performance of the Merge and Reduce method on multiple response variables.

REFERENCES

- Agarwal, A., Zolotov, V., & Blaauw, D. T. (2004). Statistical clock skew analysis considering in trade-process variations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(8), 1231–1242. <https://doi.org/10.1109/TCAD.2004.831573>
- Aissi, S., Malu, P., & Srinivasan, K. (2002). E-business process modelling: the next big step. *Computer*, 35(5), 55–62. <https://doi.org/10.1109/MC.2002.999776>
- Anowar, F., Sadaoui, S., & Selim, B. (2021). Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). *Computer Science Review*, 40, 100378. <https://doi.org/10.1016/j.cosrev.2021.100378>
- Balakrishnan, S., & Madigan, D. (2006). A one-pass sequential Monte Carlo method for Bayesian analysis of massive datasets. *Bayesian Analysis*, 1(2). <https://doi.org/10.1214/06-BA112>
- Banerjee, A., Dunson, D. B., & Tokdar, S. T. (2013). Efficient Gaussian process regression for large datasets. *Biometrika*, 100(1), 75–89. <https://doi.org/10.1093/biomet/ass068>
- Baraniuk, R., Davenport, M., DeVore, R., & Wakin, M. (2008). A Simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3), 253–263. <https://doi.org/10.1007/s00365-007-9003-x>
- Bardenet, R., Doucet, A., & Holmes, C. (2014). Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach. *International Conference on Machine Learning*, 405–413. <http://proceedings.mlr.press/v32/bardenet14.pdf>

- Baykal, C., Liebenwein, L., Gilitschenski, I., Feldman, D., & Rus, D. (2018). Data-dependent coresets for compressing neural networks with applications to generalization bounds. *ArXiv*
- Begoli, E., & Horey, J. (2012). Design principles for effective knowledge discovery from Big Data. *2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture*, 215–218. <https://doi.org/10.1109/WICSA-ECSA.2012.32>
- Benson, A. R., Gleich, D. F., & Demmel, J. (2013). Direct QR factorizations for tall-and-skinny matrices in MapReduce architectures. *2013 IEEE International Conference on Big Data*, 264–272. <https://doi.org/10.1109/BigData.2013.6691583>
- Bolstad, W. M. (2009). *Understanding Computational Bayesian Statistics*. John Wiley & Sons
- Bruno, N., & Chaudhuri, S. (2007). Physical design refinement. *ACM Transactions on Database Systems*, 32(4), 28. <https://doi.org/10.1145/1292609.1292618>
- Chatterjee, S., & Hadi, A. S. (1988). Impact of simultaneous omission of a variable and an observation on a linear regression equation. *Computational Statistics & Data Analysis*, 6(2), 129–144. [https://doi.org/10.1016/0167-9473\(88\)90044-8](https://doi.org/10.1016/0167-9473(88)90044-8)
- Chen, L., & Zhou, Y. (2020). Quantile regression in big data: A divide and conquer based strategy. *Computational Statistics & Data Analysis*, 144, 106892. <https://doi.org/10.1016/j.csda.2019.106892>

- Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications*, 19(2), 171–209. <https://doi.org/10.1007/s11036-013-0489-0>
- Chen, X., & Xie, M. (2014). A split-and-conquer approach for analysis of extraordinarily large data. *Statistica Sinica*. <https://doi.org/10.5705/ss.2013.088>
- Clarkson, K. L., & Woodruff, D. P. (2009). Numerical linear algebra in the streaming model. *Proceedings of the 41st Annual ACM Symposium on Symposium on Theory of Computing - STOC '09*, 205. <https://doi.org/10.1145/1536414.1536445>
- Cohen, M. B., Elder, S., Musco, C., Musco, C., & Persu, M. (2015). Dimensionality reduction for k-means clustering and low-rank approximation. *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, 163–172. <https://doi.org/10.1145/2746539.2746569>
- Cormode, G., & Muthukrishnan, S. (2005). An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1), 58–75. <https://doi.org/10.1016/j.jalgor.2003.12.001>
- Csilléry, K., Blum, M. G. B., Gaggiotti, O. E., & François, O. (2010). Approximate bayesian computation (ABC) in practice. *Trends in Ecology & Evolution*, 25(7), 410–418. <https://doi.org/10.1016/j.tree.2010.04.001>
- Demmel, J., Grigori, L., Hoemmen, M., & Langou, J. (2012). Communication-optimal parallel and sequential QR and LU Factorizations. *SIAM Journal on Scientific Computing*, 34(1), A206–A239. <https://doi.org/10.1137/080731992>

- Feldman, D., Schmidt, M., & Sohler, C. (2020). Turning big data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering. *SIAM Journal on Computing*, 49(3), 601–657. <https://doi.org/10.1137/18M1209854>
- Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2), 137–144. <https://doi.org/10.1016/j.ijinfomgt.2014.10.007>
- Garlasu, D., Sandulescu, V., Halcu, I., Neculoiu, G., Grigoriu, O., Marinescu, M., & Marinescu, V. (2013). A big data implementation based on grid computing. *2013 11th RoEduNet International Conference*, 1–4. <https://doi.org/10.1109/RoEduNet.2013.6511732>
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian Data Analysis*. Chapman and Hall/CRC. <https://doi.org/10.1201/b16018>
- Geng, B., Li, Y., Tao, D., Wang, M., Zha, Z.-J., & Xu, C. (2012). Parallel lasso for large-scale video concept detection. *IEEE Transactions on Multimedia*, 14(1), 55–65. <https://doi.org/10.1109/TMM.2011.2174781>
- Geppert, L. N. (2018). *Bayesian and Frequentist Regression Approaches for Very Large Data Sets* fakultät Statistik
- Geppert, L. N., Ickstadt, K., Munteanu, A., Quedenfeld, J., & Sohler, C. (2017). Random projections for Bayesian regression. *Statistics and Computing*, 27(1), 79–101. <https://doi.org/10.1007/s11222-015-9608-z>

- Geppert, L. N., Ickstadt, K., Munteanu, A., & Sohler, C. (2020). Streaming statistical models via Merge and Reduce. *International Journal of Data Science and Analytics*, 10(4), 331–347. <https://doi.org/10.1007/s41060-020-00226-0>
- Givens, G. H., & Hoeting, J. A. (2013). *Computational Statistics* (2nd ed.). Wiley & Sons
- Guhaniyogi, R., & Dunson, D. B. (2015). Bayesian compressed regression. *Journal of the American Statistical Association*, 110(512), 1500–1514. <https://doi.org/10.1080/01621459.2014.969425>
- Har-Peled, S., & Mazumdar, S. (2004). On coresets for k-means and k-median clustering. *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing - STOC '04*, 291. <https://doi.org/10.1145/1007352.1007400>
- Hashem, I. A. T., Chang, V., Anuar, N. B., Adewole, K., Yaqoob, I., Gani, A., Ahmed, E., & Chiroma, H. (2016). The role of big data in a smart city. *International Journal of Information Management*, 36(5), 748–758. <https://doi.org/10.1016/j.ijinfomgt.2016.05.002>
- Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15, 1593–1623. <http://mcmc-jags.sourceforge.net>
- Huggins, J. H., Campbell, T., & Broderick, T. (2016). Coresets for scalable Bayesian logistic regression. *Advances in Neural Information Processing Systems*, 29
- Kerber, M., & Raghvendra, S. (2014). *Approximation and streaming algorithms for projective clustering via random projections*

Khan, N., Yaqoob, I., Hashem, I. A. T., Inayat, Z., Mahmoud Ali, W. K., Alam, M., Shiraz, M., & Gani, A. (2014). Big data: Survey, technologies, opportunities, and challenges. *The Scientific World Journal*, 2014, 1–18. <https://doi.org/10.1155/2014/712826>

Khare, A. (2014). Big data: Magnification beyond the relational database and data mining exigency of cloud computing. *2014 Conference on IT in Business, Industry and Government (CSIBIG)*, 1–6. <https://doi.org/10.1109/CSIBIG.2014.7056951>

Law, J., & Wilkinson, D. J. (2018). Composable models for online Bayesian analysis of streaming data. *Statistics and Computing*, 28(6), 1119–1137. <https://doi.org/10.1007/s11222-017-9783-1>

Lee, J. D., Liu, Q., Sun, Y., & Taylor, J. E. (2017). Communication-efficient Sparse Regression. *Journal of Machine Learning Research*, 18(1), 1–30. <https://www.jmlr.org/papers/volume18/16-002/16-002.pdf>

Li, R., Lin, D. K. J., & Li, B. (2012). Statistical inference in massive data sets. *Applied Stochastic Models in Business and Industry*, 29(5), 399–401. <https://doi.org/10.1002/asmb.1927>

Lin, N., & Xi, R. (2011). Aggregated estimating equation estimation. *Statistics and Its Interface*, 4(1), 73–83. <https://doi.org/10.4310/SII.2011.v4.n1.a8>

Ma, P., Mahoney, M. W., & Yu, B. (2014). A statistical perspective on algorithmic leveraging. *International Conference on Machine Learning*, 91–99. <http://proceedings.mlr.press/v32/ma14.pdf>

Martins, T. G., Simpson, D., Lindgren, F., & Rue, H. (2013). Bayesian computing with INLA: New features. *Computational Statistics & Data Analysis*, 67, 68–83. <https://doi.org/10.1016/j.csda.2013.04.014>

McCullagh, P., & Nelder, J. A. (2019). *Generalized Linear Models*. Routledge. <https://doi.org/10.1201/9780203753736>

Munteanu, A., Schwiiegelshohn, C., Sohler, C., & Woodruff, D. P. (2018). On coresets for logistic regression. *Advances in Neural Information Processing Systems*, 31

Naeem, M., Jamal, T., Diaz-Martinez, J., Butt, S. A., Montesano, N., Tariq, M. I., De-la-Hoz-Franco, E., & De-La-Hoz-Valdiris, E. (2022). Trends and future perspective challenges in Big Data. In *Advances in Intelligent Data Analysis and Applications*, 309–325. Springer. https://doi.org/10.1007/978-981-16-5036-9_30

Nelson, J., & Nguyen, H. L. (2013). OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 117–126. <https://doi.org/10.1109/FOCS.2013.21>

Odom, P. S., & Massey, M. J. (2003). *Tiered hashing for data access*. <https://patents.google.com/patent/US6516320B1/en>

Paul, S., Boutsidis, C., Magdon-Ismail, M., & Drineas, P. (2014). Random projections for linear support vector machines. *ACM Transactions on Knowledge Discovery from Data*, 8(4), 1–25. <https://doi.org/10.1145/2641760>

- Chen, P. C. L., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, 275, 314–347. <https://doi.org/10.1016/j.ins.2014.01.015>
- Zhou, Q., Shi, P., Liu, H. & Xu, S. (2012). Neural-network-based decentralized adaptive output-feedback control for large-scale stochastic nonlinear systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(6), 1608–1619. <https://doi.org/10.1109/TSMCB.2012.2196432>
- Quiroz, M., Tran, M.-N., Villani, M., & Kohn, R. (2018). Speeding up MCMC by delayed acceptance and data subsampling. *Journal of Computational and Graphical Statistics*, 27(1), 12–22. <https://doi.org/10.1080/10618600.2017.1307117>
- Raghupathi, W., & Raghupathi, V. (2014). Big data analytics in healthcare: promise and potential. *Health Information Science and Systems*, 2(1), 3. <https://doi.org/10.1186/2047-2501-2-3>
- Raskutti, G., & Mahoney, M. W. (2015). Statistical and algorithmic perspectives on randomized sketching for ordinary least-squares. *International Conference on Machine Learning*, 617–625. <http://proceedings.mlr.press/v37/raskutti15.pdf>
- Rodríguez-Mazahua, L., Rodríguez-Enríquez, C.-A., Sánchez-Cervantes, J. L., Cervantes, J., García-Alcaraz, J. L., & Alor-Hernández, G. (2016). A general perspective of Big Data: applications, tools, challenges and trends. *The Journal of Supercomputing*, 72(8), 3073–3113. <https://doi.org/10.1007/s11227-015-1501-1>

Rue, H., Martino, S., & Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2), 319–392. <https://doi.org/10.1111/j.1467-9868.2008.00700.x>

Sagiroglu, S., & Sinanc, D. (2013). Big data: A review. *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 42–47. <https://doi.org/10.1109/CTS.2013.6567202>

Sahimi, M., & Hamzhepour, H. (2010). Efficient computational strategies for solving global optimization problems. *Computing in Science & Engineering*, 12(4), 74–83. <https://doi.org/10.1109/MCSE.2010.85>

Sarlos, T. (2006). Improved approximation algorithms for large matrices via random projections. *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, 143–152. <https://doi.org/10.1109/FOCS.2006.37>

Siddiqa, A., Hashem, I. A. T., Yaqoob, I., Marjani, M., Shamshirband, S., Gani, A., & Nasaruddin, F. (2016). A survey of big data management: Taxonomy and state-of-the-art. *Journal of Network and Computer Applications*, 71, 151–166. <https://doi.org/10.1016/j.jnca.2016.04.008>

Sookhak, M., Talebian, H., Ahmed, E., Gani, A., & Khan, M. K. (2014). A review on remote data auditing in single cloud server: Taxonomy and open issues. *Journal of Network and Computer Applications*, 43, 121–141. <https://doi.org/10.1016/j.jnc.2014.04.011>

Steyerberg, E. W., Harrell, F. E., Borsboom, G. J. J. M., Eijkemans, M. J. C., Vergouwe, Y., & Habbema, J. D. F. (2001). Internal validation of predictive models. *Journal of Clinical Epidemiology*, *54*(8), 774–781. [https://doi.org/10.1016/S0895-4356\(01\)00341-9](https://doi.org/10.1016/S0895-4356(01)00341-9)

Thompson, D., Levine, J. A., Bennett, J. C., Bremer, P.-T., Gyulassy, A., Pascucci, V., & Pebay, P. P. (2011). Analysis of large-scale scalar data using pixels. *2011 IEEE Symposium on Large Data Analysis and Visualization*, 23–30. <https://doi.org/10.1109/LDAV.2011.6092313>

Tolochinsky, E., & Feldman, D. (2018). Generic corset for scalable learning of monotonic kernels: Logistic regression, sigmoid and more. *ArXiv*

Tracy, S. J. (2010). Qualitative quality: Eight “Big-Tent” criteria for excellent qualitative research. *Qualitative Inquiry*, *16*(10), 837–851. <https://doi.org/10.1177/1077800410383121>

Wang, L., Wang, G., & Alexander, C. A. (2015). Big data and visualization: methods, challenges and technology progress. *Digital Technologies*, *1*(1), 33–38. Retrieved from <https://scholar.google.com/>

Welling, M., Teh, Y. W., Andrieu, C., Kominiarczuk, J., Meeds, T., Shahbaba, B., & Vollmer, S. (2014). Bayesian inference with big data: a snapshot from a workshop. *ISBA Bulletin*, *21*(4), 8–11. Retrieved from <https://scholar.google.com/>

William, B. J., & Lindenstrauss, J. (1984). Extensions of Lipschitz mapping into Hilbert space. *Contemporary Mathematics*, *26*(189–206), 323–341. Retrieved from <https://scholar.google.com/>

- Li, X. & Yao, X. (2012). Cooperatively coevolving particle swarms for large-scale optimization. *IEEE Transactions on Evolutionary Computation*, 16(2), 210–224. <https://doi.org/10.1109/TEVC.2011.2112662>
- Yang, J., Meng, X., & Mahoney, M. W. (2016). Implementing randomized matrix algorithms in parallel and distributed environments. *Proceedings of the IEEE*, 104(1), 58–92. <https://doi.org/10.1109/JPROC.2015.2494219>
- Yang, Z., Tang, K., & Yao, X. (2008). Large-scale evolutionary optimization using cooperative coevolution. *Information Sciences*, 178(15), 2985–2999. <https://doi.org/10.1016/j.ins.2008.02.017>
- Liu, Y., Chen, C. L. P., Wen, G. & Tong, S. (2011). Adaptive neural output feedback tracking control for a class of uncertain discrete-time nonlinear systems. *IEEE Transactions on Neural Networks*, 22(7), 1162–1167. <https://doi.org/10.1109/TNN.2011.2146788>
- Yaqoob, I., Hashem, I. A. T., Gani, A., Mokhtar, S., Ahmed, E., Anuar, N. B., & Vasilakos, A. v. (2016). Big data: From beginning to future. *International Journal of Information Management*, 36(6), 1231–1247. <https://doi.org/10.1016/j.ijinfomgt.2016.07.009>
- Zhang, Y., Duchi, J., & Wainwright, M. (2015). Divide and conquer kernel ridge regression: A distributed algorithm with optimal minimax rates. *The Journal of Machine Learning Research*, 16(1), 3299–3340. <https://www.jmlr.org/papers/volume16/zhang15d/zhang15d.pdf>

Zhao, T., Cheng, G., & Liu, H. (2016). A partially linear framework for massive heterogeneous data. *The Annals of Statistics*, 44(4). <https://doi.org/10.1214/15-AOS1410>

Zhou, L., & Song, P. X.-K. (2017). *Scalable and efficient statistical inference with estimating functions in the MapReduce paradigm for big data.*



APPENDICES

APPENDIX A: Derivation of Marginal Posteriors of Parameters

The marginal posteriors for β and σ^2 can be obtained from the joint posterior as follows.

For β ,

$$P(\beta|Y) \propto P(Y|\beta, \sigma^2)P(\beta)$$

$$P(\beta|Y) \propto N(X\beta, \sigma^2 I_n)N(\mu_\beta, \Sigma_\beta)$$

$$P(\beta|Y) \propto e^{-\frac{1}{2\sigma^2}[(Y-X\beta)'(Y-X\beta)]} \times e^{-\frac{1}{2\Sigma_\beta}[(\beta-\mu_\beta)'(\beta-\mu_\beta)]}$$

$$P(\beta|Y) \propto e^{-\frac{1}{2\sigma^2\Sigma_\beta}[\Sigma_\beta(Y-X\beta)'(Y-X\beta) + \sigma^2(\beta-\mu_\beta)'(\beta-\mu_\beta)]}$$

$$P(\beta|Y) \propto e^{-\frac{(\Sigma_\beta X'X + \sigma^2 I_r)}{2\sigma^2\Sigma_\beta}[\beta' \beta - 2\beta C]}$$

where,

$$C = \frac{Y'\Sigma_\beta + \sigma^2\mu_\beta}{(\Sigma_\beta X'X + \sigma^2 I_r)}$$

Completing squares in β of the exponent,

$$P(\beta|Y) \propto e^{-\frac{1}{2\Sigma_\beta^*}[(\beta-\mu_\beta^*)'(\beta-\mu_\beta^*)]}$$

where

$$\Sigma_\beta^* = \frac{\sigma^2\Sigma_\beta}{\Sigma_\beta X'X + \sigma^2 I_r}$$

and

$$\mu_\beta^* = \frac{Y'\Sigma_\beta + \sigma^2\mu_\beta}{\Sigma_\beta X'X + \sigma^2 I_r}$$

Thus,

$$P(\beta|Y) = N(\mu_\beta^*, \Sigma_\beta^*)$$

For σ^2 ,

$$P(\sigma^2|Y) \propto P(Y|\beta, \sigma^2)P(\sigma^2)$$

$$P(\sigma^2|Y) \propto (\sigma^2)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma^2}[(Y-X\beta)'(Y-X\beta)]} \times (\sigma^2)^{-\alpha-1} e^{-\frac{\gamma}{\sigma^2}}$$

$$P(\sigma^2|Y) \propto (\sigma^2)^{-\left(\frac{n}{2}+\alpha\right)-1} \times e^{-\frac{1}{2}[(Y-X\beta)'(Y-X\beta)]+\gamma}$$

Thus, $P(\sigma^2|Y)$ is an inverse gamma distribution.

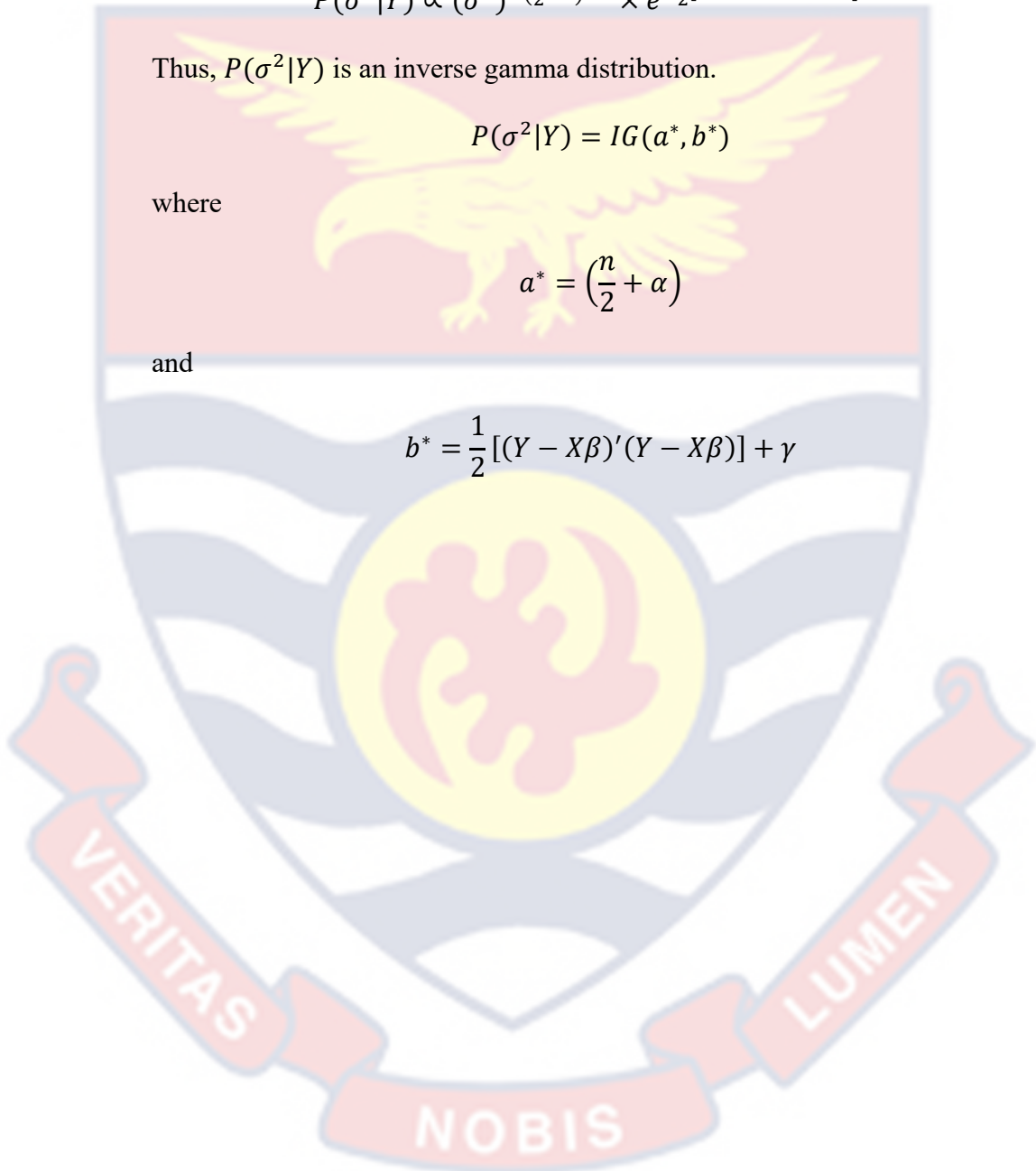
$$P(\sigma^2|Y) = IG(a^*, b^*)$$

where

$$a^* = \left(\frac{n}{2} + \alpha\right)$$

and

$$b^* = \frac{1}{2} [(Y - X\beta)'(Y - X\beta)] + \gamma$$



APPENDIX B: Running Times for Data Sets With 52 Variables

n	sktc	ϵ	Preprocessing Time			
			$\sigma = 1$	$\sigma = 2$	$\sigma = 5$	$\sigma = 10$
5×10^4	none		0.32	0.41	0.43	0.44
5×10^4	RAD	0.1	1.60	1.68	1.68	1.73
5×10^4	CW	0.1	0.01	0.01	0.01	0.01
5×10^4	RAD	0.2	0.40	0.39	0.42	0.43
5×10^4	CW	0.2	0.01	0.01	0.01	0.01
1×10^5	none		0.69	0.83	1.02	1.05
1×10^5	RAD	0.1	3.27	3.41	3.41	3.27
1×10^5	CW	0.1	0.02	0.03	0.02	0.02
1×10^5	RAD	0.2	0.76	0.84	0.84	0.80
1×10^5	CW	0.2	0.02	0.02	0.02	0.02
5×10^6	none		5.49	5.16	5.92	5.71
5×10^6	RAD	0.1	16.88	15.96	16.10	16.36
5×10^6	CW	0.1	0.09	0.09	0.09	0.10
5×10^6	RAD	0.2	3.73	4.00	4.03	3.85
5×10^6	CW	0.2	0.09	0.08	0.09	0.09
1×10^7	none		18.23	12.88	12.59	14.09
1×10^7	RAD	0.1	51.77	147.4	33.75	34.71
1×10^7	CW	0.1	0.19	0.27	0.38	0.46
1×10^7	RAD	0.2	7.92	8.46	8.38	8.21
1×10^7	CW	0.2	0.21	0.19	0.45	0.39

APPENDIX B CONT'D: Running Times for Data Sets With 52 Variables

n	sktc	ε	Analysis Time			
			$\sigma = 1$	$\sigma = 2$	$\sigma = 5$	$\sigma = 10$
5×10^4	none		1096	749.1	616.5	498.7
5×10^4	RAD	0.1	315.1	213.4	156.8	154.7
5×10^4	CW	0.1	375.2	293.9	164.6	171.8
5×10^4	RAD	0.2	23.17	26.00	17.48	21.81
5×10^4	CW	0.2	26.92	25.77	20.57	22.94
1×10^5	none				2036	1617
1×10^5	RAD	0.1	278.97	260.8	167.2	182.9
1×10^5	CW	0.1	257.56	278.2	187.0	198.8
1×10^5	RAD	0.2	21.44	23.21	17.52	23.65
1×10^5	CW	0.2	21.94	26.29	21.22	23.45
5×10^6	none					
5×10^6	RAD	0.1	279.8	313.3	165.9	198.4
5×10^6	CW	0.1	335.7	300.3	189.3	166.9
5×10^6	RAD	0.2	27.37	27.19	17.22	19.58
5×10^6	CW	0.2	26.03	25.23	24.39	22.86
1×10^7	none					
1×10^7	RAD	0.1	209.20	279.0	215.8	145.6
1×10^7	CW	0.1	281.7	232.4	175.5	144.0
1×10^7	RAD	0.2	21.27	19.93	22.87	23.43
1×10^7	CW	0.2	28.58	19.50	22.05	9.72