UNIVERSITY OF BAMENDA

FACULTY OF SCIENCE

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

POST-MODEL SELECTION AND MEASUREMENT ERROR IN HIGH-DIMENSIONAL DATA

A dissertation submitted to the Department Of Mathematics and Computer Science of the Faculty of Science, in the University of Bamenda in partial fulfillment of the requirements for the award of the Masters Degree (MSc.) in Probability and Statistics.

By

YIMTSOP ROLEX BRICE

UBa18SP045

(BSc. in Applied Mathematics)

Supervisor

PROF. NGUEFACK-TSAGUE GORGES

Co-Supervisor

Dr. KUM CLETUS KWA

DECEMBER, 2020

DECLARATION

I, **YIMTSOP ROLEX BRICE**, registration $N°$: **UBa18SP045**, Department of Mathematics and Computer Science in the Faculty of Science of the University of Bamenda hereby declare that, this dissertation titled: "**Post-Model Selection and Measurement Error in High-Dimensional Data**" is my original work. It has not been presented in any application for a degree or any academic pursuit. I have sincerely acknowledged all borrowed ideas nationally and internationally through citations.

Date:_____ Signature_____

CERTIFICATION

This is to certify that this research titled: "**Post-Model Selection and Measurement Error in High-Dimensional Data**" is the original work of **YIMTSOP ROLEX BRICE**. This work is submitted in partial fulfillment of the requirements for the award of a Master's degree (MSc.) in mathematics, Faculty of Science of the University of Bamenda, Cameroon.

_____

Prof. NGUEFACK-TSAGUE GEORGES

(Supervisor)

_____ _____

Dr. FELIX CHE SHU **Prof. TABOT CHARLES TABOT**

(Head of Department) (Dean)

Having met the stipulated requirements, the dissertation has been accepted by the Postgraduate School

Date: _____ _____

The General Coordinator

Postgraduate School

DEDICATION

This dissertation is lovingly dedicated to my beloved mother, **DJIMELI Philomene** and father, **TAGUELA François**. Their support, encouragement, and constant love have sustained me throughout my life.

ACKNOWLEDGEMENTS

This dissertation would not have been written without the help and encouragement of many people. I take this opportunity to highlight some of them.

First and foremost, I would like to express my gratitude to my supervisor, **Prof. NGUEFACK-TSAGUE GORGES**. His true researcher attitude and his careful reading of many manuscripts that I gave him, which is illustrated in being very receptive to new ideas, his passion for science, which results in being very knowledgeable and hardworking scientist, and his sense of responsibility, which results in easy reachability makes him an ideal supervisor and which I hope to continue to explore the joy of scientific research.

I express my sincere thanks to my co-supervisor **Dr. KUM Cletus Kwa** Head of Department of Mathematics of the higher teacher training college of the University of Bamenda for his encouragements and for the continuous support of my dissertation study.

My special appreciation is expressed to **Dr. SHU FELIX CHE** the head of the department of Mathematics and Computer Science,this work would not be possible without his patience, motivation, enthusiasm, and immense knowledge.

I also express my sincere thanks to **Dr. FOUTSA Emmanuel** senior lecturer of the University of Bamenda for all his advice in my live and his encouragements and orientation in my future mathematics field during this year.

To all the lecturers of Mathematics and Computer Science Department. I thank **Prof. MARTIN NDUMU, Prof. FONO LOUIS, Dr. PATRICE NDAMBOMVE, Prof. MAURICE NDIKONTAR, Dr. VUKENKENG ANDREW WUJUNG**, which through the relevance of their lesson offered a formation of quality to me.

I gratefully appreciate my beloved parents for their love, sacrifices and encouragement. I would like to express my special thanks to my friends for their suggestions and encouragement. Finally, I would like to thank every one, who supported me, but I did not mention above.

ABSTRACT

Much theoretical and applied work has been devoted to high-dimensional clean data. However, in many problems involving generalised linear models, we often face corrupted data in many applications where measurement errors cannot be ignored. It is thus necessary to extend regularisation methods, that can handle the situation where the number of covariates p is much larger than the sample size n, to the case in which covariates are also mismeasured. When the number of covariates p exceeds the sample size n, regularised methods like the Lasso or Dantzig selector are required. Several recent papers have studied methods which correct for measurement error in the Lasso or Dantzig selector for linear models in the $p > n$ setting, but many of them require knowledge about the measurement error structure or the covariance matrix of the measurement error. We study a correction for generalised linear models, based on Rosenbaum and Tsybakov's matrix uncertainty selector in which the measurement error and its covariance matrix are not known and have to be estimated from data and focus on variable selection in which the evaluation is based on simulations. In our simulation studies, we focus on linear, logistic and Poisson regression with measurement error. The proposed methods outperform the standard lasso with respect to covariate selection, by reducing the number of false positives. We also investigate a general procedure that utilizes the recently proposed Imputational-regularized Optimization algorithm for high-dimensional data with measurement error, which we implement for continuous, binary, and count response variable.

Keywords: generalised linear model, high-dimensional data, measurement error, matrix uncertainty selector

RÉSUMÉ

Beaucoup de travaux théoriques et appliqués on été consacrés aux données propres de grande dimension. Cependant, dans de nombreux problèmes impliquant des modèles linéaires généralisés, nous sommes souvent confrontés à des données corrompues dans de nombreuses applications où les erreurs de mesure ne peuvent être ignorées. il est donc nécessaire d'étendre les méthodes de régularisation, qui peuvent gérer la situation où le nombre de covariables p est beaucoup plus grand que la taille de l'échantillon n, au cas où les covariables sont également mal mesurées. Lorsque le nombre de covariables p dépasse la taille de l'échantillon n, des méthodes régularisées comme le Lasso ou le Dantzig selecteur sont requises. Plusieurs articles récents ont étudié des méthodes qui corrigent l'erreur de mesure dans le Lasso ou le Dantzig selecteur pour les modèles linéaires dans le réglage $p > n$, mais beaucoup d'entre eux nécessitaient des connaissances sur la structure d'erreur de mesure ou la matrix de covariance de la mesure d'erreur. Nous étudions une correction pour les modèles linéaires généralisés, basée sur le sélecteur d'incertitude matricielle de Rosenbaum et Tsybakov dans lequel l'erreur de mesure et sa matrice de covariance ne sont pas connues et doivent être estimées à partir de données et se concentrer sur les modèles de regression linéaire, logistique et Poisson avec erreur de mesure. La méthodes proposées surpassent le Lasso standard en ce qui concerne la sélection des covariables, en réduisant le nombre de faux positifs. Nous étudions également une procédure générale qui utilise l'algorithme d'optimisation régularisé par imputation récemment proposé pour les données de grande dimension avec erreur de mesure, que nous implémentons pour la variable de réponse continue, binaire et de comptage.

Mots clés: modèle lineaire généralisé, données de grandes dimensions, matrice de selection d'incertitude, erreur de mesure

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

HD : High-dimensional

i.i.d : Independent and Identically Distributed

AIC : AKAIC Information Criterion

BIC : Bayesian Information Criterion

DIC : Deviance Information Criterion

FIC : Focused Information Criterion

MUS : Matrix Uncertainty Selector

GMUL : Generalised Matrix Uncertainty Lasso

GLM : Generalised linear Model

GLMs : Generalised linear Models

GMUS : Generalised Matrix Uncertainty Selector

GDS : Generalised Dantzig Selector

DS : Dantzig Selector

Id : Identity

Lasso : Least Absolute Shrinkage Operator

HDME : High-Dimensional with Measurement Error.

L : Lasso

Chapter 1

GENERAL INTRODUCTION

1.1 Background and Motivation

Model selection is the task of selecting a statistical model from a set of candidate models, given data. More precisely, estimating the performance of different models in order to choose the (approximate) best one. Traditional subset selection methods combined with variable selection criteria, such as AIC, BIC, DIC, FIC are used to select a model among candidate models. These criteria may perform well in the relatively low dimensional space. However, these methods suffer from expensive computational cost and model instability for high-dimensional data (High-dimensional refers to the situations where the number covariates or predictors is much larger than the sample size). Recently, various regularisation methods through penalisation of parameters have been proposed for variable selection in high-dimensional regression analysis. Examples include the least absolute shrinkage and selection operator (LASSO) of Tibshirani (1996), the smoothly clipped absolute deviation (SCAD) (Zou and Li 2008), the least-angle regression (LARS) algorithm (Efron et al. 2004), the elastic net (Zou and Zhang 2009), the Dantzig selecor (DS) (Candes and Tao 2007), among others.

There are certain areas of classical statistics where model selection has played an important role, for example, linear regression and time series. In both sets of problems one asks essentially the same question: Which variables in a linear relation or linear predictor are worth keeping? This becomes a model selection problem if one identifies each set of retained variables with a model. It would be naive to expect the best results by including all the variables in one's model. One way of seeing this is to note that it violates the fundamental scientific principle of parsimony, which requires that of all the models that explain the data well, one should choose the simplest. Generalised linear models (GLMs) is another area in classical statistics where model selection is popular. In these applications it is used as an alternative to classical testing of hypothesis.

However, in practical applications, high-dimensional data analyses have to take into account measurement error in the covariates. It is thus necessary to extend regularisation methods, that can handle the situation where the number of covariates say, p is larger than the sample size say, n, to the case in which covariates are also mismeasured. In reality, the data in most applications are subject to at leat some measurement error. Some examples are, gene expression microarrays

subject to various sources of systematic and random error, which are noisy versions of the true gene expressions in the patients (Rocke and Durbin, 2001); food frequency questionnaires used in epidemiologic studies, in which subjects are asked about their food consumption and it is well known that people don't always tell the truth in survey (Kipnis t al., 2003); sensor network data also tend to be noisy, due to measurement error or sensor failure (Bertrand and Moonen, 2011).

In classical regression context, when $p < n$ and standard methods can be applied, it is well known that measurement error in the covariates will lead to bias in estimation of parameters and to loss of power (Carroll et al., 2006). Many correction methods are available to reduce this bias, but they generally need information about the measurement error structure. We refer to (belloni et al., 2014; Chen and Caramanis, 2013; Datta and Zou, 2015; Liang and Li, 2009; Loh and wainwright, 2012; Ma and Li, 2010; Rosenbaum and Tsybakov, 2010, 2013; Sorensen et al., 2015; Zhu et al., 2011). Typically, this require validation data or replicate measurements of the covariates in order to estimate the measurement error distribution. Most of these techniques require knowledge of the measurement error distribution or at least the covariance matrix of the measurement error, yielding estimators with good theoretical properties. However, in practice this will almost always be unknown, and estimation of the covariance matrix of the measurement error can be computationally expensive or even unfeasible when the number of variables p increases. Exceptions are the matrix uncertainty selector (MUS)(Rosenbaum and tsybakov, 2010), sparse total least squares (Zhu et al., 2011), and the modified orthogonal matching pursuit (Chen and Caramanis, 2013), that can account for measurement error without additional information about the measurement error structure, but sacrifice some statistical properties to get this robustness. The latter methods hence have a practical advantage in many applications, and in particular, they have been shown to yield fewer false positive selections than the standard Lasso and Dantzig selector. Kaul et al. (2016) proposed a two-stage non-penalised corrected least squares, which performs variable selection in the first step, and estimates the regression coefficient of the selected variables in the next step.

The methods proposed for dealing with measurement error in penalised regression all focus on linear regression, with the exception of Ma and Li (2010); Sorensen et al. (2015). Considering the importance and general applicability of GLMs, it is therefore of interest to develop penalised regression methods for GLMs which do not require an esimate of the measurement error distribution, and recover the linear regression framework as a special case. In this work, we investigate the generalised MUS (GMUS), based on a Taylor expansion of the GLM mean function around the true, but unknown, values of the covariates. The GMUS can be computed using

an iterative reweighing procedure, and when the Taylor expansion is truncated at first order, each step of this procedure requires solving a linear program. We also develop a generalised matrix uncertainty lasso (GMUL), which is a lasso type analog of the DS-based GMUS. The GMUL estimate can also be computed using an iterative reweighing procedure, in which an inner coordinate descent loop has to be run until a stopping criterion is met in each step of the algorithm. In simulation experiments with linear, logistic and Poisson regression, the GMUL and GMUS with the first order Taylor approximation are shown to give very promising covariate selection results compared to the lasso and the GDS, by detecting considerably fewer false positives (FPs) at similar numbers of true positives (TPs). We also investigate the extension to the Imputation-Regularized Optimization (IRO) algorithm for common types of generalised linear models in the presence of measurement error.

1.2 Objectives of work

The main goal of this work is to look into penalised regression for generalised linear models (GLMs) with measurement error and to compare some of these methods in a practical context, where the distribution and the variance of the measurement error is unknown and need to be estimated from data. The comparison will be based on simulations, and the focus will be on the linear regression model, logistic model and poisson models with additive measurement error.

1.3 Layout of work

The work is organised as follows. We start in Section 2 with a brief preliminary and notations. In Section 3, we give an overview of the models and methods that we will investigate and their theoretical properties. Section 4 describes the simulation setups with real data example and results, and Section 5 is devoted to the interpretation of the results and some final conclusions and recommendation

Chapter 2

Preliminaries

2.1 Generalised Linear Model

Most of the commonly used statistical distributions, e.g. Normal, Binomial, Gamma, Poisson and Inverse-Gaussian are members of the exponential family of distributions. The advantage of expressing diverse families of distributions in the common exponential form is that general properties of exponential families can then be applied to the individual cases.

Definition 2.1.1. *Let $y = (y_1, y_2, \ldots, y_n)^T$ be the response variable, $X = (x_1, x_2, \ldots, x_n)^T$ the covariate vector, and $\beta = (\beta_0, \ldots, \beta_p)^T$ be the vector of coefficients. The distribution of Y belongs to the exponential family of distributions if its density can be written in the form:*

$$f_Y(y, \theta, \phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right\}, \tag{2.1}$$

where $a(.)$, $b(.)$, and $c(.)$ are known functions that vary from one exponential family to another, ϕ is the dispersion parameter and θ is the vector of linear predictors given by

$$\theta_i = \mathbf{x}_i^T \beta = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}, \;\; i = 1, 2, \ldots, p, \tag{2.2}$$

A generalised linear model can be briefly characterised by the following three components

1. Independent random variables $y_1, y_2, \ldots, y_n$ with expected value $\mathbb{E}(y_i) = \mu_i$ and density function from the exponential family.

2. A linear predictor θ_i given by equation (2.2)

3. A link function, g, that describes how $\mathbb{E}(y_i) = \mu_i$ relates to θ_i

$$g(\mu_i) = \theta_i \tag{2.3}$$

Lemma 2.1. *The following equality can be verified using likelihood function of the exponential familly given in equation* (2.1)

$$\begin{cases} \mathbb{E}(Y) = & b'(\theta) = \mu(\theta) \\ var(Y) = & b''(\theta)\phi = V(\mu)\phi \end{cases} \tag{2.4}$$

Family	Link function $\theta_i = g(\mu_i)$	Mean function $\mu_i = g^{-1}(\theta_i)$
Gaussian	$Id(\mu_i) = \mu_i$	θ_i
Binomial	$Logit(\mu_i) = \log_e\left(\frac{\mu_i}{1-\mu_i}\right)$	$\frac{1}{1+e^{-\theta_i}}$
Poisson	$\log_e(\mu_i)$	e^{θ_i}
Gamma	μ_i^{-1}	θ_i^{-1}
Inverse-Gaussian	μ_i^{-2}	$\theta_i^{-1/2}$

Table 1: Link and mean functions for exponential family distributions

The exponential family distributions with their link and mean functions are given in table 1.

The regularized GLM aime to minimize, with respect to β, the objective:

$$Q(\beta; X, y, \lambda) = \mathcal{L}(y; x, \beta) + P(\beta; \lambda), \tag{2.5}$$

where $\mathcal{L}(y; x, \beta)$ is the negative log-likelihood function and $P(\beta; \lambda)$ is the penalty function on the coefficients. In clean HD data, the Lagrangian form of the Lasso (Tibshirani, 1996) estimates of β can be obtained by minimizing the negative log-likelihood of equation (2.1) in the following nonsmooth convex optimization problem.

$$\hat{\beta}_L = \underset{\beta}{argmin}\left[\frac{-1}{n}\sum_{i=1}^{n}\left\{y_i\theta_i - b\left(\theta_i\right)\right\} + \lambda||\beta||_1\right] \tag{2.6}$$

where $\theta_i = \mathbf{x}_i^T\beta$, $\lambda > 0$, is the regularisation parameter and $||\cdot||_p$ denotes the ℓ_p norm for vectors and matrices for $1 \leq p \leq \infty$. The GDS however solves:

$$\hat{\beta}_{DS} = \underset{\beta}{argmin}\left[||\beta||_1 : \max_{1\leq j\leq p}\left|\frac{1}{n}\sum_{i=1}^{n} x_{ij}\left\{y_i - \mu\left(\theta_i\right)\right\}\right| \leq \lambda\right] \tag{2.7}$$

2.1.1 Logistic Regression

In some regression situations, the response variable y has only two possible outcomes, for example, high blood pressure or low blood pressure, developing cancer of the esophagus or not developing it, whether a crime will be solved or not solved, document classification (presence versus absence). In such cases, the outcome y can be coded as 0 or 1 and we wish to predict the outcome (or the probability of the outcome) on the basis of one or more x's.

To illustrate a linear model in which y is binary, consider the model:

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \epsilon_i, \quad y_i = 1, 1; \;\; i = 1, 2, \ldots, n. \tag{2.8}$$

Since y_i is 0 or 1, the mean $\mathbb{E}(y_i)$ for each x_{ip} becomes the proportion of observations at x_{ip} for which $y_i = 1$. This can be expressed as

$$\begin{aligned} \mathbb{E}(y_i) &= \mathbb{P}(y_i = 1) = p_i, \\ 1 - \mathbb{E}(y_i) &= \mathbb{P}(y_i = 0) = 1 - p_i. \end{aligned} \tag{2.9}$$

The distribution $\mathbb{P}(y_i = 0) = 1 - p_i$ and $\mathbb{P}(y_i = 1) = p_i$ in (2.9) is known as the Bernoulli distribution. By (2.8) and (2.9), and the fact that $\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} = \mathbf{x}^T\beta$, with $\mathbf{x}_i = (1, x_i 1, \ldots, x_i p)$, $\beta = (\beta_0, \ldots, \beta_p)$, we have

$$\mathbb{E}(y_i) = p_i = \mathbf{x}_i^T \beta. \tag{2.10}$$

To obtain optimal estimators of β, we could use generalised least-squares estimators

$$\hat{\beta} = \left(\mathbf{X}^T \mathbf{V}^{-1} \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{V}^{-1} y \tag{2.11}$$

Since $\mathbb{E}(y_i) = p_i$ is a probability, it is limited by $0 \leq p_i \leq 1$. If we fit (2.10) by generalised least squares to obtain

$$\hat{p}_i = \mathbf{x}_i^T \hat{\beta} \tag{2.12}$$

then $\hat{p}_i$ may be less than 0 or greater than 1 for some values of $\mathbf{x}_i^T$. A model for $\mathbb{E}(y_i)$ that is bounded between 0 and 1 and reaches 0 and 1 asymptotically (instead of linearly) would be more suitable. A popular choice is the logistic regression model.

$$p_i = \mathbb{E}(y_i) = \frac{e^{\mathbf{x}_i^T \beta}}{1 + e^{\mathbf{x}_i^T \beta}}, \tag{2.13}$$

which can be linearlized by the simple transform

$$\ln\left(\frac{p_i}{1 - p_i}\right) = \mathbf{x}_i^T \beta. \tag{2.14}$$

Given the logistic model (2.14), the negative log-likelihood with Lasso regularization takes the form

$$- \frac{1}{n} \sum_{i=1}^{n} \left[y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \right] + \lambda ||\beta||_1 \tag{2.15}$$

$$= -\frac{1}{n} \sum_{i=1}^{n} \left[y_i \mathbf{x}_i^T \beta - \ln\left(1 + e^{\mathbf{x}_i^T \beta}\right) \right] + \lambda ||\beta||_1 \tag{2.16}$$

2.1.2 Poisson Regression

If the response y_i in a regression model is a count, the Poisson regression model may be useful. The Poisson probability distribution is given by

$$f(y) = \frac{\mu^y e^{-\mu}}{y!}, \quad y = 0, 1, 2, \ldots. \tag{2.17}$$

Family	$a(\phi)$	$b(\theta)$	$c(y,\phi)$
Gaussian	ϕ	$\theta^2/2$	$-1/2\left[y^2/\phi+\ln(2\pi\phi)\right]$
Binomial	$1/n$	$\log_e(1+e^\theta)$	$\log_e\binom{n}{ny}$
Poisson	1	e^θ	$-\log_e y!$
Gamma	ϕ	$-\log_e(-\theta)$	$-\phi^{-2}\log_e(y/\phi)-\log_e y-\log_e\Gamma(\phi^{-1})$
Inverse-Gaussian	ϕ	$-\sqrt{-2\theta}$	$-1/2\left[\log_e(\pi\phi y^3)+1/(\phi y)\right]$

Table 2: a(), b(.) and c(.) functions for exponential family distributions

The Poisson regression model is

$$y_i = \mathbb{E}(y_i) + \epsilon_i, \quad i = 1, 2, \ldots, n, \tag{2.18}$$

where the y_i's are independently distributed as Poisson random variables and $\mu_i = \mathbb{E}(y_i)$ is a function of $\mathbf{x}_i^T\beta = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}$.

Given the logistic model (2.17), the negative log-likelihood with Lasso regularization takes the form

$$-\frac{1}{n}\sum_{i=1}^{n}\left[y_i \ln\left(\mathbf{x}_i^T\beta\right) - \mathbf{x}_i^T\beta - \ln\left(y!\right)\right] + \lambda||\beta||_1 \tag{2.19}$$

2.1.3 Linear Regression

In the linear model

$$y_i = X\beta + \epsilon, \tag{2.20}$$

where the residual error ϵ is normally distributed with mean 0 and variance σ^2, i.e., $\epsilon \sim N(0, \sigma^2)$. then $y \sim N(\mu, \sigma^2)$ with $\mu = X\beta$. The lasso (Tbshirani, 1994) estimates the regression coefficient vector by

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{2n}||y - X\beta||_2^2 + \lambda||\beta||_1. \tag{2.21}$$

The distribution of y can be expression into the form of the exponential family as

$$f(y,\theta,\phi) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left\{-\frac{1}{\sigma^2}(y-\mu)\right\} = \exp\left\{\frac{y\theta-\theta^2/2}{\phi} - \frac{1}{2}\left[\frac{y^2}{\phi}+\ln(2\pi\phi)\right]\right\}, \tag{2.22}$$

where $\theta = \mu$, $\phi = \sigma^2$, $a(\phi) = \phi$, $b(\theta) = \theta^2/2$ and $c(y,\phi) = -1/2\left[y^2/\phi + \ln(2\pi\phi)\right]$.

Similarly, the Poisson, gamma, Binomial and inverse-Gaussian families can all be put into the form of exponential family using the results given in Table 2.

where $\Gamma(.)$ is the gamma function define as $\Gamma(x) = \int_0^\infty e^{-t}t^{x-1}dt$.

2.2 Convex Optimisation Problem

A mathematical optimization problem, has the form

$$\text{minimised } f(x) \text{ subject to } g_i(x) \leq b_i, \quad i = 1, \ldots, m, \tag{2.23}$$

were the vector $x = (x_1, \ldots, x_n)$ is the optimization variable of the problem, the function $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function, the functions $g_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, are the (inequality) constraint functions, and the constants $b_1, \ldots, b_m$ are the limits, or bounds, for the constraints. A vector x^* is called optimal, or a solution of the problem (2.23), if it has the smallest objective value among all vectors that satisfy the constraints: for any z with $g_1(z) \leq b_1, \ldots, g_m(z) \leq b_m$, we have $f(z) \geq f(x^*)$. The optimization problem (2.23) is called a linear program if the objective and constraint functions $f, g_1, \ldots, g_m$ are linear.

An important class of optimization problems involves convex constraint and convex objective functions. A convex optimization problem is one in which the objective and constraint functions are convex

2.2.1 Convex Optimality for Differentiable Problems

Definition 2.2.1. *A set $\mathcal{A} \subseteq \mathbb{R}$ is said to be convex if for all $\beta_1, \beta_2 \in \mathcal{A}$ and all scalars $t \in [0, 1]$, the vector $t\beta_1 + (1-t)\beta_2 \in \mathcal{A}$. A function $f : \mathbb{R}^p \to \mathbb{R}$ is is said to be convex if for any two vectors $\beta_1, \beta_2 \in \mathbb{R}^p$ and any scalar $t \in (0, 1)$, we have*

$$f(t\beta_1 + (1-t)\beta_2) \leq tf(\beta_1) + (1-t)f(\beta_2) \tag{2.24}$$

Consider the constrained optimization problem

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}} f(\beta) \quad \text{such that } \beta \in \mathcal{A}, \tag{2.25}$$

where $f : \mathbb{R}^p \to \mathbb{R}$ is a convex objective function to be minimised, and $\mathcal{A} \subset \mathbb{R}^p$ is a convex constraint set. When the objective function f is differentiable, then a necessary and sufficient condition for a vector $\beta^* \in \mathcal{A}$ to be a global optimum is that

$$\langle \bigtriangledown f(\beta^*), \beta - \beta^* \rangle \geq 0, \quad \text{for all } \beta \in \mathcal{A} \tag{2.26}$$

where $\bigtriangledown$ is the gradient. In particular, when $\mathcal{A} = \mathbb{R}^p$, the problem (2.25) is unconstrained and then the first order condition (2.26) reduces to the classical zero-gradient condition $\bigtriangledown f(\beta^*) = 0$.

Proposition 2.2.1. *For any convex function $g : \mathbb{R}^p \to \mathbb{R}$, the sublevel set $\{\beta \in \mathbb{R}^p | g(\beta) \leq 0\}$ is a convex set.*

The convex optimization problem

$$\underset{\beta\in\mathbb{R}^p}{\text{minimize}} f(\beta) \quad \text{such that } g_j(\beta) \leq 0 \text{ for } j = 1, 2, \ldots, m, \tag{2.27}$$

where $g_j, j = 1, 2, \ldots, m$ are convex functions that express constraints to be satisfied, is an instance of the general program (2.25). An important function associated with the problem (2.25) is the Lagrangian $L : \mathbb{R}^p \times \mathbb{R}^m_+ \to \mathbb{R}$, defined by

$$L(\beta; \lambda) = f(\beta) + \sum_{j=1}^{m} \lambda_j g_j(\beta). \tag{2.28}$$

The nonnegative weights $\lambda \geq 0$ are known as the Lagrange multipliers; the purpose of the multiplier λ_j is to impose a penalty whenever the constraint $g_j(\beta) \leq 0$ is violated. Indeed, if we allow the multipliers to be chosen optimally, then we recover the original program (2.27), since

$$\sup_{\lambda\geq 0} L(\beta; \lambda) = \begin{cases} f(\beta) & \text{if } g_j(\beta) \leq 0 \text{ for all } j = 1, 2, \ldots, m \\ +\infty & \text{otherwise,} \end{cases} \tag{2.29}$$

Then if f^* denote the optimal value of the optimization problem (2.27), we have

$$f^* = \inf_{\beta\in\mathbb{R}^p} \sup_{\lambda\geq 0} L(\beta; \lambda). \tag{2.30}$$

Definition 2.2.2. *(The KKT condition) The Karush-Kuhn-Tucker conditions relate the optimal Lagrange multiplier vector $\lambda^* \geq 0$, also known as the dual vector, to the optimal primal vector $\beta^* \in \mathbb{R}^p$:*

1. *Primal feasibility: $g_j(\beta^*) \leq 0$ for all $j = 1, 2, \ldots, m$.*
2. *Complementary slackness: $\lambda^*_j g_j(\beta^*) = 0$ for all $j = 1, 2, \ldots, m$.*
3. *Lagrange condition: the pair (β^*, λ^*) satisfies the condition*

$$0 = \bigtriangledown_\beta L(\beta^*; \lambda^*) = \bigtriangledown f(\beta^*) + \sum_{j=1}^{m} \lambda^*_j \bigtriangledown g_j(\beta^*). \tag{2.31}$$

The KKT conditions are necessary and sufficient condition for β^* to be a global optimum.

2.2.2 Convex Optimality for Nondifferentiable Function

In practice, many optimization problems arising in statistics involve convex but nondifferentiable objective functions.

Example 2.2.1. *The ℓ_1-norm $g(\beta) = ||\beta||_1 = \sum_{j=1}^{p} |\beta_j|$ is a convex function, but it fails to be differentiable at any point where at least one coordinate β_j is equal to zero. For such problems, the optimality conditions* (2.26) *and* (2.31) *are not directly applicable, since they involve gradients of the objective and constraint functions.*

Definition 2.2.3. *Let $f : \mathbb{R}^p \to \mathbb{R}$ be a convex function. a vector $z \in \mathbb{R}^p$ is said to be a subgradient of f at β_0 if*

$$f(\beta) \geq f(\beta_0) + \langle z, \beta - \beta_0 \rangle \quad \textit{for all} \;\; \beta \in \mathbb{R}^p. \tag{2.32}$$

The set of all subgradients of f at β_0 is called the sub-differential, denoted by $\partial f(\beta_0)$. At points of nondifferentiability, the subdifferential is a convex set containing all possible subgradients.

Example 2.2.2. *For the absolute value function $f(\beta) = |\beta|$, we have*

$$\partial f(\beta) = \begin{cases} +1 & if \;\; \beta > 0 \\ -1 & if \;\; \beta < 0 \\ [-1, +1] & if \;\; \beta = 0 \end{cases} \tag{2.33}$$

We frequently write $z \in sign(\beta)$ to mean that z belongs to sub-differential of the absolute value function at β. Recall the convex optimization problem (2.27), and assume that one or more of the functions $\{f, g_j\}$ are convex but nondifferentiable. In this case, the zero-gradient Lagrangian condition (2.31) no longer makes sense. But under mild conditions on the functions, the generalised KKT theory can still be applied using the modified condition

$$0 \in \partial f(\beta^*) + \sum_{j=1}^{m} \lambda_j^* \partial g_j(\beta^*), \tag{2.34}$$

In which we replace the gradients in the KKT condition (2.31) with the subdifferentials.

Example 2.2.3. *Suppose that we want to solve a minimisation problem of the form* (2.27) *with a convex and differentiable objective function f, and a single constraint specified by $g(\beta) = \sum_{j=1}^{p} |\beta_j| - R$ for some positive constant R. Thus, the constraint $g(\beta) \leq 0$ is equivalent to requiring that β belongs to an ℓ_1-ball of radius R. Recalling the form of the subdifferential* (2.33) *for the absolute value function, condition* (2.34) *becomes*

$$\bigtriangledown f(\beta^*) + \lambda^* z^* = 0, \tag{2.35}$$

where the subgradient vector satisfies $z_j^* \in sign(\beta_j^*)$ for each $j = 1, 2, \ldots, p$.

When the objective function f takes the form of the squared error $f(\beta) = \frac{1}{2n}||y - X\beta||_2^2$, this condition is equivalent to

$$-\frac{1}{n} \langle x_j, y - X\beta \rangle + \lambda s_j = 0, \quad j = 1, 2, \ldots, p. \tag{2.36}$$

where $s_j = sing(\beta_j)$ if $\beta_j \neq 0$ and some value lying in $[-1, 1]$ otherwise.

2.2.3 Some Inequalities

Definition 2.2.4. *(Triangle Inequality) for* $x, y \in \mathbb{C}$*,*

$$|x + y| \leq |x| + |y|. \tag{2.37}$$

More generally, for $x_i \in \mathbb{C}$, $i = 1, \ldots, n$, the generalised triangle inequality for finite sums is given by

$$\left| \sum_{i=1}^{n} x_i \right| \leq \sum_{i=1}^{n} |x_i|. \tag{2.38}$$

Definition 2.2.5. *(Hölder's Inequality) For* $x_i, y_i \in \mathbb{C}$*,* $i = 1, \ldots, n$*, if* $1 \leq p < \infty$ *and* $1/p + 1/q = 1$ *then the Hölder's Inequality for finite sums is given by*

$$\sum_{i=1}^{n} |x_i y_i| \leq \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} + \left(\sum_{i=1}^{n} |x_i|^q \right)^{1/q} \tag{2.39}$$

In the particular case where $p = q = 2$, the inequality (2.39) becomes the well known, Schwartz's inequality.

Chapter 3

Model Setup and Methods

3.1 Measurement error in covariates

In the additive measurement error model, the unobservable variable X is altered by adding random measurement error U, so what we observe is

$$w_i = x_i + u_i, \quad for \;\; i = 1, \ldots, n. \tag{3.1}$$

This can be re-formulated in matrix notation as:

$$W = X + U, \tag{3.2}$$

where U ia an $n \times p$ random noise matrix with covariance matrix Σ_U. If the covariates are measured with error, and we end up working with a contaminated covariate W, the naive Lasso estimate obtained by simply replacing X by W in (2.6) can be systematically biased (Rosenbaum and Tsybakov, 2010).

A key assumption is that the measurement error is standardized to have 0 mean and unit variance; i.e., for $j = 1, 2, \ldots, p$,

$$\frac{1}{n}\sum_{i=1}^{n} w_{ij} = 0, \quad \frac{1}{n}\sum_{i=1}^{n} w_{ij}^2 = 1. \tag{3.3}$$

Define the vector $\epsilon = (\epsilon_1, \ldots, \epsilon_n)^T$, and the generalised residual $\epsilon_i = y_i - \mu(\theta_i)$, $i = 1, \ldots, n$ Without measurement error, β is contained in the feasible set of the GDS if λ is chosen such that

$$\frac{1}{n}||X^T\epsilon||_\infty \leq \lambda, \tag{3.4}$$

where $||.||_\infty$ is the maximum component norm, holds (Antoniadis et al., 2010; Candes and Tao, 2007). Hence, for any GDS solution $\hat{\beta}_{DS}$ as well as the for the true regression coefficients β, the maximum correlation of any covariate with the residual is bounded by λ. This also means that under restricted eigenvalue conditions, $\hat{\beta}_{DS}$ and β are close, and that their maximun possible distance increases in λ. Hence, a natural starting point for a theoretical analysis is to assume that the bound (3.4) holds (B$\ddot{u}$hlmann and van de Geer, 2011, p. 103). Both of the previous selection methods assume the true covariates of the measurement to be observed. This is often not the case. When X is measured with error, the true coefficient vector β may not be part of the feasible set, even when λ is set to its theoretically optimal value in (3.4). The reason is that λ is a bound on

the residual of model, while in the case of measurement error, a bound on the measurement matrix U is also needed. The Matrix Uncertainty Selector (MUS) first introduced by Rosenbaum and Tsybakov (2010) as a modification of the Dantzig Selector for data with measurement error. In the special case of linear regression, Rosenbaum and Tsybakov (2010) developed the MUS by adding to the DS a new parameter δ that bounds the magnitude of the measurement error. They showed that when:

$$\frac{1}{n}||W^T\epsilon||_\infty \leq \lambda \text{ and } ||U||_\infty \leq \delta, \tag{3.5}$$

the vector of the true parameters β is a feasible solution of the MUS, defined as follows:

$$\hat{\beta}_{MUS} = \underset{\beta}{argmin} \left[||\beta||_1 : \frac{1}{n}||W^T(y - W\beta)||_\infty \leq \lambda + \delta||\beta||_1 \right]. \tag{3.6}$$

Note that the MUS does not require an estimate of the measurement error covariance matrix Σ_U. This might be a practical advantage in some cases, when an estimate of Σ_U is hard to obtain. It is worth noting that the estimation error bounds for the MUS do not go to zero when $n \to \infty$. This is the price to pay for not knowing the measurement error distribution. The choice of the unknown tuning parameters in the MUS is from a practical point of view not trivial. We chose to select the parameter λ through cross-validation. In particular we used the parameter selected by cross-validation of the naive lasso, i.e. the lasso of y on W. The choice of the parameter δ is critical. Low values of δ implies that the measurement error is not taken into account, while big values will overweight this effect. According to "elbow rule" (Rosenbaum and Tsybakov, 2010), the choice of the parameter value δ can be chosen where the curve starts to level off. This is not always easy in practice, but in the following plot, figure 3.1 a value between 0.05 and 0.1 may be reasonable.

3.2 Generalised Matrix Uncertainty Selector

The Generalised Matrix Uncertainty Selector (GMUS) is an extension of the MUS to generalised linear models, introduced by Sorensen et al. (2018).

Proposition 3.2.1. *GMUS is defined as the solution to the optimization problem:*

$$\hat{\beta}_{GMUS} = \underset{\beta}{argmin} \left[||\beta||_1 : \frac{1}{n}||W^T(y - \mu(W\beta))||_\infty \leq \lambda + \sum_{r=1}^{R} \frac{\delta^r}{r!\sqrt{n}}||\beta||_1^r||\mu^{(r)}(W\beta)||_2 \right] \tag{3.7}$$

where $\mu^{(r)}(.)$ is the rth derivative of the vector valued mean function $\mu(.)$ of the generalised linear model.

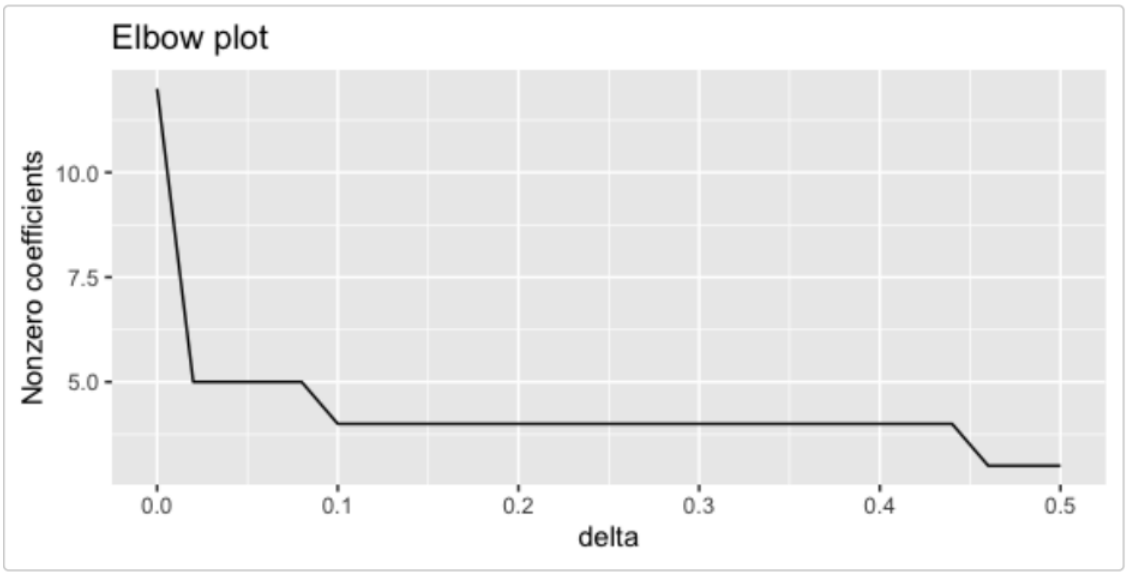

Figure 1: Elbov plot for HDME setting with $n = 50$, $p = 1000$

Proof. It follows from the Taylor series expansion of the mean function $\mu(\theta_i) = \mu(x_i^T\beta)$ around the scalar point $\mathbf{w}_i^T\boldsymbol{\beta}$ that

$$\mu\left(\mathbf{x}_i^T\boldsymbol{\beta}\right) = \sum_{r=0}^{\infty} \frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}\right)^r.$$

Which implies that

$$\mu\left(\mathbf{w}_i^T\boldsymbol{\beta}\right) = \mu\left(\mathbf{x}_i^T\boldsymbol{\beta}\right) - \sum_{r=1}^{\infty} \frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}\right)^r.$$

This gives, for $j = 1, \ldots, p$,

$$\begin{aligned}
\frac{1}{n}\left|\sum_{i=1}^{n} w_{ij}\left\{y_i - \mu\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)\right\}\right| &= \frac{1}{n}\left|\sum_{i=1}^{n} w_{ij}\left\{\epsilon_i + \sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}\right)^r\right\}\right| \\
&\leq \frac{1}{n}\left|\sum_{i=1}^{n} w_{ij}\epsilon_i\right| + \frac{1}{n}\left|\sum_{i=1}^{n} w_{ij}\sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}\right)^r\right| \\
&\leq \lambda + \frac{1}{n}\left|\sum_{i=1}^{n} w_{ij}\sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}\right)^r\right| \\
&\leq \lambda + \frac{1}{n}\sum_{i=1}^{n}\left|w_{ij}\sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}\right)^r\right|,
\end{aligned}$$

where we inserted $\epsilon_i = y_i - \mu\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)$ in the first step, we used the triangle inequality in the second step, we inserted the left bound in (3.5) in the third step, and finally used the generalised triangle

inequality. Next, we have

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}\left|w_{ij}\sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}\right)^r\right| \leq\ & \frac{1}{n}\left(\sum_{i=1}^{n}w_{ij}^2\right)^{\frac{1}{2}}\left[\sum_{i=1}^{n}\left\{\sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}\right)^r\right\}^2\right]^{\frac{1}{2}} \\ =\ & \frac{1}{\sqrt{n}}\left[\sum_{i=1}^{n}\left\{\sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}\right)^r\right\}^2\right]^{\frac{1}{2}},\end{aligned}$$

where we used Schwartz's inequality in the first step and the assumption (3.3) that the covariates are standardized to have mean zero and unit variance in the second step. We now note that the last term above is the ℓ_2-norm $\|\mathbf{v}\|_2$ of a vector $\mathbf{v}\in\mathbb{R}^n$ with elements

$$v_i=\sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}\right)^r,\quad i=1,\ldots,n.$$

We thus have

$$\begin{aligned}\frac{1}{\sqrt{n}}\left[\sum_{i=1}^{n}\left\{\sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}\right)^r\right\}^2\right]^{\frac{1}{2}} \leq\ & \frac{1}{\sqrt{n}}\sum_{r=1}^{\infty}\left[\sum_{i=1}^{n}\left\{\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)}{r!}\right\}^2\left(-\mathbf{u}_i^T\boldsymbol{\beta}\right)^{2r}\right]^{\frac{1}{2}} \\ \leq\ & \frac{1}{\sqrt{n}}\sum_{r=1}^{\infty}\left[\sum_{i=1}^{n}\left\{\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)}{r!}\right\}^2\|\mathbf{u}_i\|_\infty^{2r}\|\boldsymbol{\beta}\|_1^{2r}\right]^{\frac{1}{2}} \\ \leq\ & \sum_{r=1}^{\infty}\frac{\delta^r\|\boldsymbol{\beta}\|_1^r}{r!\sqrt{n}}\left[\sum_{i=1}^{n}\left\{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)\right\}^2\right]^{\frac{1}{2}} \\ =\ & \sum_{r=1}^{\infty}\frac{\delta^r\|\boldsymbol{\beta}\|_1^r}{r!\sqrt{n}}\left\|\boldsymbol{\mu}^{(r)}\left(\mathbf{W}\boldsymbol{\beta}\right)\right\|_2,\end{aligned}$$

where we used the triangle inequality in the first step, Hölder's inequality in the second step, and finally used the right bound in (3.5) in the second last step.

Putting the pieces together, it follows that

$$\frac{1}{n}\left|\sum_{i=1}^{n}w_{ij}\left\{y_i-\mu\left(\mathbf{w}_i^T\boldsymbol{\beta}\right)\right\}\right|\leq\lambda+\sum_{r=1}^{\infty}\frac{\delta^r}{r!\sqrt{n}}\|\boldsymbol{\beta}\|_1^r\left\|\boldsymbol{\mu}^{(r)}\left(\mathbf{W}\boldsymbol{\beta}\right)\right\|_2$$

for $j=1,\ldots,p$, which proves that the result. □

R is a parameter controlling the number of Taylor expansion terms which are included. When $R\to\infty$, the true solution is a member of the feasible set given that the bounds

$$\frac{1}{n}||W^T\epsilon||_\infty\leq\lambda\ \text{ and }\ ||U||_\infty\leq\delta, \tag{3.8}$$

hold. For computational reason in the rest of this work, we set $R=1$. When $R=1$, the GMUS can be solved using a sequence of linear programming problems on the same form as the MUS, with an iterative reweighing algorithm.

Algorithm 1 Iterative reweighing procedure for GMUS

input: initial estimate $\beta^{(0)}$, $k = 0$, $R \in \mathbb{N}$,

repeat

Compute z, $V^{(r)}$, $\tilde{W}$, $\tilde{z}$ and $\beta^{(k+1)}$, as given in equation (3.9), (3.10), (3.11) and (3.12).

$k \leftarrow k + 1$

until $||\beta^{(k)} - \beta^{(k+1)}|| < \epsilon_{tol}$

return $\hat{\beta}_{GMUS} = \beta^{(k)}$

Table 3: Iterative reweighing procedure for GMUS

3.2.1 Iterative Reweighing Algorithm for GMUS

We assume that the contaminated current estimate of the linear predictor for sample i after completing the kth iteration is $\theta_i^{W(k)} = w_i^T \beta^{(k)}$. We define the adjusted dependent covariate as

$$z_i = \theta_i^{W(k)} + \left[y_i - \mu\left(\theta_i^{W(k)}\right)\right]\left[\mu'\left(\theta_i^{W(k)}\right)\right]^{-1}, \quad i = 1, 2, \ldots, n. \tag{3.9}$$

We define a weight vector for each term in the Taylor expansion with R terms as

$$V^{(r)} = \left[\mu^{(r)}\left(\theta_1^{W(k)}, \ldots, \theta_n^{W(k)}\right)\right]^T. \tag{3.10}$$

We introduce the matrix $\tilde{W} \in \mathbb{R}^{n \times p}$ and the vector $\tilde{z} \in \mathbb{R}$ as

$$\tilde{w_{ij}} = w_{ij}\sqrt{V_i^{(1)}}, \quad \tilde{z_i} = z_i\sqrt{V_i^{(1)}}, \quad r = 1, 2, \ldots, R, \;\; i = 1, 2, \ldots, n, \;\; j = 1, 2, \ldots, p. \tag{3.11}$$

The next iterate $\beta^{(k+1)}$ is now given by

$$\hat{\beta}_{GMUS}^{(k+1)} = \underset{\beta}{argmin}\left[||\beta||_1 : \frac{1}{n}||\tilde{W}^T(\tilde{z} - \tilde{W}\beta)||_\infty \leq \lambda + \sum_{r=1}^{R}\frac{\delta^r}{r!\sqrt{n}}||\beta||_1^r||V^{(r)}||_2\right]. \tag{3.12}$$

The iterative reweighing procedure for GMUS is then define as

Theorem 3.2.1. *The solution obtained in algorithm* 1 *satisfied the inequality* (3.7) *in proposition* 3.2.1

Proof. Consider a fixed point of the Algorithm 1. At a fixed point, $\beta^{(k+1)} = \beta^{(k)}$. It follows that

$$z_i = \mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)} + \left[y_i - \mu\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}\right]\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}^{-1}, \; i = 1, \ldots, n$$

and

$$\mathbf{V}^{(r)} = \left[\mu^{(r)}\left\{\mathbf{w}_1^T\boldsymbol{\beta}^{(k+1)}\right\}, \ldots, \mu^{(r)}\left\{\mathbf{w}_n^T\boldsymbol{\beta}^{(k+1)}\right\}\right]^T,$$

and accordingly,

$$\tilde{w}_{ij} = \sqrt{\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}}w_{ij}, \quad \tilde{z}_i = \sqrt{\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}}z_i.$$

Inserting this into the constraint set in (3.12), we get for the left-hand side

$$\begin{aligned}
&\tilde{\mathbf{W}}^T\left(\tilde{\mathbf{z}} - \tilde{\mathbf{W}}\boldsymbol{\beta}^{(k+1)}\right) = \sum_{i=1}^{n}\tilde{w}_{ij}\left(\tilde{z}_i - \sum_{l=1}^{p}\tilde{w}_{il}\beta_l^{(k+1)}\right) = \\
&\sum_{i=1}^{n}\sqrt{\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}}w_{ij}\left(\sqrt{\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}}z_i - \sum_{l=1}^{p}\sqrt{\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}}w_{ij}\beta_l^{(k+1)}\right) = \\
&\sum_{i=1}^{n}\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}w_{ij}\left(z_i - \mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right) = \\
&\sum_{i=1}^{n}\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}w_{ij}\left(\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)} + \left[y_i - \mu\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}\right]\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}^{-1} - \mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right) = \\
&\sum_{i=1}^{n}w_{ij}\left(y_i - \mu\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}\right) = \\
&\mathbf{W}\left(\mathbf{y} - \boldsymbol{\mu}\left\{\mathbf{W}\boldsymbol{\beta}^{(k+1)}\right\}\right).
\end{aligned}$$

For the right-hand side of the inequality in (3.12), we get

$$\lambda + \sum_{r=1}^{R}\frac{\delta^r}{r!\sqrt{n}}\left\|\boldsymbol{\beta}^{(k+1)}\right\|_1^r\left\|\mathbf{V}^{(r)}\right\|_2 = \lambda + \sum_{r=1}^{R}\frac{\delta^r}{r!\sqrt{n}}\left\|\boldsymbol{\beta}^{(k+1)}\right\|_1^r\left\|\boldsymbol{\mu}^{(r)}\left\{\mathbf{W}\boldsymbol{\beta}^{(k+1)}\right\}\right\|_2.$$

It follows that at any fixed point of Algorithm 1,

$$\begin{aligned}
&\boldsymbol{\beta}^{(k+1)} = \\
&\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left\{\|\boldsymbol{\beta}\|_1 : \frac{1}{n}\left\|\tilde{\mathbf{W}}^T\left(\tilde{\mathbf{z}} - \tilde{\mathbf{W}}\boldsymbol{\beta}\right)\right\|_\infty \le \lambda + \sum_{r=1}^{R}\frac{\delta^r}{r!\sqrt{n}}\|\boldsymbol{\beta}\|_1^r\left\|\mathbf{V}^{(r)}\right\|_2\right\} = \\
&\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left\{\|\boldsymbol{\beta}\|_1 : \frac{1}{n}\left\|\mathbf{W}\left(\mathbf{y} - \boldsymbol{\mu}\left\{\mathbf{W}\boldsymbol{\beta}\right\}\right)\right\|_\infty \le \lambda + \sum_{r=1}^{R}\frac{\delta^r}{r!\sqrt{n}}\|\boldsymbol{\beta}\|_1^r\left\|\boldsymbol{\mu}^{(r)}\left\{\mathbf{W}\boldsymbol{\beta}\right\}\right\|_2\right\}.
\end{aligned}$$

Thus, any fixed point of Algorithm 1 is a solution to the original problem (3.7). □

The computation of solution (3.12) can be simplified by introducing the auxiliary variable $\mathbf{u} \in \mathbb{R}^p$ (see, e.g., Candes and Tao (2007, eq. (1.9)) or Boyd and Vandenberghe (2004, p. 617)) as follow

$$\begin{aligned}
&\text{mininimize } \mathbf{1}_p^T\mathbf{u} \text{ (with respect to } \mathbf{u}, \boldsymbol{\beta}) \text{ subject to } -\mathbf{u} \le \boldsymbol{\beta} \le \mathbf{u}, \\
&-\sum_{r=1}^{R}\frac{\delta^r}{r!\sqrt{n}}\left(\mathbf{1}_p^T\mathbf{u}\right)^r\left\|\mathbf{V}^{(r)}\right\|_2\mathbf{1}_p + \frac{1}{n}\tilde{\mathbf{W}}^T\tilde{\mathbf{W}}\boldsymbol{\beta} \le \lambda\mathbf{1}_p + \frac{1}{n}\tilde{\mathbf{W}}^T\tilde{\mathbf{z}}, \text{ and} \\
&-\sum_{r=1}^{R}\frac{\delta^r}{r!\sqrt{n}}\left(\mathbf{1}_p^T\mathbf{u}\right)^r\left\|\mathbf{V}^{(r)}\right\|_2\mathbf{1}_p - \frac{1}{n}\tilde{\mathbf{W}}^T\tilde{\mathbf{W}}\boldsymbol{\beta} \le \lambda\mathbf{1}_p - \frac{1}{n}\tilde{\mathbf{W}}^T\tilde{\mathbf{z}}.
\end{aligned}$$

When $R = 1$, (3.12) is thus equivalent to the linear program

$$\begin{aligned}
&\text{mininimize } \mathbf{1}_p^T\mathbf{u} \text{ (with respect to } \mathbf{u}, \boldsymbol{\beta}) \text{ subject to } -\mathbf{u} \leq \boldsymbol{\beta} \leq \mathbf{u},\\
&-\frac{\delta}{\sqrt{n}}\mathbf{1}_p^T\mathbf{u}\left\|\mathbf{V}^{(1)}\right\|_2\mathbf{1}_p + \frac{1}{n}\tilde{\mathbf{W}}^T\tilde{\mathbf{W}}\boldsymbol{\beta} \leq \lambda\mathbf{1}_p + \frac{1}{n}\tilde{\mathbf{W}}^T\tilde{\mathbf{z}}, \text{ and}\\
&-\frac{\delta}{\sqrt{n}}\mathbf{1}_p^T\mathbf{u}\left\|\mathbf{V}^{(1)}\right\|_2\mathbf{1}_p - \frac{1}{n}\tilde{\mathbf{W}}^T\tilde{\mathbf{W}}\boldsymbol{\beta} \leq \lambda\mathbf{1}_p - \frac{1}{n}\tilde{\mathbf{W}}^T\tilde{\mathbf{z}},
\end{aligned}$$

A convenient choice of $\beta^{(0)}$ is to take the GDS solution given in equation (2.7), corresponding to $\delta = 0$.

3.3 Generalised Matrix Uncertainty Lasso (GMUL)

The Matrix Uncertainty Lasso (MUL) defines in Rosenbaum and Tsybakov (2010) is given by

$$\hat{\beta}_{MUL} = \arg\min_{\beta}\left[\frac{1}{2n}||y - W\beta||_2^2 + \lambda||\beta||_1 + \frac{\delta}{2}||\beta||_1^2\right] \tag{3.13}$$

Proposition 3.3.1. *$\hat{\beta}_{MUL}$ are contained in the feasible set of the MUS, i.e.,*

$$\hat{\beta}_{MUL} \in \left\{\beta \in \mathbb{R}^p : \frac{1}{n}||W^T(y - W\beta)||_\infty \leq \lambda + \delta||\beta||_1\right\}. \tag{3.14}$$

We now define the GMUL as a vector $\hat{\beta}_{GMUL}$ which satisfies

$$\frac{-1}{n}\sum_{i=1}^{n} w_{ij}\left\{y_i - \mu(\theta_i^W)\right\} = \tau_j\left\{\lambda + \sum_{r=1}^{R}\frac{\delta^r}{r!\sqrt{n}}||\beta||_1^r||\mu^{(r)}(W\beta)||_2\right\}, \tag{3.15}$$

where $|\tau_j| \leq 1$ and $\tau_j 1_{(\beta_j \neq 0)} = sign(\beta_j)$ for $j = 1, 2, \ldots, p$ (Buhlmann and van de Geer, 2011). Which can be solved using an iterative reweighing procedure. From (3.9), (3.10), (3.11) and (3.12), and iterate $\beta^{(k+1)}$ given $\beta^{(k)}$, is a solution to

$$\frac{-1}{n}\tilde{w}_i^T\left(\tilde{z} - \tilde{W}\beta\right) = \tau_j\left\{\lambda + \sum_{r=1}^{R}\gamma_r(r+1)||\beta||_1^r\right\}, j = 1, 2, \ldots, p \tag{3.16}$$

where

$$\gamma_r = \frac{\delta^r}{(r+1)!\sqrt{n}}||V^{(r)}||_2, \quad r = 1, 2, \ldots, R \tag{3.17}$$

Hoever, (3.16) are the KKT conditions corresponding to the convex optimization problem

$$\text{minimize}\left\{\frac{1}{2n}||\tilde{z} - \tilde{W}\beta||_2^2 + \lambda||\beta||_1 + \sum_{r=1}^{R}\frac{\delta^r||V^{(r)}||_2}{(r+1)!\sqrt{n}}||\beta||_1^{r+1}\right\}. \tag{3.18}$$

Hence, we can find a vector $\hat{\beta}_{GMUL}$ satisfying (3.3) by solving (3.18) in each step of an iterative reweighing procedure. The convexity of (3.18) folows from the fact that the composition of the convex ℓ_1-norm, $||.||_1$ and the convex and nondecreasing power function $||.||_1^{r+1}$, is itself a convex

function (Boyd and Vandenberghe, 2004, p. 84). Thus the GMUL can be seen as a lasso analog of the GMUS.

Considering the $R = 1$ case, it turns out that we can solve a Lasso problem at each iteration, rather than the challenging problem (3.18). When $R = 1$, (3.18) is equivalent to minimize

$$\frac{-1}{n}\tilde{z}^T\tilde{\mathbf{W}}\beta + \beta^T\left(\frac{1}{2n}\tilde{\mathbf{W}}^T\tilde{\mathbf{W}} + \gamma_1\mathbf{I}_p\right)\beta + \lambda||\beta||_1 + \gamma_1\sum_{j=1}^{p}\sum_{l\neq j}|\beta_j||\beta_l|. \tag{3.19}$$

Replacing the last penalty term in this expression with a weighted ℓ_1-penalty, depending on the current estimate $\beta^{(k)}$, and will hence be updated at each iteration. i.e., we compute

$$\beta^{(k+1)} = \arg\min_{\beta}\left[\frac{-1}{n}\tilde{z}^T\tilde{\mathbf{W}}\beta + \beta^T\left(\frac{1}{2n}\tilde{\mathbf{W}}^T\tilde{\mathbf{W}} + \gamma_1\mathbf{I}_p\right)\beta + \sum_{j=1}^{p}\omega_j^{(k)}|\beta_j|\right], \tag{3.20}$$

where the weights are given by

$$\omega_j^{(k)} = \lambda + \gamma_1\sum_{l\neq j}|\beta_l^{(k)}|, \quad j = 1, 2, \ldots, p. \tag{3.21}$$

For $\gamma_1 \geq 0$, the Hessian $\frac{1}{2n}\tilde{\mathbf{W}}^T\tilde{\mathbf{W}} + \gamma_1\mathbf{I}_p$ is always positive semi-definite and then (3.20) is an ℓ_1-constrained convex optimization problem, which can be efficiently solved with a coordinate descent algorithm.

Proposition 3.3.2. *Using the coordinate descent algorithm, the solution of* (3.20) *take the form*

$$\beta_j \leftarrow \left(\frac{1}{n}\sum_{i=1}^{n}\tilde{w}_{ij}^2 + 2\gamma_1\right)^{-1} S\left\{\frac{1}{n}\sum_{i=1}^{n}\tilde{w}_{ij}|\left(\tilde{z}_i - \sum_{l\neq j}\tilde{w}_{il}\beta_l\right), w_j^{(k)}\right\}, \tag{3.22}$$

for $j = 1, 2, \ldots, p, 1, \ldots$ until the stopping criterion is met, where the soft-thresholding operator $S(.,.)$ is given by

$$S(a,b) = \begin{cases} a - b, & if \;\; a > 0 \;\; and \;\; b < |a| \\ a + b, & if \;\; a < 0 \;\; and \;\; b < |a| \\ 0, & if \;\; b \geq |a| \end{cases} \tag{3.23}$$

Proof. Our goal is to find a $\boldsymbol{\beta}$ minimizing the function

$$\begin{aligned} f(\boldsymbol{\beta}) = & \quad - - \frac{1}{n}\tilde{\mathbf{z}}^T\tilde{\mathbf{W}}\boldsymbol{\beta} + \boldsymbol{\beta}^T\left\{\frac{1}{2n}\tilde{\mathbf{W}}^T\tilde{\mathbf{W}} + \gamma_1\mathbf{I}_p\right\}\boldsymbol{\beta} + \sum_{j=1}^{p}\omega_j^{(k)}|\beta_j| \\ = & \quad -\frac{1}{n}\sum_{i=1}^{n}\tilde{z}_i\sum_{j=1}^{p}\tilde{w}_{ij}\beta_j + \frac{1}{2n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\tilde{w}_{ij}\beta_j\right)^2 + \gamma_1\sum_{j=1}^{p}\beta_j^2 + \sum_{j=1}^{p}\omega_j^{(k)}|\beta_j|. \end{aligned}$$

The partial derivatives of $f(\boldsymbol{\beta})$ with respect to β_j, $j = 1 \ldots, p$, can be written as

$$\frac{\partial f}{\partial\beta_j} = -\frac{1}{n}\sum_{i=1}^{n}\tilde{z}_i\tilde{w}_{ij} + \frac{1}{n}\sum_{i=1}^{n}\tilde{w}_{ij}\sum_{l\neq j}\tilde{w}_{il}\beta_l + \frac{1}{n}\beta_j\sum_{i=1}^{n}\tilde{w}_{ij}^2 + 2\gamma_1\beta_j + \omega_j^{(k)}\tau_j,$$

Algorithm 2 Iterative Reweighing and Coordinate Descent Algorithm for GMUL

imput: An initial estimate $\beta^{(0)}$ exists, $k = 1$.

repeat

Compute **z**, $\mathbf{V}^{(1)}$, $\tilde{\mathbf{W}}$, $\tilde{\mathbf{z}}$, γ_1 and $w_j^{(k)}$, as given in equation (3.9), (3.10), (3.11), (3.17) and (3.21) respectively

Let $l = 0$ and $\beta^{(k+1,0)} \leftarrow \beta^{(k)}$

repeat

for $j = 1, 2, \ldots, p$ **do**

Compute $\beta_j^{(k+1,l+1)}$ using (3.22)

end for $l \leftarrow l + 1$

until $||\beta^{(k+1,l+1)} - \beta^{(k+1,l)}|| < \epsilon_{tol}$

return $\beta^{(k+1)} = \beta^{(k+1,l+1)}$

$k \leftarrow k + 1$

until $||\beta^{(k)} - \beta^{(k-1)}|| < \epsilon_{tol}$

return $\hat{\beta}_{GMUL} = \beta^{(k)}$

where $\tau_j = 1$ if $\beta_j > 0$, $\tau_j = -1$ if $\beta_j < 0$, and $\tau_j \in [-1, 1]$ if $\beta_j = 0$. Setting $\partial f / \partial \beta_j = 0$, we find the analytical solution to (3.20),

$$\beta_j = \frac{n^{-1} \sum_{i=1}^{n} \tilde{w}_{ij} \left(\tilde{z}_i - \sum_{l \neq j} \tilde{w}_{ij} \beta_l \right) - \omega_j^{(k)} \tau_j}{n^{-1} \sum_{i=1}^{n} \tilde{w}_{ij}^2 + 2\gamma_1}$$

for $j = 1, \ldots, p$. Since $\boldsymbol{\tau}$ is implicitly defined, we compute $\boldsymbol{\beta}$ iteratively using the coordinate descent updates

$$\hat{\beta}_j \leftarrow \frac{S \left\{ n^{-1} \sum_{i=1}^{n} \tilde{w}_{ij} \left(\tilde{z}_i - \sum_{l \neq j} \tilde{w}_{il} \hat{\beta}_l \right), \omega_j^{(k)} \right\}}{\frac{1}{n} \sum_{i=1}^{n} \tilde{w}_{ij}^2 + 2\gamma_1},$$

for $j = 1, \ldots, p, 1, \ldots$ until convergence (Friedman, 2007). On convergence, we set $\boldsymbol{\beta}^{(k+1)} = \hat{\boldsymbol{\beta}}$. □

The iterative reweighing procedure for computing $\hat{\beta}_{GMUL}$ with $R = 1$, is then summarized in algorithm 2 as follows.

Chapter 4

Simulation study

The goal of this simulation study is to compare the GMUS with respect to the GDS and the GMUL with respect to the Lasso, in high-dimensional data setting where the measurement error and corresponding parameters are unknown and must be estimated from data. we will focus on variable selection, but we will also look at estimation error.

4.1 Simulations Design

We simulate data from the true model:

$$y = X\beta + \epsilon, \quad \epsilon \sim N(0, \sigma_\epsilon) \tag{4.1}$$

where the matrix X has i.i.d. entries $x_{ij} \sim N(0,1)$, and conditional on X.

The observed data were generated with two replicated measurements per observation

$$w_{ij} = x_{ij} + u_{ij} \tag{4.2}$$

The measurement error has been generated as

$$u_{ij} \sim N(0, \sigma_u), \text{ for } i = 1, \ldots, n \text{ and } j = 1, \ldots, p. \tag{4.3}$$

The response y_i was distributed with mean $\left(1 + \exp\left(-x_i^T\beta\right)\right)^{-1}$ for logistic regression, distributed with mean $\exp\left(x_i^T\beta\right)$ for Poisson regression and distributed with mean $\theta_i = x_i^T\beta$ for normal distribution. We investigate different choices of measurement error structures, different sizes of the active sets s, sample sizes n and covariates p. In particular, we considered:

- for the first choice of data parameter, we considered $n = 200$ and $p = 1000$,
- for the second choice of data parameter, we considered $n = 500$ and $p = 1500$,
- also the active set S of dimension s has been chosen with $s = 5$ and $s = 10$ and the nonzero regression coefficients were set to $\beta_j = 1$ for $j = 1, \ldots, s$,
- the measurement error standard deviation has been chosen to be $\sigma_u = 0.2$ and $\sigma_u = 0.5$.

For each problem instance with data (W, y), GDS and Lasso fits were computed via ten-fold cross-validation: one fit corresponding to the minimum cross-validated deviance, whose regularization parameter we denote $\hat{\lambda}_{min}$, and one fit corresponding to the largest regularization parameter we

denote $\hat{\lambda}_{1se}$. The solution to the GMUL was computed over a discrete grid of δ values, fixing λ at the $\hat{\lambda}_{min}$ or $\hat{\lambda}_{1se}$ obtained in the lasso fit. According to the *elbow rule* (Rosenbaum and Tsybakov, 2016), δ were chosen where the curve begins to flatten.

The simulations have been carried out using R version 3.6.2 on the HP PC, AMD E2-1800 APU with HD graphics 1.70 GHz and 4 GB RAM. For the estimation of the GDS and GMUS, we used the GLPK package (GNU Linear Programming Kit, https://www.gnu.org/software/glpk/) and the connected R package 'Rglpk' (Theussl and Hornik, 2013). The Lasso solution was computed using the R package 'glmnet' (Friedman et al., 2010)

4.2 Simulation Results

We will describe the results dividing the simulations into three subsections, Poisson, logistic and Normal.

4.2.1 Poisson Regression

4.2.2 Logistic Regression

4.2.3 Normal regression

Chapter 5

Discussion, Conclusion and Recommendation

BIBLIOGRAPHY