Regularized Relative Risk Regression: A Non-GLM Approach with Emphasis on Large p, Small N Simulations

by

Xinyuan (Chloe) You

B.Sc., University of British Columbia, 2020

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Statistics)

The University of British Columbia
(Vancouver)

August 2023

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Regularized Relative Risk Regression: A Non-GLM Approach with Emphasis on Large p, Small N Simulations

submitted by **Xinyuan (Chloe) You** in partial fulfillment of the requirements for the degree of **Master of Science** in **Statistics**.

Examining Committee:

Gabriela V. Cohen Freue, Associate Professor, Statistics, UBC
Supervisor

Paul Gustafson, Professor, Statistics, UBC
Supervisory Committee Member

Abstract

In clinical research, the determination of the association's strength between two events is paramount. This may involve probing the relationship between a risk factor and a health outcome, or evaluating the link between a treatment and its efficacy. The Odds Ratios (OR) and Relative Risks (RR) stand out as the predominant measures for such evaluations. While logistic regression is commonly employed for OR modeling, and Poisson regression for RR, each has its set of limitations in practical applications.

In light of these limitations, Richardson et al. (2017) introduced a novel non-GLM binary regression approach for direct RR estimation using a log odds-product nuisance model. This technique elegantly sidesteps the intertwined dependence of RR on baseline risk. However, this method encountered challenges in high-dimensional and sparse model estimation ($p > N$). To address these issues, this study introduces a novel estimator founded on the binary regression model, which is further refined with an algorithm using Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) to solve the optimization problem. This algorithm encourages sparsity in the solution and enables variable selection, thereby improving the utility for high-dimensional and sparse models. This thesis examines the properties of the estimator through simulation studies and discusses the potential for future enhancements and applications. The presented work represents a step forward in creating alternative methodologies for estimating relative risks in diverse data landscapes.

Lay Summary

Health data often demands risk comparisons between scenarios or populations, like assessing lung cancer risk between smokers and non-smokers or heart disease risk based on gene variants. Two standard tools for these comparisons are odds ratios (OR) and relative risks (RR). While RR is more intuitive, showing how many times more likely an event is in one group than another, its calculation can be intricate, especially with limited data or numerous variables like age, diet, and genetic markers. Hence, the simpler OR is used, though it can occasionally overestimate risks, especially for common events. This thesis introduces a novel method, merging traditional research challenges with advanced computational techniques. This synergy allows more accurate RR estimations, even with abundant data and multiple variables. Combining classic and contemporary statistical methods, this research offers a vital tool for health professionals, promising improved health guidelines and informed public health decisions.

Preface

The thesis is original, unpublished work by the author, Xinyuan You, under the supervision of Professor Gabriela V. Cohen Freue, and in collaboration with Professor Linbo Wang from the University of Toronto. All implementation, simulations and analyses were contributed by the author. All simulations were run in clusters provided by the Digital Research Alliance of Canada. Implementation of the functions were done in R, and can be found in https://github.com/ChloeYou/regularized-RR-regression.

Table of Contents

List of Tables

List of Figures

Acknowledgments

Firstly, I wish to express my heartfelt appreciation to my supervisor, Professor Gabriela V. Cohen Freue. Her support, and encouragement were instrumental throughout my graduate studies. Her mentorship fostered an environment of exploration and discovery, truly enriching my master's journey. I also extend my gratitude to Professor Paul Gustafson, my second reader. I owe particular gratitude to Professor Daniel J. McDonald, who generously provided me the opportunity to contribute to the Delphi project. I also wish to acknowledge Professor Linbo Wang for sharing his innovative ideas generously.

To Jana and Maggie, your constant support was a beacon during this academic pursuit. Our late-night group projects, Whistler ski trips, and hangouts have become fond memories that I will cherish always. The camaraderie and support from friends and colleagues from the department were invaluable and enriched my experience.

My heartfelt thanks go to David, who has been a constant source of support, be it during academic pursuits, professional endeavors, or personal challenges. His unwavering companionship amidst my busiest days, along with his culinary finesse, always brought a welcomed comfort to my days.

Lastly, my deepest gratitude goes to my family. My mother has always served as my role model, consistently guiding me. I am forever indebted to her for her sacrifices and encouragement. To my father, for always telling me to become a better version of myself, and his unconditional love. To my sister and best friend, Helen, who always provides emotional support and care. This accomplishment would not have been possible without you all.

Chapter 1

Introduction

In clinical research, it is often crucial to determine the strength of association between two events. This could involve assessing the relationship between a risk factor and a health outcome, or between a treatment and its effectiveness. Two common measures used to evaluate this association are the Odds Ratios (OR) and Relative Risks (RR).

The OR is a measure of effect size, describing the strength of association or non-independence between two binary data values. On the other hand, the RR is a ratio of the probability of an event occurring in an exposed group versus a non-exposed group. It is an indicative statistic used to compare the risk of a particular event happening in two different groups. To estimate these measures, Generalized Linear Models (GLMs) are often utilized. Specifically, logistic regression models are widely used for OR estimation, while Poisson regression models are typically employed for RR estimation. These regression models allow us to estimate the relationship between the covariates and the response.

While logistic regression offers computational ease and an OR estimate, it falls short in depicting the RR dependence on the covariates. This limitation prevents the direct discernment of the functional form of relative risk or identification of key baseline variables impacting it. Additionally, directly estimating relative risk given covariates can be achieved through log-binomial regressions. However, log-binomial regressions are infamous for numerical instability due to its constrained parameter space. While some challenges are addressed by Donoghoe

and Marschner [3], issues persist when estimates lie on the parameter space boundary. As an alternative, log-Poisson regression is often employed, shown as a robust approximation to log-binomial regression by Zou [22] and Lumley et al. [9]. It estimates asymptotic variance using a sandwich estimator [22], with a key limitation that predicted probabilities may exceed the [0,1] range. Moreover, it is variation-dependent, tying the relative risk to the baseline risk.

Richardson et al. [14] suggested a non-GLM model focusing on the impact of a binary exposure and its interaction with other covariates while addressing the problems of meaningless predictions, and variation dependence between relative risk and the baseline risk. This approach tackles the requirement of having a collapsible measure of association and a model with a less constrained parameter space. The parameters from this proposed model can be estimated using maximum likelihood, and unlike log-Poisson regression, the predicted probabilities are always within the [0, 1] interval as the model leads to a genuine likelihood.

While the model proposed by Richardson et al. [14] resolves many challenges associated with relative risk estimation, it comes with its limitations. These include potential significant bias in parameter estimation with small sample sizes or a large number of exploratory variables, the possibility of infinite maximum likelihood estimates under certain parameter combinations, and a non-convex objective function complicating the global maximum search.

In this thesis, we mitigate the first limitation of our model by developing a penalized likelihood approach, thereby introducing sparsity into the estimated coefficients. This method, though, is known to result in biased coefficients - the coefficients derived from our approach may not precisely represent the true magnitude of the relationship between the covariates and the outcome. Instead, they may depict a shrunken version of that relationship, a characteristic of such penalized methods. However, the limitation of possible infinite maximum likelihood estimates and non-convexity of the objective function persists and are discussed in section 5.

This thesis is structured into five distinct chapters. In Chapter 2, we review existing literature, covering topics related to relative risk estimation. Additionally, we focus on understanding the intricacies of the Least Absolute Shrinkage and Selection Operator (Lasso) method that we rely on to achieve sparsity. This chap-

ter also encompasses an exploration of the optimization techniques applicable in this context, providing a robust groundwork for the following chapters. Chapter 3 marks the introduction of our primary contribution - the ℓ_1-regularized relative risk regression. Our simulation studies are laid out in Chapter 4. In this section, we list the scenarios we considered, the data generation process, the metrics used to evaluate performance. Following this, we present and discuss the results, comparing our proposed method's performance with existing techniques. Finally, in Chapter 5, we conclude the thesis and discuss potential directions for future research.

Chapter 2

Background

In this chapter, we begin with the motivation of relative risk regressions, and outline the common choices for modeling relative risk. We introduce the binary regression model proposed by Richardson et al. [14] on which our work is built upon. We move on to review LASSO and related techniques that encourage sparsity in the estimated coefficients and can be used for variable selection. We end the chapter by discussing different optimization methods that are either derivative-free optimization algorithms or variants of first-order methods such as accelerated gradient descent (FISTA). Finally, we discuss accelerated proximal gradient methods for non-convex programming.

2.1 Binary Regression Models on Relative Risk Estimation

The logistic regression is commonly used to model the association between the binary outcome and covariates. The measure of association that comes from the logistic regression is odds ratio (OR). The regression coefficients are often estimated via maximum likelihood estimation (MLE).

However, relative risks (RR) as a measure of association has gained popularity in medical and public health research for its interpretability and rigorous estimation in studies of frequent outcomes [10]. Although odds ratios approximate the relative risk for rare outcomes, they can differ when the outcomes are prevalent.

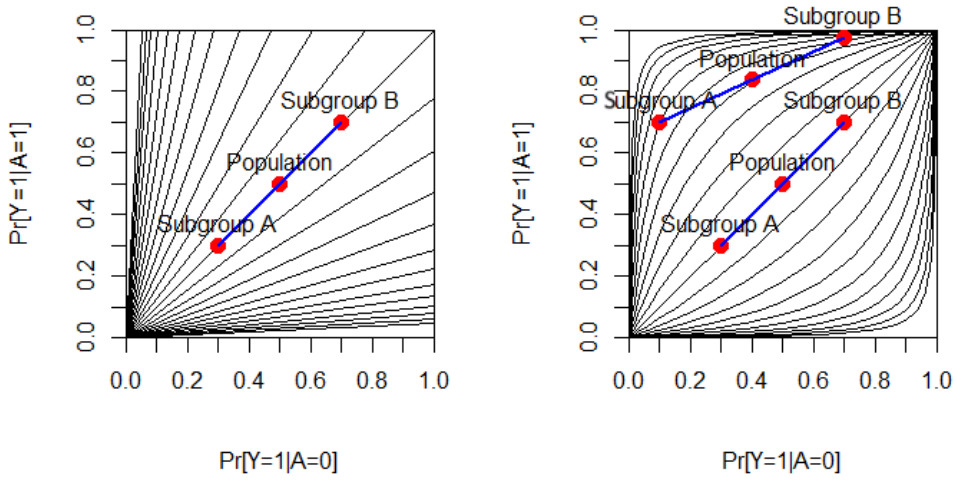

Figure 2.1: Illustration of Collapsibility of Relative Risk. (Left) Lines of constant log relative risk. (Right) Curves of constant log odds ratio.

Particularly, the estimated odds ratio overestimates the relative risk in these studies with frequent outcomes. Furthermore, relative risks are collapsible, such that the marginal relative risk lies in the convex hull of stratum-specific relative risks. In other words, a weighted average of stratum-specific RR equals the ratio of the pooled relative risk. On the contrary, OR is non-collapsible, which makes it hard to make valid comparisons of logistic regression coefficients from different studies [16]. This effect can be shown by L'Abbé plots in Figure 2.1. The left plot in Figure 2.1 shows that for two subgroups with the same relative risk, the population relative risk is a simple average of the two, which is intuitive. The right plot in Figure 2.1 shows that for two subgroups with the same odds ratio, when the odds are low, the odds are approximately collapsible. However, when odds are high, two subgroups with identical odds ratios can result in varying population odds ratios. This non-collapsibility can lead to less intuitive inferences based on population odds. The discrepancy between stratum-specific odds ratios and the population odds ratio is sometimes termed as Simpson's paradox [5].

Therefore, other methods are considered to estimate relative risk directly. Common choices are the log-binomial model and the Poisson regression with robust standard errors [22], which both fall in the generalized linear model framework.

However, there are some drawbacks to the two models. The parameter estimations via MLE for log-binomial model fail to converge in certain settings on standard statistical software [19], such as `glm()` in R. Additionally, if the estimated log relative risk in the log-binomial model can take any value from $-\infty$ to ∞, then the model can produce predicted values that are greater than 1. Alternatively, the `logbin()` function in the `logbin` R package [3] may converge even in cases where `glm()` fails, but it does not provide the option of adding regularization when we have a high-dimensional setting. One may reach for `glmnet` in R in hope of a regularized log-binomial model, but it does not natively support the choice of modeling binomial data with the logarithm of the probability as the link function. A common workaround is to approximate the log-binomial model using a Poisson regression with robust standard errors.

The Poisson regression is one of the native models supported by `glmnet` and therefore allows for regularization applied to the maximum likelihood functions. However, the Poisson regression also poses some difficulties in prediction and computation. Firstly, the predicted probability of new observations could be outside of the range of [0,1]. Next, we define X as a binary treatment, Y as a binary outcome, and V as a vector of covariates, which is the notation we use consistently throughout the thesis. Difficulties arise from the fact that the relative risk is variation dependent on the baseline probability $\mathrm{P}(Y = 1|X = 0)$. For example, for some set of variables v, we have $RR(v) = \frac{P(Y=1|X=1,v)}{P(Y=1|X=0,v)} = 2$, then it follows that $\mathrm{P}(Y = 1|X = 0, v) \leq 0.5$, since $\mathrm{P}(Y = 1|X = 1, v)$ must be less than or equal to 1. This suggests that there is a restricted domain over which the values of the relative risk and the baseline probability are able to provide a valid probability distribution and is shown in Figure 2.2.

Richardson et al. [14] proposed a binary regression model that uses a non-GLM approach to solve the issue of having a measure of association that is collapsible. Moreover, the parameters from the proposed model can be estimated via maximum likelihood.

The proposed model consists of two parts– a model of interest for the log relative risk and a nuisance model for the log odds product. Let $W = \omega(V), Z = z(V)$

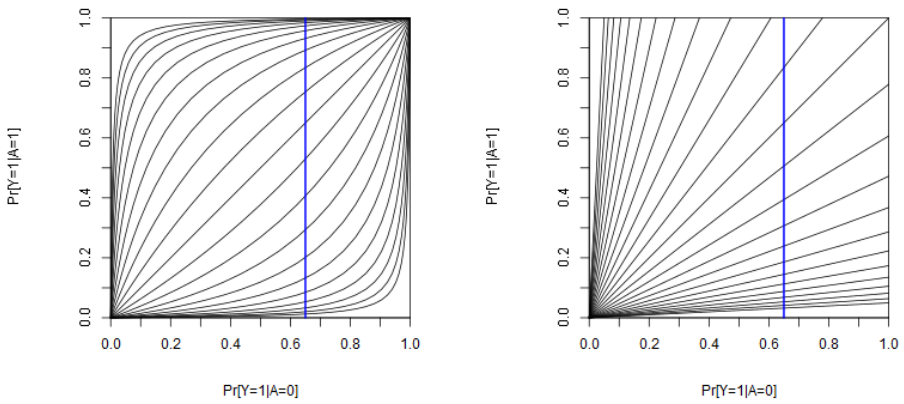

Figure 2.2: Illustration of Variation Dependence of Relative Risk on the Baseline Risk. (Left) Curves of constant log odds ratio. (Right) Lines of constant log relative risk. The vertical lines indicate the baseline risk of 0.65 (or a baseline odds of 1.86). [14]

be known vector functions of V. The proposed model is expressed as follows:

$$\log \mathrm{RR}(V) = \alpha^{\mathrm{T}} W, \tag{2.1}$$

$$\log \mathrm{OP}(V) = \beta^{\mathrm{T}} Z, \tag{2.2}$$

where the relative risk(RR) and odds product(OP) are expressed as:

$$\mathrm{RR}(v) = \frac{P(Y=1|X=1,V=v)}{P(Y=1|X=0,V=v)},$$

$$\mathrm{OP}(v) = \frac{P(Y=1|X=0,V=v)P(Y=1|X=1,V=v)}{(1-P(Y=1|X=0,V=v))(1-P(Y=1|X=1,V=v))}.$$

When visualized via the L'Abbé plot in Figure 2.3 compared to Figure 2.2, the log odds product is variation independent of the relative risk, while still having a collapsible parameter of interest.

Next we lay out the steps to identify and estimate the parameters α, β. For simplicity, we define $\theta(V) = \alpha^T W$, $\phi(V) = \beta^T Z$, $p_0(v) = P(Y=1|X=0,V=v)$,

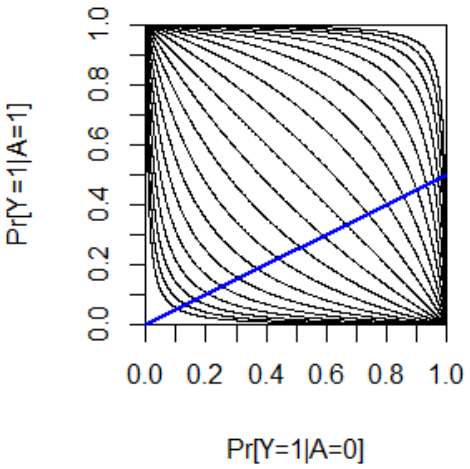

Figure 2.3: Lines of constant log odds product. RR (blue line) and OP are variation independent.

and $p_1(v) = P(Y = 1|X = 1, V = v)$. Solving a quadratic function after taking the logarithm on both sides of 2.1 and 2.2, we obtain the closed form expression of $p_0(v)$ and $p_1(v)$ presented below which is expressed in terms of α, β, and is used in the expression of the maximum likelihood function.

$$p_0(v) = \frac{-\left(e^{\theta(v)}+1\right)e^{\phi(v)} + \sqrt{e^{2\phi(v)}\left(e^{\theta(v)}+1\right)^2 + 4e^{\theta(v)+\phi(v)}\left(1-e^{\phi(v)}\right)}}{2e^{\theta(v)}\left(1-e^{\phi(v)}\right)}, \quad (2.3)$$

$$p_1(v) = p_0(v)e^{\theta(v)}. \quad (2.4)$$

It is important to note that when the log odds-product is 0 the expression 2.3 is not defined. Therefore, finding the limit of the expression as the log odds-product approaches 0 is necessary. In the R package `brm` [14], the authors incorporated the limiting case $p_0(v) = \frac{1}{1+\exp\theta(v)}$, which is applied when the log odds-product approaches zero. Furthermore, the representation of very small or very large probabilities can lead to numerical instability and precision issues due to the limitations of finite precision arithmetic, particularly with the frequent use of logarithmic transformations of probabilities in the authors proposed method. Therefore, other constraints exist in their implementation to ensure that when the log relative risk

and log odds product are very large or very small, the corresponding probabilities $p_0(v)$ and $p_1(v)$ are computed in a numerically stable manner. These adjustment handles potentially challenging situations, though it does make some trade-offs with respect to certain boundary cases.

Remark 1. *When $\phi(v) = 0$, then by definition log odds-product is 0, which implies that odds-product is 1. Since we have $p_1 = p_0 \exp\theta(v)$, then substituting the odds-product:*

$$OP = \frac{p_0 p_1}{(1-p_0)(1-p_1)} = \frac{p_0^2 \exp(\theta(v))}{(1-p_0)(1-p_0 \exp\theta(v))} = 1.$$

After simplification, we obtain the following expression $p_0 = \frac{1}{1+\exp\theta(v)}$.

This implies that any possible pairs of $\theta(v)$ and $\phi(v)$ has $\{p_0(v), p_1(v)\} \in (0,1) \times (0,1)$. In other words, the nuisance parameter is unconstrained and variation independent of the relative risk, which is shown in Figure 2.3. The full coverage of the likelihood allows us to make predictions and use maximum likelihood for parameter estimations.

Subsequently, we provide the formulation of the log-likelihood that serves as the foundation to derive the Maximum Likelihood Estimates (MLE) for α and β. The log-likelihood for one observation is expressed as:

$$\ell(\alpha,\beta) = Y\log(P(Y=1|X,V;\alpha,\beta)) + (1-Y)\log(P(Y=0|X,V;\alpha,\beta)). \quad (2.5)$$

The implementation of the maximum likelihood can be found in the `brm` package on CRAN.

For any observation given the treatment and outcome, the log-likelihood can be further refined and represented as one of the four cases below.

1. Consider the case with $X = 0, Y = 1$. Using Equations 2.3 and 2.5, the log-

likelihood function can be further expressed as

$$\begin{aligned}
&\ell(\alpha,\beta) = \log(p(Y=1 \mid X=0, V=v)) = \log(p_0(v)) \\
&= \log \frac{-\left(e^{\theta(v)}+1\right)e^{\phi(v)} + \sqrt{e^{2\phi(v)}\left(e^{\theta(v)}+1\right)^2 + 4e^{\theta(v)+\phi(v)}\left(1-e^{\phi(v)}\right)}}{2e^{\theta(v)}\left(1-e^{\phi(v)}\right)} \\
&= \log\left\{-e^{\theta(v)\phi(v)} - e^{\phi(v)} + \sqrt{e^{2\phi(v)}(e^{\theta(v)}+1)^2 + 4e^{\theta(v)+\phi(v)}(1-e^{\phi(v)})}\right\} \\
&\quad - \log(e^{\phi(v)}-1) - \log(2).
\end{aligned}$$

2. Consider the case with $X=1, Y=1$. Using Equations 2.4 and 2.5, the log-likelihood function can be further expressed as

$$\begin{aligned}
&\ell(\alpha,\beta) = \log(p(Y=1 \mid X=1, V=v)) \\
&= \log(p_0(v)) + \log(e^{\theta}(v)) \\
&= \log\left\{-e^{\theta v\phi(v)} - e^{\phi v} + \sqrt{e^{2\phi(v)}(e^{\theta(v)}+1)^2 + 4e^{\theta(v)+\phi(v)}(1-e^{\phi(v)})}\right\} \\
&\quad - \log(e^{\phi(v)}-1) - \log(2) + \log(e^{\theta}(v)).
\end{aligned}$$

3. Consider the case with $X=0, Y=0$. Using Equations 2.3 and 2.5, the log-likelihood function can be further expressed as

$$\begin{aligned}
&\ell(\alpha,\beta) = \log(p(Y=0 \mid X=0, V=v)) = \log(1-p_0(v)) \\
&= \log\left\{1 - \frac{-\left(e^{\theta(v)}+1\right)e^{\phi(v)} + \sqrt{e^{2\phi(v)}\left(e^{\theta(v)}+1\right)^2 + 4e^{\theta(v)+\phi(v)}\left(1-e^{\phi(v)}\right)}}{2e^{\theta(v)}\left(1-e^{\phi(v)}\right)}\right\}.
\end{aligned}$$

4. Lastly, we consider the case with $X=1, Y=0$. Using Equations 2.4 and 2.5, the log-likelihood function can be further expressed as

$$\begin{aligned}
&\ell(\alpha,\beta) = \log(1-p_1(v)) = \log(1-p_0(v)e^{\theta(v)}) \\
&= \log\left\{1 - \frac{-\left(e^{\theta(v)}+1\right)e^{\phi(v)} + \sqrt{e^{2\phi(v)}\left(e^{\theta(v)}+1\right)^2 + 4e^{\theta(v)+\phi(v)}\left(1-e^{\phi(v)}\right)}}{2\left(1-e^{\phi(v)}\right)}\right\}.
\end{aligned}$$

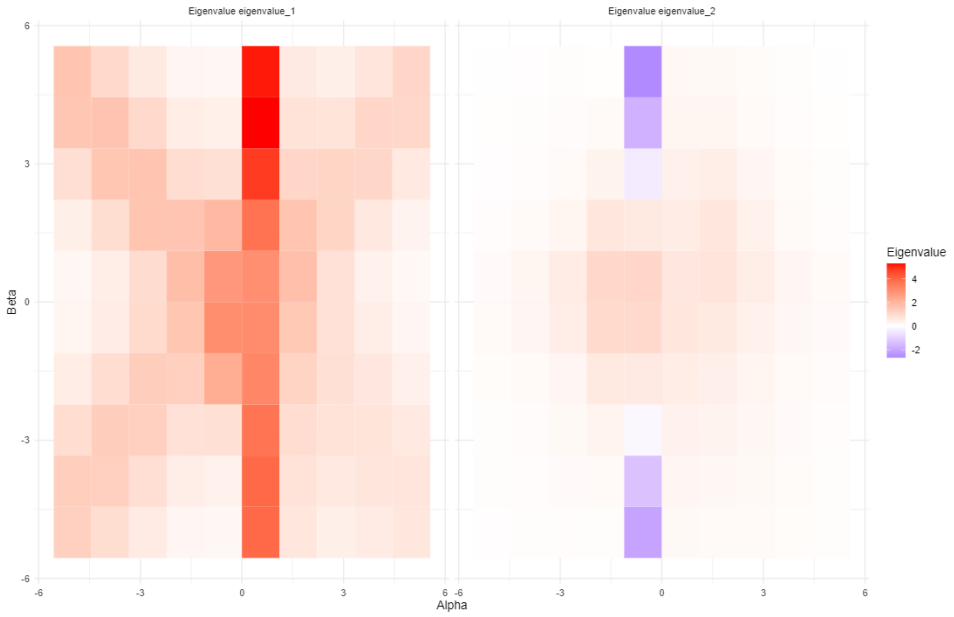

Figure 2.4: Hessian matrix eigenvalues at different points in parameter space

Through numerical methods, we demonstrate the non-convex nature of the log-likelihood function. This is achieved by computing the Hessian matrix and its eigenvalues across different points in the parameter space. We further identify points where the Hessian matrix exhibits a negative eigenvalue.

We create a grid of parameter values for α and β, both ranging from -5 to 5 with 10 equally spaced points. Using this grid, we compute the eigenvalues of the Hessian matrix for each combination of α and β, setting $p = 1$ for both parameters. The resulting eigenvalues are visualized in Figure 2.4 to show how they vary across the parameter space. In Figure 2.4, blue tiles represent negative eigenvalues, white tiles indicate eigenvalues close to zero, and red tiles represent positive eigenvalues. The heatmap identifies regions in the parameter space with negative eigenvalues, which indicates the non-convex nature of the objective function.

2.2 Least Absolute Shrinkage and Selection Operator

The Least Absolute Shrinkage and Selection Operator (LASSO) proposed by Tibshirani [17] is a regularization method designed for statistical models. Primarily applied in the context of linear regression, LASSO introduces a penalty term to the log-likelihood function, controlling the complexity of the model and discouraging overly complex solutions that overfit the data. Consequently, LASSO has the effect of shrinking the estimated coefficients towards zero and can even set some coefficients to exactly zero, effectively performing variable selection. In this thesis, we demonstrate how LASSO can be applied to our proposed model in Chapter 3, while the principles and mechanics of LASSO first illustrated in the simpler context of linear regression for clarity and simplicity. Furthermore, LASSO is often used when the number of covariates are more than the number of observations. It shrinks the absolute values of the regression coefficients towards zero in order to reduce the complexity of the model and control for overfitting.

LASSO aims to minimize the residual sum of squares between the predicted and the observed outcomes. In other words, LASSO solves the following problem:

$$\min_{(\beta_0,\beta)\in\mathbb{R}^{p+1}} \left[\sum_{i=1}^{N} \left(Y_i - \beta_0 - X_i^\top \beta \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right].$$

In this equation, $Y_i \in \mathbb{R}$ is the observed response for the i^{th} observation and $X_i \in \mathbb{R}^p$ is the corresponding p-dimensional vector of predictors. We have N pairs of observations (X_i, Y_i), each i ranging from 1 to N. The uppercase represent random vectors, and the lowercase represents the realization of the random vectors. The terms β_0 and β are the intercept and the vector of coefficients to be estimated, respectively. However, it's important to note that the aforementioned equation applies to linear models. The same exact expression does not extend directly to the log-likelihood function of the model presented by Richardson et al. [14]. It is worth pointing out that LASSO is sensitive to scaling of the predictor variables, otherwise the LASSO penalty does not apply evenly to all variables, and ends up penalizing variables with larger scales. Therefore it is important to standardize the observed

values of X before fitting the model:

$$\sum_{i=1}^{N} x_{ij} = 0, \quad \frac{1}{N}\sum_{i=1}^{N} x_{ij}^2 = 1, \quad j = 1, \ldots, p.$$

The regularization parameter λ determines the amount of shrinkage applied to the regression coefficients. Larger values of λ implies that more coefficients are shrunk to zero. However, for high-dimensional data where $p > N$, the optimal solutions of LASSO have up to N non-zero entries due to the nature of the convex optimization problem [18].

Friedman et al. [4] suggested a data-driven approach in creating a decreasing sequence of values for λ, starting at the smallest value $\lambda_{\max}$ for which the entire vector of coefficients are shrunk to zero. They obtained $\lambda_{\max}$ as follows:

$$\lambda_{\max} = \frac{1}{N}\max_j |\langle x_j, y \rangle|.$$

and constructed a sequence of J values of λ decreasing from $\lambda_{\max}$ to $\varepsilon\lambda_{\max}$ on the log scale. Typical values are $\varepsilon = 0.001$ and $J = 100$. Next, there are several ways to choose the optimal regularization parameter from the sequence of λ either by efficient sample reuse (cross-validation) or analytically (AIC, BIC, eBIC):

1. Cross-validation (CV): CV is a widely used method for model validation. In particular, k-fold cross-validation involves partitioning the data into k equal-sized subsets. The model is then trained on k-1 subsets and tested on the remaining subset. This process is repeated k times, each time with a different subset held out for testing. The average prediction error across all k iterations is calculated for each λ, and the λ resulting in the lowest prediction error is selected [6]. CV can be used with any loss function, as it directly estimates the prediction error.

2. Information Criterion: The *Akaike information criterion* and *Bayesion information criterion* are applicable in settings where the fitting of the model is carried out by maximization of the a log-likelihood function. For a binary

regression model, such as the logistic regression, we have:

$$\text{AIC} = -\frac{2}{N} \cdot \text{loglik} + 2 \cdot \frac{d}{N},$$

$$\text{BIC} = -2 \cdot \text{loglik} + (\log N) \cdot d,$$

where d stands for the number of parameters fitted in the model. The information criterion methods attempt to quantify both the model performance on the training dataset and the complexity of the model. The aim is to select a model with smaller value of AIC or BIC. However, AIC tends to choose complex models as $N \to \infty$ while vice versa for BIC [6]. On the other hand, BIC has the drawback of choosing an overly simple model when there is a finite sample size, due to its heavier penalty on model complexity when N is small.

There has since been many variants of information criteria to address the limitations. For example, eBIC was proposed by Chen and Chen [2] that consider both the number of unknown parameters and the complexity of the model space, and is particularly suitable for model selection fitted on high-dimensional data.

2.3 Optimization Methods

Logistic regression, widely used for odds ratio estimation, benefits from a convex likelihood function that facilitates efficient parameter estimation via maximum likelihood methods. Estimating relative risk, as in the binary regression model, presents a unique challenge due to the non-convexity of the likelihood function. This feature complicates the application of standard gradient-based optimization methods as they may fail to converge to a global optimum. While there are optimization methods designed to handle non-convex problems, such as simulated annealing, they are often computationally demanding, making them less practical for routine use. In this chapter, we review various optimization methods used in relative risk regression estimation and discuss their pros and cons with a focus on finding reliable stationary points of the objective function.

2.3.1 Derivative-free Optimization Algorithms

Derivative-free optimization methods offer several advantages such as the ability to handle noisy and expensive objective functions and can explore the parameter space more extensively. These advantages make derivative-free optimization methods suitable for problems where derivative information is unavailable, unreliable, or impractical to obtain [15].

Derivative-free optimization methods can be dated back to the Nelder-Mead simplex algorithm [12], which is a direct search method, and has been widely used in practice. The Nelder-Mead algorithm is the default optimization method for the `optim` function in R and is robust but relatively slow. Furthermore, the Nelder-Mead algorithm may not converge to a global optimum, since it is a local search method that only explores a small region around the initial starting point to find a local optimum. However, depending on the objective function and initial starting point, it may get stuck in a local optimum and fail to find the global optimum.

The binary regression model [14] in the `brm` package utilizes the default Nelder-Mead algorithm in `R` to estimate the parameter of interest α and the nuisance parameter β.

2.3.2 First-order Optimization Algorithms

First-order optimization algorithms use the first derivative of a function to find its optimum.

Accelerated Proximal Gradient Descent (FISTA)

A classical approach for solving ℓ_1-regularized objective functions is to utilize proximal gradient descent (PGD) methods, also known as iterative shrinkage-thresholding algorithms (ISTA). Given a convex optimization problem of the form:

$$\min_{\theta} \{ f(\theta) + \lambda \|\theta\|_1 \}, \tag{2.6}$$

where $f(\theta)$ is a convex function, λ is a regularization parameter, and θ represents a vector of parameters we aim to optimize. The proximal gradient descent method iteratively updates the parameter θ through the following steps:

Input: initial guess θ_0, step size $t > 0$:

1. Set $\theta_k = \theta_0$ for $k = 0$
2. For $k = 1, 2, \ldots$, do the following:
   (a) Compute the gradient g_k of the objective function f at θ_k
   (b) Set $\theta_{k+1} = \text{soft-thresholding}(\theta_k - t g_k, \lambda t)$

where the soft-thresholding operator is defined as:

$$\text{soft-thresholding}(x, \lambda) = \text{sign}(x) \max(0, |x| - \lambda).$$

In the above algorithm, g_k is the gradient of f at θ_k, which is computed using the standard gradient descent update:

$$g_k = \nabla f(\theta_k).$$

The step size t is a hyperparameter that controls the step size in the gradient descent step. However, the slow convergence of the proximal gradient descent method has been well-discussed in the literature, and led to the development of the accelerated proximal gradient descent method (FISTA) [1].

Beck and Teboulle [1] considered the following general formulation of the problem:

$$\min_{\theta} \left\{ f(\theta) + g(\theta) : \theta \in R^d \right\}, \tag{2.7}$$

where $f : R^d \to R$ is some smooth convex function, and $g : R^d \to R$ is a convex but non-smooth function but continuous. The FISTA algorithm with constant step size is summarized in Algorithm 1.

The proximal operator of g with parameter t is defined as:

$$\text{prox}_{t,g}(y) = \arg\min_{x \in R^d} \left\{ g(x) + \frac{1}{2t} |x - y|^2 \right\},$$

where x is the variable over which the optimization occurs, and y denotes the current point in the iteration process of the proximal gradient descent method. Lastly, R^d signifies the d-dimensional real space, indicating that x can take any real values

Algorithm 1 FISTA with constant step size

Require: Initial value x_0, step size $t > 0$, momentum parameter $\alpha \in (0,1)$, tolerance $\varepsilon > 0$.

Ensure: Solution x^*.

Set $y_0 = x_0$, $t_k = t$, $k = 0$.
while not converged **do**
    Compute the gradient g_k of f at y_k.
    Compute $x_{k+1} = \text{prox}_{tg}(y_k - tg_k)$
    Compute $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$.
    Compute $y_{k+1} = x_{k+1} + \frac{t_k}{t_{k+1}}(x_{k+1} - x_k)$.
    if $\|x_{k+1} - x_k\| < \varepsilon$ **then**
        Terminate and return x_{k+1}.
    else
        Set $k = k + 1$.
    end if
end while

in d dimensions. The proximal operator essentially facilitates optimization problems by handling non-differentiable parts of a function and creating a surrogate function that is easier to minimize.

The choice of hyperparameters t, α in optimization algorithms can significantly impact the convergence rate and quality of the solution. By combining gradient descent with a momentum term, FISTA can take larger steps in the direction of the gradient, which helps avoid oscillations near the minimum and accelerates convergence. The use of the proximal operator step allows FISTA to handle nonsmoothness in the objective function, as in the case of an ℓ_1 penalty, and promote sparsity in the solution.

It is worth noting that the FISTA algorithm has certain constraints on the type of functions it can optimize. Specifically, the objective function $F(x) = f(x) + g(x)$ should in theory be convex [1], where $f(x)$ is a smooth convex function with a Lipschitz continuous gradient, and $g(x)$ is a convex function that is possibly nonsmooth to ensure the tractability of the algorithm.

Although convex problems are well-studied in the literature and can be globally optimized, many real-world problems are nonconvex, such as the objective

function of the binary regression model proposed by Richardson et al. [14]. In the next section, we touch on extensions of the accelerated proximal gradient descent methods for nonconvex optimization algorithm discussed.

Extensions of PGD Algorithms for Nonconvex Optimization

Although it is desirable to have convex optimization problems, many real-world problems are nonconvex. There has since been a lot of interest in developing algorithms that can solve nonconvex optimization problems. For example, Li and Lin [7] proposed the Monotone APG (mAPG) algorithm 2, which generates a sufficiently decreasing sequence of function values, and by using the Kurdyka-Lojasiewicz (KL) property, further established asymptotic convergence rates. However, the mAPG algorithm takes two proximal steps which can be computationally expensive.

Algorithm 2 mAPG

Require: $\mathbf{y}_1 = \mathbf{x}_1 = \mathbf{x}_0, t_1 = 1, t_0 = 0, \eta < \frac{1}{L}$

for $k = 1, 2, \cdots$ **do**
  $\mathbf{y}_k = \mathbf{x}_k + \frac{t_{k-1}}{t_k}(\mathbf{z}_k - \mathbf{x}_k) + \frac{t_{k-1}-1}{t_k}(\mathbf{x}_k - \mathbf{x}_{k-1})$.
  $\mathbf{z}_{k+1} = \text{prox}_{\eta g}(\mathbf{y}_k - \eta \nabla f(\mathbf{y}_k))$.
  $\mathbf{v}_{k+1} = \text{prox}_{\eta g}(\mathbf{x}_k - \eta \nabla f(\mathbf{x}_k))$.
  $t_{k+1} = \frac{\sqrt{4t_k^2+1}+1}{2}$.
  if $F(\mathbf{z}_{k+1}) \leq F(\mathbf{v}_{k+1})$ **then**
    $\mathbf{x}_{k+1} = \mathbf{z}_{k+1}$.
  else if $F(\mathbf{v}_{k+1}) \leq F(\mathbf{z}_{k+1})$ **then**
    $\mathbf{x}_{k+1} = \mathbf{v}_{k+1}$.
  end if
end for

On the other hand, Li et al. [8] proposed the the APGnc+ with adaptive momentum algorithm 4 outlined in the appendix, which not only exploits the KL-property, but also uses a single proximal step, which is computationally more efficient.

The field of non-convex optimization is still relatively new, and there are many other algorithms that have been proposed in the literature, although we only highlighted a few of them here.

Chapter 3

ℓ_1-Regularized Relative Risk Regression Model

Epidemiological studies frequently employ models that estimate relative risk to explore the relationships between diseases and their associated risk factors. The innovative binomial regression model introduced by Richardson et al. [14] offers significant advantages as it directly estimates the relative risk and deploys a log odds-product nuisance model, leading to independent parameter spaces with respect to variations. This model has also been adapted to delineate the multiplicative effect of treatments on binary outcomes [21].

However, the initial experiments conducted with this model were constrained to datasets possessing a relatively small number of risk factors, where the number of predictors (p) was less than the number of observations (N). Furthermore, the algorithm exhibits certain limitations when it comes to estimating sparse solutions, especially in scenarios where the underlying model inherently possess sparsity. In such situations, the algorithm might fail to correctly identify the nonzero elements or might inaccurately estimate their magnitudes, thus leading to suboptimal performance.

In response to these challenges, this chapter proposes a novel estimator and develops an algorithm for the relative risk model. This approach facilitates variable selection even in contexts where p exceeds N.

We initiate our discussion by justifying the need for such a model, followed by

a comprehensive description of the model itself. Subsequently, we elucidate the algorithms used for fitting the model to the data.

3.1 Motivation

The parameter of interest α and nuisance parameter β from Equations 2.2 and 2.1 may be estimated directly via maximum likelihood estimation (MLE). The R package `brm` implements a methodology based on the Nelder-Mead simplex algorithm. This algorithm operates primarily by constructing a polytope with $p+1$ vertices in p dimensions. In the next step, the algorithm manipulates this simplex—reflecting, expanding, contracting, and shrinking it—based on the function values at its vertices. This iterative process continues until a predefined convergence criterion is satisfied. The Nelder-Mead optimization method, while capable of handling cases where the number of parameters (p) surpasses the number of observations (N), might still pose challenges. Overfitting and non-uniqueness of solutions are potential issues that could emerge. Crucially, without regularization, the MLE is not well-defined.

Furthermore, when most predictors are not related to the outcome, a non-regularized model may face challenges in correctly identifying the number of true predictors. The algorithm may overfit to the noise in the data, assigning non-zero estimated coefficients to irrelevant predictors and thereby reducing the model's predictive performance and interpretability.

In the following example, we demonstrate the challenges of fitting the `brm` function to a dataset with sparse predictors. We simulate datasets where the treatment/exposure X_i for an individual i is assigned according to the parametric model for propensity score:

$$P(X_i = 1|V_i;\gamma) = \text{expit}(\gamma^T V_i) = \frac{1}{1+\exp(-\gamma^T V_i)},$$

where γ is set to be 5 for the first five predictors and zero for the rest. We also include an intercept term which we compute based on the data such that the distribution of $\mathbf{X}$ has approximately half of $X_i = 0$, and the other half $X_i = 1$.

The entries of the covariate vector V_i are independently drawn from $\text{Unif}(-1,1)$.

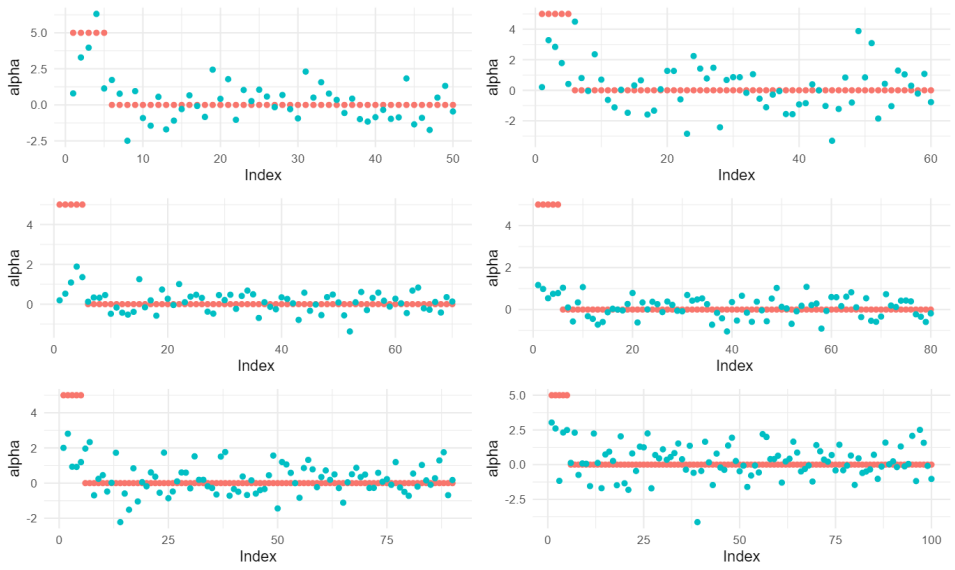

Figure 3.1: Estimated parameters of α from `brm` (blue) and the true parameter values (red). The x-axis shows the index of the predictor for α, the y-axis shows the value of α.

Let p_α represent the number of coefficients we need to estimate for α, p_β represent the number of coefficients we need to estimate for β. Since it is always the case that $p_\alpha = p_\beta$, we simplify the remaining discussion by setting $p = p_\alpha = p_\beta$. We set the first five coefficients of α and β to be 5 and the rest to be zero. The dimension p is set to 50, 60, 70, 80, 90, 100, while keeping the observation size N fixed at 100. The outcome Y_i is generated based Equations 2.1 and 2.2. It is worth noting that in total there are $p_\alpha + p_\beta$ parameters to be estimated.

The estimated parameters of α from the `brm` function are visualized in Figure 3.1 and compared with the true parameter values (red).

In Figure 3.1, the x-axis shows the index of the predictor for α, the y-axis shows the value of α. Blue points represent the estimated parameters of α from `brm`, while the red points represent the true parameter values. While the method appears to pick up signals from the initial five predictors, it also estimates non-zero values for several irrelevant predictors, thereby capturing noise in the data. In such scenarios, methods that leverage regularization techniques, like employing an ℓ_1-

penalty to the objective function, are often more effective at distinguishing relevant predictors from noise. These techniques promote sparsity in the estimated model and help reduce the risk of overfitting.

Given this motivation, we introduce our estimator for estimating relative risk which incorporates regularization to promote sparsity. We also develop an algorithm for fitting the model to the data.

3.2 Model-Specification

Building on 2.5, we specify the ℓ_1-regularized negative log-likelihood function for the relative risk model:

$$\begin{aligned} PNLLH(\alpha,\beta,\lambda_1,\lambda_2) &= -(\sum_{i=1}^{n} y_i \log P(Y_i=1|X_i=x_i,V_i=v_i) \\ &\quad + (1-y_i)\log P(Y_i=0|X_i=x_i,V_i=v_i)) \\ &\quad + \sum_{j=1}^{p} \lambda_1|\alpha_j| + \lambda_2|\beta_j| \\ &= -(\sum_{i=1}^{n} y_i \log\left(x_i p_1(v_i) + (1-x_i)p_0(v_i)\right) \\ &\quad + (1-y_i)\log\left(x_i(1-p_1(v_i)) + (1-x_i)(1-p_0(v_i))\right)) \\ &\quad + \sum_{j=1}^{p} \lambda_1|\alpha_j| + \lambda_2|\beta_j|, \end{aligned}$$

where x_i represents the treatment/exposure X for the i-th observation and is binary, v_i represents the predictors V for the i-th observation, and $p_1(v_i)$ and $p_0(v_i)$ are expressed in 2.3 and 2.4. To recap, $p_1(v_i)$ represents the probability of the event $Y_i = 1$ given $X_i = 1$ and $V_i = v_i$, and $p_0(v_i)$ represents the probability of the event $Y_i = 1$ given $X_i = 0$ and $V_i = v_i$. The terms $|\alpha_j|$ and $|\beta_j|$ represent the absolute values of the coefficients, and λ_1, λ_2 are the regularization parameters.

The proposed method introduces regularization to mitigate overfitting, a situation where a model performs well on training data but poorly on unseen data due to excessive complexity. By imposing a penalty on the model's complexity through the λ terms, the model is discouraged from fitting too closely to noise in the training data.

In high-dimensional settings, where the number of predictors is large, the penalty functions as a form of automatic feature selection. The penalization promotes sparsity in the estimated coefficients, reducing the effective dimensionality of the problem. This leads to more interpretable models, as irrelevant predictors are more likely to have their estimated coefficients shrunk to zero.

Furthermore, the introduction of the penalty terms achieves an optimal balance between bias and variance in the model's predictions, which is a key consideration in model selection. While unpenalized methods may suffer from high variance due to overfitting, the introduction of a penalty term effectively shrinks the estimated coefficients towards zero, thereby reducing the variance of the estimators at the cost of introducing a slight bias. This controlled bias-variance trade-off often enhances the model's predictive performance, making it more reliable for prediction.

In practice, it can be justified that λ_1 and λ_2 may take on different values. Consider a scenario where the predictors in V have a strong association with the log relative risk, yet they exhibit uncertain or more tenuous connections to the log odds product. In such a scenario, we might choose a smaller λ_1 (less regularization) for the α parameters and a larger λ_2 (more regularization) for the β parameters. This choice of different λ values could also be based on the underlying assumption about the sparsity of the true model. If we assume that the true model related to α (or β) is sparse (i.e., most of its coefficients are zero), a larger λ value could be chosen to induce more sparsity in the estimated model.

The choice of different λ values should ideally be guided by domain knowledge, scientific understanding of the relationships between predictors and the outcomes, or empirical evidence (e.g., cross-validation). However, for the purpose of this paper, we assume that λ_1 and λ_2 are the same and denote them as λ. We leave the investigation of different pairs of λ_1, λ_2 values for future work. Therefore the objective function can be further simplified to:

$$
\begin{aligned}
PNLLH(\alpha,\beta,\lambda_1,\lambda_2) = &-(\sum_{i=1}^{n} y_i \log P(Y_i=1|X_i=x_i,V_i=v_i) \\
&+ (1-y_i)\log P(Y_i=0|X_i=x_i,V_i=v_i)) \\
&+ \sum_{j=1}^{p} \lambda(|\alpha_j|+|\beta_j|) \\
= &-(\sum_{i=1}^{n} y_i \log\left(x_i p_1(v_i) + (1-x_i)p_0(v_i)\right) \\
&+ (1-y_i)\log\left(x_i(1-p_1(v_i)) + (1-x_i)(1-p_0(v_i))\right)) \\
&+ \sum_{j=1}^{p} \lambda(|\alpha_j|+|\beta_j|).
\end{aligned}
$$

We aim to solve the following problem:

$$
\min_{\alpha,\beta\in\mathbb{R}^p} PNLLH(\alpha,\beta,\lambda).
$$

3.3 Algorithm

Having laid out the specifics of the proposed penalized model, we now transition into a discussion on the algorithmic approach employed to estimate the model parameters. This section dives into the computational techniques and processes, which are vital for transforming the theoretical underpinnings of our model into a practical tool. We discuss the optimization algorithms along with how they are implemented in the context of our model. Furthermore, we elaborate on practical considerations such as choice of tuning parameters. Finally, we address the algorithm's limitations and possible strategies to enhance its performance.

3.3.1 Optimization

Despite the non-convex nature of our objective function, we employ the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) for its effectiveness in large-scale optimization problems [1]. FISTA is originally designed for convex problems, notably those involving ℓ_1-regularization, which resembles the penalty term in our

method which can be written in the form of 2.6. However, FISTA can be, and has been, adapted and applied to certain non-convex problems with promising results [7, 20]. By leveraging FISTA, we benefit from its accelerated convergence rate, especially when compared to the standard ISTA (Iterative Shrinkage-Thresholding Algorithm).

We seek to employ the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) to solve the optimization problem posed by the ℓ_1-regularized relative risk regression model. The algorithm `rbrm` is outlined in Algorithm 3. We denote $L(\alpha, \beta)$ as the non-penalized negative log-likelihood function.

Algorithm 3 Regularized Binary Regression Model for Relative Risk (rbrm)

Input: $X, Y, V, \lambda, \varepsilon, s_1, s_2, m$
Output: α, β
Initialize $\alpha^{(0)} = \beta^{(0)} = 0$
$t^{(0)} = 1$
for $k = 1, 2, \ldots, m$ **do**
    $t^{(k)} = \frac{1+\sqrt{1+4(t^{(k-1)})^2}}{2}$
    $z^{(k)} = \alpha^{(k-1)} - s_1 \nabla_\alpha L(\alpha^{(k-1)}, \beta^{(k-1)})$
    $\alpha^{(k/2)} = \text{sign}(z^{(k)}) \max(|z^{(k)}| - \lambda s_1, 0)$
    $\alpha^{(k)} = \alpha^{(k/2)} + \frac{t^{(k-1)}-1}{t^{(k)}}(\alpha^{(k/2)} - \alpha^{(k-1)})$
    $z^{(k)} = \beta^{(k-1)} - s_2 \nabla_\beta L(\alpha^{(k)}, \beta^{(k-1)})$
    $\beta^{(k/2)} = \text{sign}(z^{(k)}) \max(|z^{(k)}| - \lambda s_2, 0)$
    $\beta^{(k)} = \beta^{(k/2)} + \frac{t^{(k-1)}-1}{t^{(k)}}(\beta^{(k/2)} - \beta^{(k-1)})$
    if $\max(|\nabla_\alpha L(\alpha^{(k)}, \beta^{(k)})|) < \varepsilon$ and $\max(|\nabla_\beta L(\alpha^{(k)}, \beta^{(k)})|) < \varepsilon$ **then**
        break
    end if
end for

The parameter updating involves two stages at each iteration. First, the gradient of the non-penalized likelihood function is calculated and used to make an update - denoted by $\alpha^{(k/2)}$, $\beta^{(k/2)}$. The impact of this gradient step is controlled by the step size parameters s_1 for α and s_2 for β.

Next, the soft-thresholding function is applied to $\alpha^{(k/2)}$, $\beta^{(k/2)}$, which intro-

duces sparsity into the parameter estimates and corresponds to the ℓ_1 regularization in our objective function. This operation is controlled by the regularization parameter λ. Finally, a momentum term $\frac{t^{(k-1)}-1}{t^{(k)}}$ is added to the updates to accelerate the algorithm's convergence.

3.3.2 Choice of Regularization Parameters

The tuning parameter λ is selected via k-fold cross-validation. The optimal λ is chosen as the values that minimize the cross-validation error, which in our case is the binomial deviance.

$$\text{deviance} = -2\frac{1}{n}\sum_{i=1}^{n}\{y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)\}$$

Based on the method suggested in Friedman et al. [4], we create a decreasing sequence of λ values and use the largest λ such that it is set to be the maximum absolute correlation between the predictors and the binary outcomes. The sequence of J values is on the log scale from $\lambda_{\max}$ to $\lambda_{\min}$, where $\lambda_{\min} = \varepsilon\lambda_{\max}$. In our implementation, we set $\varepsilon = 0.001$.

3.3.3 Discussion and Limitations

The proposed algorithm exhibits several tunable parameters that can potentially enhance its performance. Among these are the step size parameters s_1 and s_2, which dictate the impact of the gradient step in the calculation of α and β respectively. An empirical analysis suggests that a larger s_1 in comparison to s_2 yields superior results, however, additional fine-tuning may be necessary to optimize performance for different datasets. For our simulations, we have set s_1 and s_2 to values of 0.06 and 0.02, respectively. We set the maximum number of iterations m to 3000, which is sufficient for convergence in most cases.

The selection of λ poses its own challenges. We observed multiple instances where certain λ values led to over-regularization, thus yielding a model without any predictors. To address this issue, we implemented a warm start in our cross-validation approach, where the solution from the previous λ value is used as the initial values for the next. This generally improves the efficiency and stability of

the optimization. In the future, we plan to explore alternative methodologies to determine optimal λ values and further reduce instances of null model outputs.

Common practice, as seen in solutions such as `glmnet`, frequently utilize a grid of 100 λ values coupled with 10-fold cross-validation. However, the selection of J and k involves a delicate balance between computational efficiency and accuracy. Given that our functions are entirely implemented in `R`, which introduces higher computational costs compared to `glmnet`, we opted for $J = 50$ and $k = 3$ in our simulations.

It's crucial to note that our algorithm, due to the non-convex nature of the objective function, does not guarantee convergence to the global minimum. Therefore, future work may include exploring alternative stopping criteria. Furthermore, the initial values for α and β play a significant role in the algorithm's efficacy. While our default setting initializes them to 0 for α and 0.01 for β, leveraging prior knowledge about these parameters could be advantageous, serving as potentially more insightful initial values.

Chapter 4

Experiments

We conduct a series of simulations, particularly focusing on scenarios where the number of parameters to estimate exceeds the number of observations, to understand the properties of our proposed algorithm. First, we outline the methodology utilized for data generation and delineate the simulation settings adhered to during our experiments. Next, we introduce the competing models involved and detail the metrics employed to evaluate performance. Lastly, we present the results derived from our simulations and provide an analysis of our findings.

4.1 Data Generation and Simulation Settings

We consider simulations where $p_\alpha > N$, and has sparsity in the real coefficients, which resembles many real world applications such as gene expression data.

We define the covariate vector V_i by randomly drawing entries from a uniform distribution, specifically Unif$(-1,1)$, and $i = 1,2,\cdots,N$. For the remaining section, the parameter p denotes the number of coefficients that we aim to estimate for the parameter vector α. The treatment variable X_i is determined based on the propensity score model given by:

$$P(X_i = 1|V_i = v_i;\gamma) = \frac{1}{1+\exp(-\gamma^T v_i)}. \tag{4.1}$$

Here, γ represents a vector of coefficients. We compute an additional intercept term for γ to adjust the propensity score model such that it results in approxi-

mately equal likelihoods of the treatment being assigned or not. In other words, an intercept is computed to ensure that, once γ and v_i are transformed through the sigmoid function, on average, the probability aligns closely with a value of 0.5. The pseudo-algorithm is presented in the appendix. With every observation given a propensity score, we then generate the treatment variable X_i by drawing from a Bernoulli distribution with probability $P(X_i = 1|V_i = v_i; \gamma)$.

Next, the outcome Y_i is generated based on Equations 2.1, 2.2, and 4.1. By specifying the true coefficients α, β that we apply to Equations 2.3 and 2.4, we obtain the probability of the outcome Y_i being equal to 1 given the treatment X_i. We then draw from a Bernoulli distribution with probability $P(Y_i = 1|X_i; \alpha, \beta)$.

We consider the following simulation settings, each with 15 replicates:

1. $N = 100, p = 150$ and $\gamma = (\underbrace{2,\ldots,2}_{10},\underbrace{0,\ldots,0}_{140})$ with an additional intercept computed to offset the propensity score model.

$$\alpha = \beta = (\underbrace{2,\ldots,2}_{10},\underbrace{0,\ldots,0}_{140})$$

2. $N = 100, p = 200$ and $\gamma = (\underbrace{2,\ldots,2}_{10},\underbrace{0,\ldots,0}_{190})$ with an additional intercept computationally obtained.

$$\alpha = \beta = (\underbrace{2,\ldots,2}_{10},\underbrace{0,\ldots,0}_{190})$$

3. $N = 100, p = 500$ and $\gamma = (\underbrace{2,\ldots,2}_{50},\underbrace{0,\ldots,0}_{450})$ with an additional intercept computationally obtained.

$$\alpha = \beta = (\underbrace{2,\ldots,2}_{50},\underbrace{0,\ldots,0}_{450})$$

In Simulation 1, we estimate 300 parameters from a sample size of 100, producing a sample-to-predictor ratio of 0.67. From these 300 parameters, only 20 carry significant signals. Of these, 10 are sparse signals located within α. The

	Sample-to-Predictor Ratio	Sparsity Ratio
Simulation 1	0.67	0.93
Simulation 2	0.5	0.95
Simulation 3	0.2	0.9

Table 4.1: Simulation Sample-to-Predictor Ratio and Sparsity Ratio Summary.

same applies to the β. Thus, we have a sparsity ratio of 0.93.

In Simulation 2, with the same sample size of 100, we estimate 400 parameters. This results in a decreased sample-to-predictor ratio of 0.5. Here, only 20 out of the 400 parameters carry significant signals, half of which are sparse signals found within α, which is the same setting for β. This leads to a sparsity ratio of 0.95.

Finally, Simulation 3 keeps the sample size constant at 100 but increases the total number of parameters $p_\alpha + p_\beta$ to be estimated to 1000. Simulation 3 has an even lower sample-to-predictor ratio of 0.2. The sparsity ratio is 0.9, which is around the range of the sparsity ratio in the previous two simulations.

For each simulation, we have a separate random subset of 100 observations to evaluate prediction performance on new data.

4.2 Competing Models

The current standard approach for modeling the dependence of relative risk on baseline covariates relies on the generalized linear model (GLM) framework, which can be expressed as:

$$g(E(Y_i|X_i,V_i)) = X_i\alpha^T V_i + \beta^T V_i, \tag{4.2}$$

where $g(\cdot)$ is the link function. With g being the log link function, and $Y_i \sim Bernoulli(b)$, with b being the probability of having $Y = 1$. Equation 4.2 represents the mean function induced by a log-binomial or Poisson regression model. However, in practice standard statistical software may report failed convergence when attempting to fit log-binomial models in certain settings [19], which was the case in Richardson et al. [14]. Therefore, our focus is on the Poisson regression

model, which can be rewritten in the following equivalent form:

$$\log(RR(v_i)) = \alpha^T v_i, \tag{4.3}$$

$$\log(p_0(v_i)) = \beta^T v_i, \tag{4.4}$$

where $p_a(v_i) \equiv E(Y_i|X_i = x_i, V_i = v_i), \quad x_i = 0, 1$. To see this, we add the left hand sides of 4.3 and 4.4 to obtain the sum of the right hand sides:

$$\log(p_1(v_i)) \equiv \log E(Y_i|X_i = 1, v_i) = \alpha^T v_i + \beta^T v_i = X_i \alpha^T v_i + \beta^T v_i$$

Equation 4.3 models the parameter of interest, while Equation 4.4 is a nuisance model or a baseline model used in the estimation of the parameter of interest. This way, we can essentially represent the Poisson regression model in the same form as Equation 2.1 and Equation 2.2 from the binary regression model.

In `R`, the Poisson regression model is fit using the `glm()` function which works well for dataset with small p. However, for large p small N, the `glm()` function likely overfits the data and result in poor prediction performance. Therefore, we resort to using the `glmnet()` function from the `glmnet` package [4] to fit an ℓ_1-regularized Poisson regression.

For consistency, we fit both the binary regression model and the Poisson regression without an intercept term after standardizing the data. It then follows that the input matrix for the `glmnet` function is a $N \times 2p$ matrix of covariates in the following form:

$$\begin{bmatrix} v_{11} & v_{12} & v_{13} & \cdots & v_{1p} & v_{11} & v_{12} & v_{13} & \cdots & v_{1p} \\ 0 & 0 & 0 & \cdots & 0 & v_{21} & v_{22} & v_{23} & \cdots & v_{2p} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ v_{n1} & v_{n2} & v_{n3} & \cdots & v_{np} & v_{n1} & v_{n2} & v_{n3} & \cdots & v_{np} \end{bmatrix}, \tag{4.5}$$

here, the first p columns are the covariates for α and the last p columns are the covariates for β. For example, the first row represents an observation with $X = 1$, and the second row represents an observation with $X = 0$ hence the first p columns are zero. In our simulations, `glmnet` performs variable selection by shrinking the coefficients of the covariates towards zero and we choose the optimal tuning

Method	Results for the following examples:		
	Setting 1	Setting 2	Setting 3
brm	4.6	4.1	6.0
regularized RR regression	2.8	2.4	2.9
regularized Poisson	2.3	2.4	2.5

Table 4.2: Median Absolute Errors (MAE) of $\log RR$ based on 15 replicates.

parameter λ using 10-fold cross validation and set it to the minimum λ that yields the minimum deviance. Finally, the estimated α from the regularized Poisson regression is used to get an estimate of the relative risk.

Additionally, we evaluate the prediction performance of the regularized relative risk regression (rbrm) and regularized Poisson regression (rpoisson) against the binary regression model (brm) developed by Richardson et al. [14], available in the `brm` package. It's worth noting that we do not assess variable selection for the brm model in this comparison, as the estimation process for this particular model does not incorporate sparsity in the parameters. We call `brm:::max.likelihood()` that returns the estimate of the coefficients. Directly invoking the `brm()` function can lead to error messages. These errors originate from the calculation of variance - when $p > N$, the Fisher Information matrix becomes non-invertible.

4.3 Simulation Results

4.3.1 Prediction Evaluation

Table 4.2 and Figure 4.1 summarize the prediction results. We use the median absolute error (MAE) to measure the prediction performance since it is robust to outliers. Each replicate predicts the log relative risk for 100 new observations. The median absolute error, defined as the median of $|\widehat{\log(RR)}_i - \log(RR)_i|$ is then calculated across the new observations $i = 1, 2, \cdots, 100$. Lower values of MAE indicate superior predictive performance as the predictions are closer to the target values. Given its limitations in handling sparse and high-dimensional models, we anticipate that the brm model will exhibit higher MAE values across all three simulation settings compared to the two regularized models.

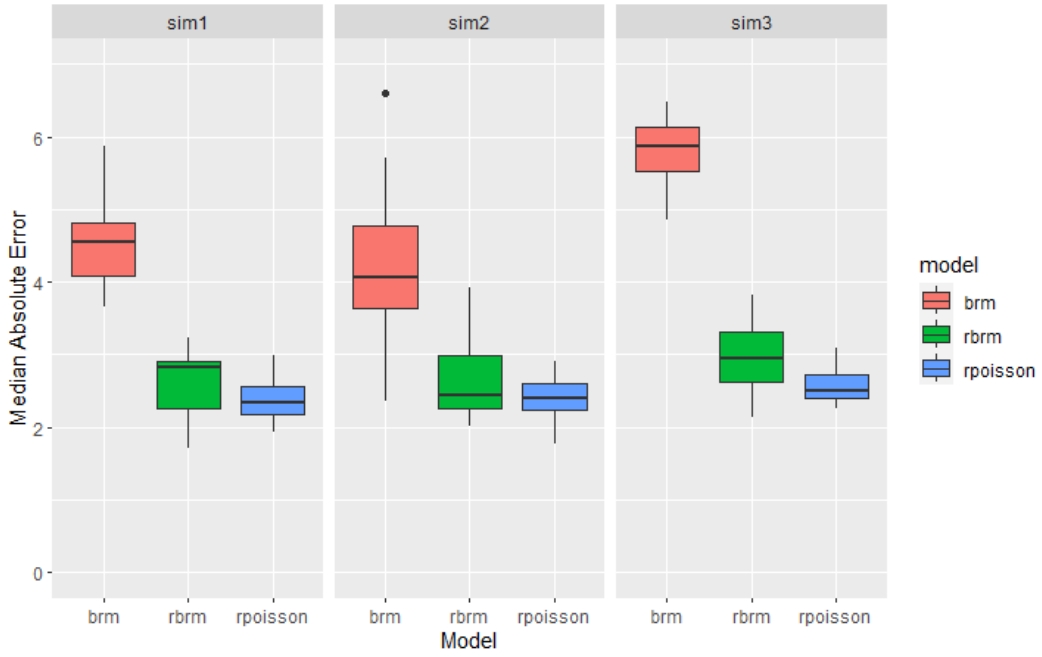

Figure 4.1: Distribution of Median Absolute Errors (MAE) of log *RR* replicates.

Method	Setting 1			Setting 2			Setting 3		
	brm	rbrm	rpoisson	brm	rbrm	rpoisson	brm	rbrm	rpoisson
Min.	-24.73	-2391.33	-11.44	-23.55	-15.62	-10.36	-24.62	-25.76	-27.38
1st Qu.	-4.47	-2.75	-2.23	-4.07	-2.52	-2.28	-6.19	-5.56	-5.56
Median	-0.37	0.16	0.09	0.15	-0.08	0.09	0.12	0.06	0.12
Mean	-0.16	1.28	0.10	-0.02	0.00	0.09	0.06	-0.02	0.04
3rd Qu.	3.94	2.89	2.49	4.06	2.60	2.56	6.06	5.37	5.44
Max.	23.13	2201.81	13.57	21.13	14.24	10.91	25.27	30.99	30.54

Table 4.3: Summary statistics of individual log relative risk bias.

Table 4.2 shows that across all the simulation settings, the MAE of the brm is indeed higher than the two regularized methods. However, we found that the regularized Poisson regression has a lower MAE than the regularized RR regression in all settings despite the varying sample-to-predictor ratio and sparsity ratio. Figure 4.1 further shows that the MAE of brm is higher and has a wider spread than the other two methods, but the the regularized RR regression is more spread out than the regularized Poisson regression. Simulation 1 and 2 exhibited similar magnitude

Method	Results for the following examples:		
	Setting 1	Setting 2	Setting 3
regularized RR regression	26	26	36
regularized Poisson	1	1	1

Table 4.4: Median number of non-zero α based on the 15 replicates.

in differences between the MAE of brm compared to the two regularized methods. However, in Simulation 3, the MAE of brm is much higher than the two regularized methods, likely due to overfitting from a lower sample-to-predictor ratio, where it is capturing noise in the coefficient estimates rather than the actual sparse signal. Furthermore, all methods showed an increase in MAE in simulation 3 compared to Simulation 1 and 2, which is expected given the lower sample-to-predictor ratio, but the regularized methods had a smaller increase in MAE than the brm method, which is likely the benefit of regularization preventing overfitting.

Table 4.3 shows the summary statistics of the individual log relative risk bias collected across the 15 replicates, 1500 observations in total. Log relative risk bias is defined as the difference between the oracle log RR and the estimated log RR. Summary statistics of the individual log relative risk bias in Table 4.3 gives us a better understanding of the prediction performance on the individual level. The brm method has a wider spread within the first and third quartile but the difference becomes smaller in Simulation 3. We also see that the regularized RR regression has a higher variance in the log relative risk bias than the regularized Poisson regression across all simulation settings. The median of rbrm decreases from Simulation 1 to 3, while the median of rpoisson slightly increases from Simulation 1 to 3. Prediction performance of both methods overlap and agree within the first and third quartile, but the regularized RR regression spread out more in the two tails. This is possibly due to the computational instability of the regularized RR regression on the boundary of the parameter space, which is recently discussed in [13]. Overall, the distribution of the individual-level bias of does not show evidence of strongly over- or under-estimating the log relative risk.

4.3.2 Variable Selection Evaluation

Table 4.4 summarizes the variable selection through the median number of non-zero α across the 15 replicates in each setting.

We define the True Positive Rate (TPR) as the proportion of true active covariates, in other words, covariates with non-zero coefficients, that are correctly selected by the model. The False Positive Rate (FPR) is defined as the proportion of true inactive covariates that are incorrectly selected as active by the model.

$$TPR_\alpha = \frac{\#\{i : \hat{\alpha}_i \neq 0 \wedge \alpha_i \neq 0\}}{\#\{i : \alpha_i \neq 0\}},$$

$$FPR_\alpha = \frac{\#\{i : \hat{\alpha}_i \neq 0 \wedge \alpha_i = 0\}}{\#\{i : \alpha_i = 0\}},$$

Matthew's correlation coefficient (MCC) takes into account true and false positives and negatives and is generally regarded as a balanced measure and is applicable to cases when classes are of very different sizes. It is also used here to measure the variable selection performance. The MCC can be calculated from the confusion matrix:

$$MCC_\alpha = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}},$$

TP is the number of true positives, TN the number of true negatives, FP the number of false positives and FN the number of false negatives.

Table 4.4 and Figure 4.2 shows that the regularized RR regression tends to select more covariates than the regularized Poisson regression. In the first row of Figure 4.2, while the ideal scenario includes only 10 non-zero coefficients, the rbrm method tends to select more than 10. In contrast, rpoisson typically selects fewer than 10. This same trend persists in Simulation 2, as illustrated in the second row. However, in Simulation 3, a different pattern emerges. Both rbrm and rpoisson methods select fewer than the desired 50 non-zero coefficients. Here, rbrm provides a more reasonable estimate, maintaining closer adherence to the target number of non-zero coefficients.

Overall, we see that the regularized Poisson regression tends to over-shrink the

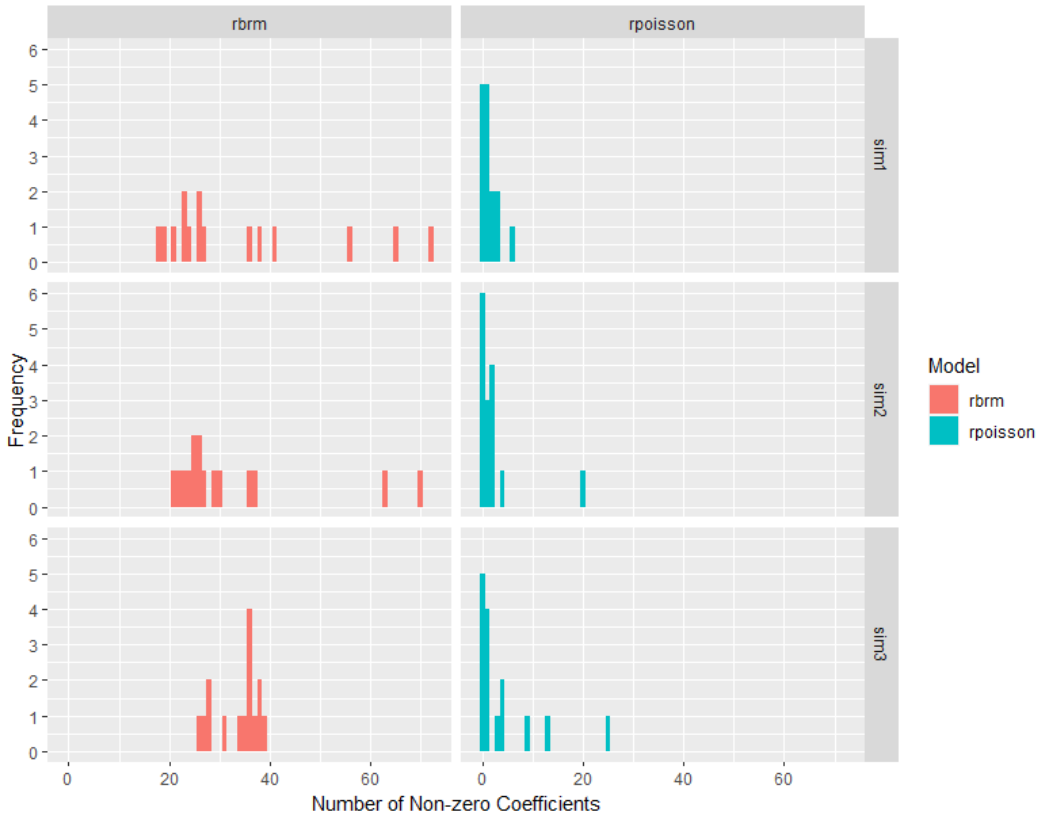

Figure 4.2: Frequency of non-zero α across the 15 replicates.

coefficients to zero, in comparison to the regularized RR regression which keeps more non-zero coefficients. This observation is further corroborated by Figure 4.3. The regularized RR regression exhibits a higher true positive rate in all settings, indicating its a better ability to correctly identify the relevant non-zero coefficients compared to the regularized Poisson regression. In the rbrm settings, the true positive rate declines from Simulation 1 to Simulation 3. This drop can be attributed to the substantial increase in the number of coefficients estimated in Simulation 3, where 1000 parameters in total are estimated, compared to 300 and 400 in Simulation 1 and 2, respectively.

However, the higher TPR in rbrm comes with a trade-off, as it also demonstrates a higher false positive rate, suggesting a tendency to erroneously identify

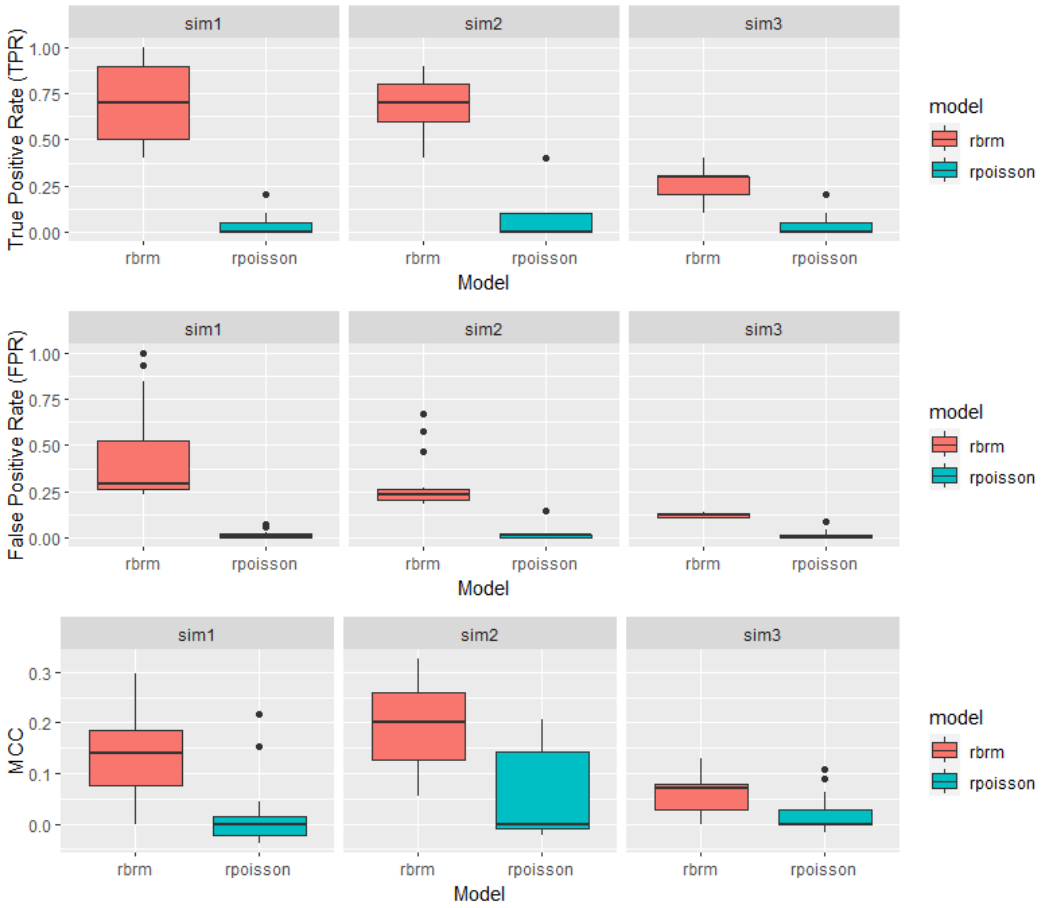

Figure 4.3: True positive rate, false positive rate, Matthew's correlation coefficient (MCC) of variable selection.

some zero coefficients as non-zero. Conversely, the regularized Poisson model manifests both lower true and false positive rates. While the lower false positive rate indicates reduced misclassification of zero coefficients, the lower true positive rate points to a limited ability to correctly identify non-zero coefficients.

This trade-off between true and false positive rates is also reflected in the Matthew's Correlation Coefficient (MCC), a measure that considers both the sensitivity and specificity of the model. Although the regularized RR regression has a slightly higher MCC, indicating a marginally better balance between the correct

and incorrect classifications, the overall range of MCC values for both models is relatively low. This suggests that the binary classification performance - correctly identifying zero and non-zero coefficients - of both methods is somewhat limited, necessitating further improvements or alternative strategies for sparse, high-dimensional data.

4.3.3 Discussion

In our analysis, we compared three models: the binary regression model (brm), the regularized Relative Risk regression (rbrm), and the regularized Poisson regression (rpoisson). We found some key differences and also some similarities between them. Notably, our method had better prediction accuracy than the brm when we measured it using the median absolute error (MAE).

When we look at the middle portion of the data, specifically the inter-quartile range, both the regularized methods seem to perform similarly. This suggests they might have the same kind of bias within this specific range. However, when we look at the far ends or tails of the distribution, we start to see differences. For instance, the regularized RR regression has results that spread out more. This might mean there is more variability in its predictions or there could be potential computational instability in these areas.

In terms of feature selection, the two regularized methods diverge. The regularized RR regression exhibits a propensity to over-select, identifying a larger number of covariates than those pointed out by the oracle, implying a possible increase in model complexity and a potential improvement in model interpretability. On the other hand, the regularized Poisson regression tends to under-select covariates, which could result in an oversimplified model that may not capture all the relevant signal in the data. These observations underscore important areas for further investigation and potential model improvement.

Chapter 5

Conclusion and Future Work

Building upon the work of Richardson et al. [14], we developed an innovative estimator, grounded in the binary regression model. To improve its efficiency and applicability, we utilized the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) for optimization. FISTA promotes sparsity in the solution and empowers variable selection, making our approach particularly suitable for high-dimensional and sparse models. Through simulation studies, we examined the properties of our proposed estimator. Our results suggest that the introduced estimator shows performance comparable to the existing regularized Poisson regression estimators in terms of predictive accuracy within the first and third quartiles of data distribution. In terms of variable selection, our approach tends to have a broader spread, suggesting its increased sensitivity to signal detection in high-dimensional settings.

However, our estimator is not without limitations. A recent study by Pozza et al. [13] demonstrated that when the log odds-product is 0 the expression 2.3 is not defined. Although Richardson et al. [14] dealt with this issue by imposing constraints in the estimation algorithm, the problem persists in our estimator. This is a significant limitation, and can be potentially addressed by using an alternative specification of the model proposed by Pozza et al. [13]. In this recent 2023 study, the authors suggest using a nuisance model expressed as ($\eta_0^A = \log[p_0(v)/(1-p_0(v)(1-p_1(v)))]$ and $\eta_1 = \log RR$, keeping the model of interest the same. The specification changes the expression of Equation 2.3 to the follow-

ing:

$$p_0 = \frac{\left[1+\exp\left(\eta_0^A\right)\left\{1+\exp\left(\eta_1\right)\right\}\right] - \sqrt{\left[1+\exp\left(\eta_0^A\right)\left\{1+\exp\left(\eta_1\right)\right\}\right]^2 - 4\exp\left(\eta_1+2\eta_{i0}^A\right)}}{2\exp\left(\eta_1+\eta_0^A\right)}.$$

The variational independence remains intact while the issue of computational instability is effectively mitigated. For future studies, it could be beneficial to investigate the performance of our estimator with this alternative specification. This approach might alleviate the boundary issues observed during optimization, potentially resulting in more accurate predictions of relative risk. This, in turn, could enhance the practicality and effectiveness of our estimator. Notably, this improvement could address the problems observed in simulation 2, particularly with respect to the unreasonable estimates of log relative risk.

While our estimator has demonstrated strong performance in certain settings, it tends to over-select variables for the model, which is an aspect that could benefit from further refinement. One potential avenue to improve this aspect of our estimator could be the incorporation of the relaxed lasso penalty. Proposed by Meinshausen [11], the Relaxed Lasso procedure starts with a Lasso process to identify significant variables, and subsequently refines the model on the selected set with the aim of achieving a less biased estimation.

However, it's crucial to note that the primary goal of the Relaxed Lasso approach is to reduce bias rather than to directly improve variable selection performance. Therefore, while the integration of the Relaxed Lasso technique with regularized RR regression could help manage high-dimensional, sparse data and provide less biased estimates of relative risks, its impact on variable selection may be indirect. As such, this approach might be an effective means to balance the bias-variance tradeoff, an aspect that deserves more detailed exploration in future research to potentially enhance the accuracy and efficiency of our relative risk estimation method.

Last but not least, our implementation of the algorithm can be further optimized and improved. Currently, the algorithm is implemented in R, which is not the most efficient language for computation. For the simulations, even with the use of parallel computing, the algorithm still takes a long time to run. In future work, we can consider implementing the algorithm in C++ or Python, which are

more efficient languages for computation. This will allow us to run the algorithm on larger datasets, and significantly improve the speed of computation which will be especially useful for the cross-validation step. Otherwise, to combat the issue of long computation time, we can consider using AIC, BIC, eBIC to select the tuning parameter instead of cross-validation because they don't require re-fitting the model on different subsets of the data. This will significantly reduce the computation time, although it may not be as accurate as cross-validation where an accurate estimate of out-of-sample error is obtained.

Bibliography

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1): 183–202, 2009. → pages 16, 17, 24

[2] J. Chen and Z. Chen. Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771, 2008. → page 14

[3] M. W. Donoghoe and I. C. Marschner. logbin: an r package for relative risk regression using the log-binomial model. *Journal of Statistical Software*, 86: 1–22, 2018. → pages 2, 6

[4] J. Friedman, T. Hastie, and N. Simon. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1): 1–22, 2010. → pages 13, 26, 31

[5] S. Greenland, J. Pearl, and J. M. Robins. Confounding and collapsibility in causal inference. *Statistical science*, 14(1):29–46, 1999. → page 5

[6] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009. → pages 13, 14

[7] H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. *Advances in neural information processing systems*, 28, 2015. → pages 18, 25

[8] Q. Li, Y. Zhou, Y. Liang, and P. K. Varshney. Convergence analysis of proximal gradient with momentum for nonconvex optimization. In *International Conference on Machine Learning*, pages 2111–2119. PMLR, 2017. → page 18

[9] T. Lumley, R. Kronmal, and S. Ma. Relative risk regression in medical research: models, contrasts, estimators, and algorithms. 2006. → page 2

[10] L.-A. McNutt, C. Wu, X. Xue, and J. P. Hafner. Estimating the relative risk in cohort studies and clinical trials of common outcomes. *American journal of epidemiology*, 157(10):940–943, 2003. → page 4

[11] N. Meinshausen. Relaxed lasso. *Computational Statistics & Data Analysis*, 52(1):374–393, 2007. → page 40

[12] J. A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965. → page 15

[13] F. Pozza, E. C. Kenne Pagui, and A. Salvan. Improved and computationally stable estimation of relative risk regression with one binary exposure. *Statistical Methods in Medical Research*, page 09622802231167436, 2023. → pages 34, 39

[14] T. S. Richardson, J. M. Robins, and L. Wang. On modeling and estimation for the relative risk and risk difference. *Journal of the American Statistical Association*, 112(519):1121–1130, 2017. → pages ix, 2, 4, 6, 7, 8, 12, 15, 18, 19, 30, 32, 39

[15] L. M. Rios and N. V. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56:1247–1293, 2013. → page 15

[16] K. J. Rothman, S. Greenland, T. L. Lash, et al. *Modern epidemiology*, volume 3. Wolters Kluwer Health/Lippincott Williams & Wilkins Philadelphia, 2008. → page 5

[17] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. → page 12

[18] R. J. Tibshirani. The lasso problem and uniqueness. *Electronic Journal of statistics*, 7:1456–1490, 2013. → page 13

[19] T. Williamson, M. Eliasziw, and G. H. Fick. Log-binomial models: exploring failed convergence. *Emerging themes in epidemiology*, 10(1): 1–10, 2013. → pages 6, 30

[20] Q. Yao, J. T. Kwok, F. Gao, W. Chen, and T.-Y. Liu. Efficient inexact proximal gradient algorithm for nonconvex problems. *arXiv preprint arXiv:1612.09069*, 2016. → page 25

[21] J. Yin, S. Markes, T. S. Richardson, and L. Wang. Multiplicative effect modelling: the general case. *Biometrika*, 109(2):559–566, 2022. → page 19

[22] G. Zou. A modified poisson regression approach to prospective studies with binary data. *American journal of epidemiology*, 159(7):702–706, 2004. → pages 2, 5

Appendix A

Supporting Materials

A.1 Pseudo-algorithm of APGnc with adaptive momentum (APGnc$^+$)

Algorithm 4 APGnc with adaptive momentum (APGnc$^+$)

Require: $\mathbf{y}_1 = \mathbf{x}_0, \beta, t \in (0,1), \eta < \frac{1}{L}$.

for $k = 1, 2, \cdots$ **do**
  $\mathbf{x}_k = \text{prox}_{\eta g}(\mathbf{y}_k - \eta \nabla f(\mathbf{y}_k))$.
  $\mathbf{v}_k = \mathbf{x}_k + \beta(\mathbf{x}_k - \mathbf{x}_{k-1})$.
  if $F(\mathbf{x}_k) \leq F(\mathbf{v}_k)$ **then**
    $\mathbf{y}_{k+1} = \mathbf{x}_k$, $\beta \leftarrow t\beta$.
  else if $F(\mathbf{v}_k) \leq F(\mathbf{x}_k)$ **then**
    $\mathbf{y}_{k+1} = \mathbf{v}_k$, $\beta \leftarrow \min\left\{\frac{\beta}{t}, 1\right\}$.
  end if
end for

A.2 Pseudo-algorithm of Calculating γ Intercept to Offset Propensity Score

Algorithm 5 Pseudo-algorithm of Calculating γ Intercept to Offset Propensity Score

Require: $M = 10000$, coefficient vector γ, diagonal covariance matrix $\Sigma = \text{diag}(\text{rep}(1, pa))$, target average probability *proportion*

Ensure: Optimal γ_0

1: Draw M samples, x_{data}, from multivariate normal with mean 0 and Σ.
2: Compute $\text{coef}_{\text{fit}} = x_{\text{data}} \cdot \gamma$.
3: **function** PROPORTIONDIFFERENCE(γ_0, coef_fit, proportion)
4: Compute transformed coefficients: $\text{prob}_{\text{test}} = \text{sigmoid}(\text{coef}_{\text{fit}} + \gamma_0)$
5: Compute the average probability: $\text{computed\_proportion} = \text{mean}(\text{prob}_{\text{test}})$
6: **return** absolute difference between computed_proportion and *proportion*
7: **end function**
8: Optimal γ_0 = argmin of PROPORTIONDIFFERENCE over γ_0
9: **return** Optimal γ_0

A.3 Estimated parameters of α in simulations 1, 2, and 3

The following figures A.1, A.2, and A.3 show the comparison of estimated parameters of α from rbrm model (red), regularized poisson model (green) and the true parameter values (blue). Notice that the estimated parameters of α from rbrm model in setting 1 replicate 4 yielded very scattered results, hinting at some numerical instability.

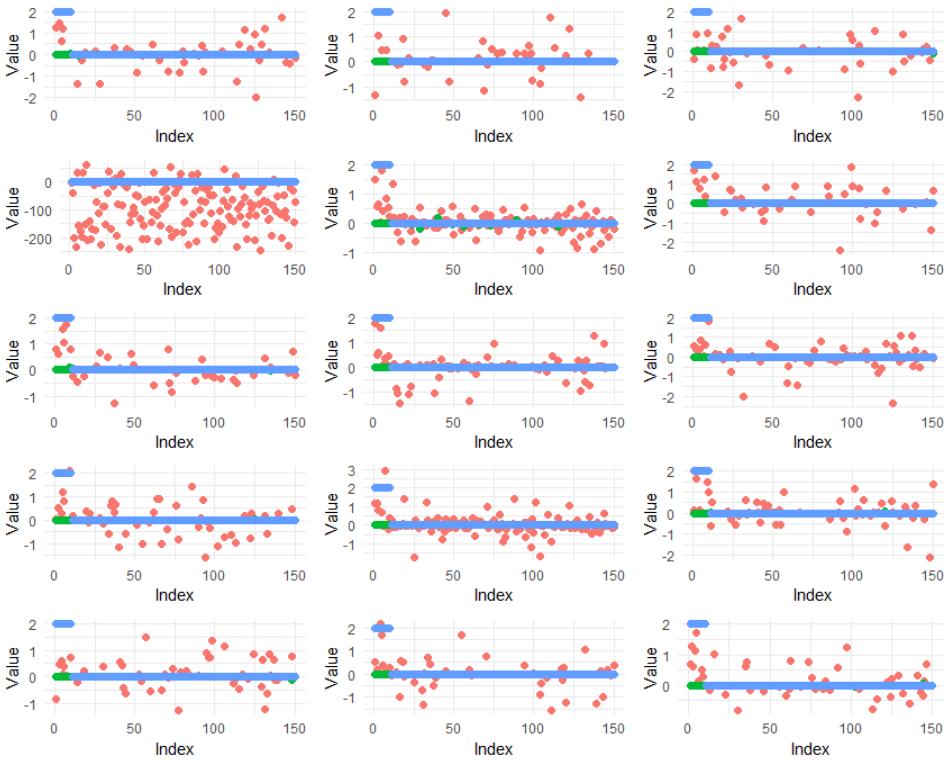

Figure A.1: Estimated parameters of α from our rbrm model (red), the regularized poisson model (green) and the true parameter values (blue) in simulation setting 1 across the 15 replicates.

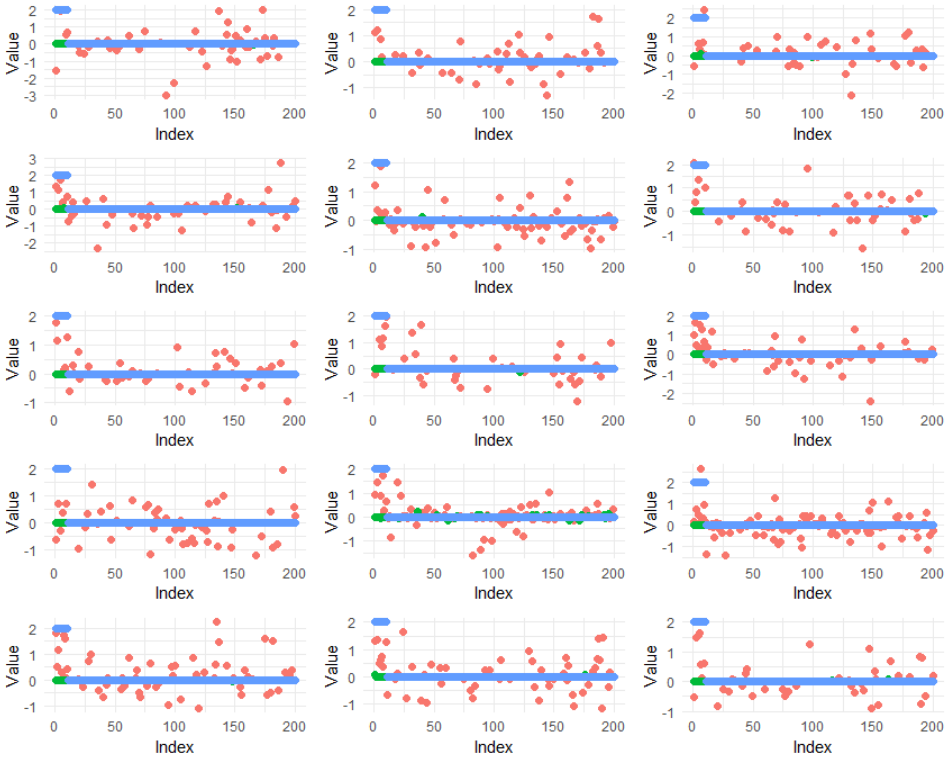

Figure A.2: Estimated parameters of α from our rbrm model (red), the regularized poisson model (green) and the true parameter values (blue) in simulation setting 2 across the 15 replicates.

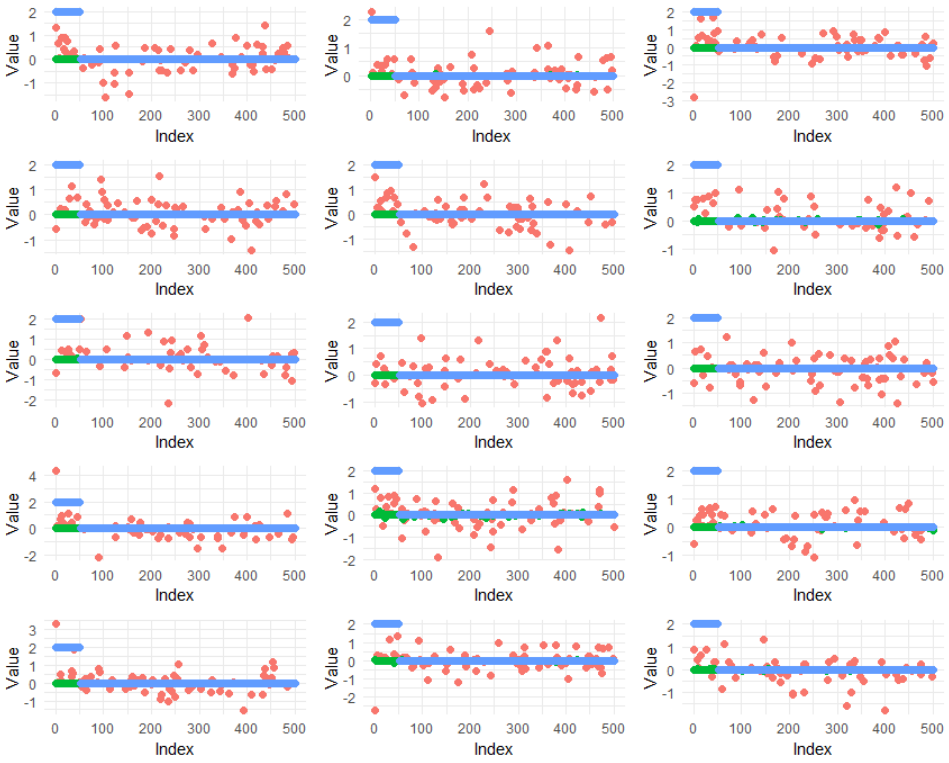

Figure A.3: Estimated parameters of α from our rbrm model (red), the regularized poisson model (green) and the true parameter values (blue) in simulation setting 3 across the 15 replicates.