Streamlined Generalized Fiducial Inference for Modern Statistical Problems

By

WEI DU
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

STATISTICS

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Thomas C.M. Lee, Chair

Prabir Burman

Jairo A Fuquene Patino

Committee in Charge

2024

To my family.

Contents

Abstract

The concept of fiducial inference was introduced by R. A. Fisher in the 1930s as a response to the limitations he perceived in Bayesian inference, notably the requirement for a subjective prior distribution on model parameters when no prior information exists. However, Fisher's initial fiducial approach lost favor due to its complexity, especially in multi-parameter settings. A resurgence of interest in the early 2000s led to the development of generalized fiducial inference (GFI), which extends Fisher's ideas and offers a promising framework for addressing a wide array of inferential challenges. Despite its potential, the adoption of GFI has been limited by its complex mathematical derivations and the need for sophisticated Markov Chain Monte Carlo (MCMC) algorithms.

This dissertation addresses these implementation challenges by proposing novel variants of GFI that simplify the sampling process and improve accessibility for researchers and practitioners. Specifically, Chapter 3 introduces AutoGFI, an intuitive algorithm that facilitates the application of GFI across diverse inference problems involving additive noise. Chapter 4 presents Fiducial Selector, a method specifically developed for high-dimensional linear regression within the GFI framework. Chapter 5 introduces AutoGFI-B and its regularized version AutoGFI-BR, which extend AutoGFI's application to binary response models, thus broadening its use from continuous to binary data. Theoretical and empirical evaluations in this work validate the effectiveness of these innovative approaches, underscoring the significant potential of GFI in modern inference challenges. Overall, this research paves the way for a more accessible and powerful application of GFI across various practical domains, substantially expanding the toolkit available for robust statistical inference.

Acknowledgments

First and foremost, I wish to express my deepest gratitude to my advisor, Professor Thomas Lee, for his continuous support and guidance throughout my PhD journey. His insightful advice and boundless enthusiasm for both research and life have profoundly inspired me, guiding me through challenging yet rewarding times. Professor Lee is not only an exceptional advisor who genuinely listens and responds to the needs of his students but also an encouraging and dependable mentor. His belief in my potential has encouraged me to reach where I am today, from a naive doctoral newcomer to now becoming an independent researcher. Beyond providing professional guidance, Professor Lee has also been a wonderful friend. He frequently organized gatherings that fostered camaraderie and brought joy into our lives. Moments spent enjoying dinners and bowling added a layer of sweetness to my PhD experience. These memories are as significant as the academic accomplishments, and I am immensely thankful for both.

I am also deeply grateful to Professor Jan Hannig for his invaluable collaboration and mentorship. His sharp intellect and profound understanding of statistics have greatly influenced my research. The breadth of his knowledge and his intuitive grasp of the field significantly alleviated the challenges I faced. The insights gleaned from discussions between Professor Lee and Professor Hannig during our regular weekly meetings have been particularly invaluable, providing clarity and direction that have substantially enhanced my academic growth.

I would like to express my heartfelt thanks to the professors who served on my qualifying exam and dissertation committees: Professor Prabir Burman, Professor Jairo A. Fuquene Patino, Professor Hao Chen, and Professor Danielle J. Harvey. I am profoundly grateful for their kind assistance, encouragement, and invaluable feedback, which have been pivotal in shaping my academic journey.

Furthermore, I extend my sincere gratitude to our department and the PhD program, which I hold in high regard. Special thanks go to the dedicated staff Sarah Driver, Andi

Carr, and Nehad Ismailwho have consistently provided patient and invaluable support with numerous issues. I also deeply appreciate all the professors whose outstanding courses have laid a solid foundation in both statistical theory and applied skills.

I would also like to thank my friends and fellow classmates in the Department of Statistics. Over the past five years, we have collaborated on numerous class projects and engaged in thought-provoking discussions about statistical problems. Beyond our academic pursuits, we have also shared countless happy moments together, from hanging out and enjoying each other's company to even watching our pets become great friends. The bonds we have forged through our shared experiences, both in and out of the classroom, have been an integral part of my PhD journey. These friendships have made the past five years not only academically enriching but also filled with cherished memories.

Last but not least, my deepest gratitude goes to my parents for their unconditional love and unwavering support. Their belief in me has been the cornerstone of my success and the driving force behind my PhD journey. I am forever indebted to them for their sacrifices, care, and wisdom that have guided me through the most challenging times. I am also incredibly fortunate to have Yutan by my side, whose encouragement and companionship have been a constant source of strength and happiness. His presence and love have been truly invaluable. A special mention goes to Texi, my lovely corgi, whose affectionate cuddles and playful antics have brought immense joy and comfort during the most stressful moments.

I dedicate this dissertation to my family, Yutan, and Texi, hoping to make them proud and express my heartfelt appreciation for their steadfast support and love.

CHAPTER 1

Introduction

Fiducial inference, originally proposed by Fisher (1930, 1933, 1935), aimed to address what he considered an issue of Bayesian inference, specifically, the use of a prior distribution when no prior information is available. Fisher introduced the concept of the fiducial distribution, a prior-free probability distribution derived directly from observed data, which could be used for inferential purposes in a manner analogous to Bayesian posterior distributions. While fiducial inference proved effective in single-parameter settings, closely aligning with classical confidence intervals, it faced challenges in multi-parameter settings. These challenges included a lack of uniqueness in the frequentist sense and ambiguity in defining the fiducial distribution. As a result, fiducial inference was marginalized by mainstream statisticians for a long time. For further details on the controversies surrounding fiducial inference, readers are referred to Zabell et al. (1992).

In spite of initial controversies, the past two decades have witnessed a significant resurgence of interest in modern adaptations of fiducial inference. These developments include Dempster-Shafer theory (Edlefsen et al., 2009) and inferential models (Martin et al., 2010; Zhang and Liu, 2011; Martin and Liu, 2013), which provide posterior probabilistic inferences about parameters without relying on priors. Another innovative approach, confidence distributions (Singh et al., 2007; Minge Xie and Strawderman, 2011; Xie and Singh, 2013), aims to establish inferentially meaningful distributions of parameters from a frequentist perspective. Additionally, objective Bayesian inference uses model-based, non-subjective priors within the Bayesian framework. Most recently, Xie and Wang (2022) have synthesized concepts from both confidence distributions and inferential models to develop an algorithmic-based inference method. A common thread among these methodologies is to obtain some inferentially meaningful probability statements about the parameter space without subjective prior information.

Despite all these efforts, generalized fiducial inference (GFI) (Hannig et al., 2016), another modification of fiducial inference, still has its edge in many areas. For example, GFI often offers good alternatives in terms of both performance and usability, e.g., the generalized fiducial distribution (GFD), which plays a similar role to the posterior distribution in the Bayesian context, is never improper. We believe that GFI and its quickly evolving variants have the potential to uncover profound and essential understandings of statistical inference.

GFI has been successfully applied across a diverse array of fields, demonstrating promising results in various applications such as wavelet regression (Hannig and Lee, 2009), ultrahigh-dimensional regression (Lai et al., 2015), binary response models (Liu and Hannig, 2016), exoplanet detection (Han and Lee, 2022), and others (McNally et al., 2003; Lidong E and Iyer, 2008). Additionally, the theoretical properties of GFI have been extensively examined using asymptotic methods, as documented in Hannig (2009); Hannig et al. (2006); Hannig (2013); Sonderegger and Hannig (2014); Majumder and Hannig (2016).

However, the practical implementation of GFI often necessitates either a complete or partial calculation of the GFD. This process requires careful mathematical derivation and may need integration with Markov Chain Monte Carlo (MCMC) techniques to generate what are known as fiducial samples from the parameter space. Such procedures can be exceedingly tedious and, in some instances, impossible, thereby reducing GFI's appeal to practitioners. Furthermore, GFI's performance can be compromised in overparameterized settings, which are prevalent in many contemporary applications.

In this dissertation, we propose three innovative approaches to address these challenges. Specifically, in Chapter 3, we introduce a novel approach named "AutoGFI," designed to simplify and enhance the application of GFI in additive noise models. Unlike traditional methods, which solve the GFD analytically and employ MCMC methods for sampling, AutoGFI generates fiducial samples directly from its definition, transforming the sampling process into a sequence of optimization problems. This algorithm significantly streamlines the implementation of GFI, facilitating its application to complex modern models, such as tensor regression and matrix completion. Subsequently, in Chapter 4, we tailor this approach for high-dimensional linear regression by incorporating a specialized de-biasing operation. We refer to this tailored approach as "Fiducial Selector". We demonstrate that Fiducial Selector achieves selection consistency and provides an unbiased estimator for the signal parameters in high-dimensional settings under certain conditions. This approach is also straightforward to implement and computationally efficient. In Chapter 5, we extend the AutoGFI framework to accommodate binary response models through reasonable approximation techniques. This extension, termed "AutoGFI-B", and its regularized version, "AutoGFI-BR", leverage the beneficial properties of the original AutoGFI algorithm and adapts them for discrete models, maintaining its effectiveness across various data types. Overall, these approaches offer robust solutions to the challenge of implementing GFI in complex practical applications, making it a more accessible, appealing, and viable option for researchers and practitioners in solving inference problems.

The remainder of this dissertation is structured as follows: Chapter 2 provides a general background on GFI. Building on this foundation, Chapter 3 introduces AutoGFI, a streamlined framework for applying GFI to additive noise models. Chapter 4 discusses Fiducial Selector, an approach inspired by the same idea as AutoGFI but specifically tailored for high-dimensional linear regression. Expanding the scope of AutoGFI, Chapter 5 extends the framework from additive noise models to binary response models with the introduction of AutoGFI-B and its regularized version, AutoGFI-BR. Chapter 6 presents the concluding remarks and discusses future prospects. Technical details and supplementary simulation results are provided in Appendix A and B.

CHAPTER 2

Background of Generalized Fiducial Inference

2.1. Definition of GFI

The development of GFI is fundamentally inspired by our understanding of Fisher's fiducial argument. We explain this argument by connecting it to the widely accepted concept of the likelihood function, as both are based on a role-switching mechanism. Recall that $f(x;\theta)$ represents the probability density function (pdf) of a random variable X, with θ as a fixed, unknown parameter, and x as a realized value. The likelihood function emerges when the roles of the random variable and the parameter are interchanged. In other words, given observed data x, the pdf $f(x;\theta)$ is reinterpreted as the likelihood function $l_x(\theta)$, which calibrates our belief in various values of θ. Similarly, fiducial inference derives a distribution for θ based purely on the observed data, mirroring this role reversal.

Building upon this idea, we now formally introduce the concept of GFI and provide a formal definition of the GFD. GFI begins with a data generating equation, expressed as

$$\boldsymbol{Y} = \boldsymbol{F}(\boldsymbol{\theta}, \boldsymbol{U}), \tag{2.1}$$

where $\boldsymbol{Y}$ denotes the observable random variable, $\boldsymbol{F}$ is a deterministic function, $\boldsymbol{\theta}$ represents the unknown parameters in the model, and $\boldsymbol{U}$ is the random component with a known distribution. The data generating equation provides a comprehensive description of how the random component $\boldsymbol{U}$ interacts with the parameter $\boldsymbol{\theta}$ to produce $\boldsymbol{Y}$. Thus, the distribution of $\boldsymbol{Y}$ can be characterized by the distribution of $\boldsymbol{U}$ and the fixed value of $\boldsymbol{\theta}$ through the data generating equation. Conversely, upon observing a specific value $\boldsymbol{y}$ of $\boldsymbol{Y}$, the roles of the random variable and the parameter can be switched. With $\boldsymbol{Y}$ fixed at $\boldsymbol{y}$, the data generating equation can be "inverted" to derive a distribution for $\boldsymbol{\theta}$. This derived distribution is termed the fiducial distribution and can be utilized to make inferences about $\boldsymbol{\theta}$.

Here, we illustrate the idea with a simple example. The data generating equation of a random variable Y from a normal distribution with unknown mean μ and variance 1 can be written as $Y = \mu + U$, where $U \sim \mathcal{N}(0, 1)$ is the random component with known distribution and $\boldsymbol{\theta} = \mu$ is the unknown parameter. After observing a value y of Y, the data generating equation can be inverted to derive $\mu = y - U$. Let U^* be an independent copy of U. The fiducial distribution of μ then follows $y - U^* \sim \mathcal{N}(y, 1)$. For further examples and a detailed discussion, interested readers can refer to Hannig et al. (2016).

Formally, for any observed data $\boldsymbol{y}$, we define the following "inverse" mapping of the data generating equation (2.1):

$$\boldsymbol{Q}_{\boldsymbol{y}}(\boldsymbol{u}) = \arg\min_{\boldsymbol{\theta}} \rho\left(\boldsymbol{F}(\boldsymbol{\theta}, \boldsymbol{u}), \boldsymbol{y}\right), \tag{2.2}$$

where ρ is a smooth semi-metric, e.g., the squared ℓ_2 norm $\rho(\boldsymbol{y}, \boldsymbol{y}^*) = \|\boldsymbol{y} - \boldsymbol{y}^*\|_2^2$. Given a realization $\boldsymbol{u}$ of $\boldsymbol{U}$, $\boldsymbol{Q}_{\boldsymbol{y}}(\boldsymbol{u})$ is a value of the parameter $\boldsymbol{\theta}$ such that $\boldsymbol{F}(\boldsymbol{\theta}, \boldsymbol{u})$ comes closest to the observed data $\boldsymbol{y}$.

Ideally, we want the minimal value of (2.2) to be 0, in other words, $\boldsymbol{Q}_{\boldsymbol{y}}(\boldsymbol{u})$ serving as an exact inverse of the data generating equation for $\boldsymbol{\theta}$ such that

$$\boldsymbol{y} = \boldsymbol{F}(\boldsymbol{Q}_{\boldsymbol{y}}(\boldsymbol{u}), \boldsymbol{u}). \tag{2.3}$$

This is also what Fisher's original fiducial argument suggests. However, there are statistical problems in which the exact inverse property (2.3) cannot be guaranteed for all $\boldsymbol{y}$ and $\boldsymbol{u}$. Nevertheless, since we assume that the data $\boldsymbol{y}$ could have been generated using (2.1), there must exist $\boldsymbol{u}^*$ for which the equality (2.3) holds. Let's denote the set of all such $\boldsymbol{u}^*$ by $\mathcal{U}_{\boldsymbol{y},0}$, i.e.,

$$\mathcal{U}_{\boldsymbol{y},0} := \{\boldsymbol{u}^* : \boldsymbol{y} = \boldsymbol{F}(\boldsymbol{Q}_{\boldsymbol{y}}(\boldsymbol{u}^*), \boldsymbol{u}^*)\}. \tag{2.4}$$

Additionally, we point out that the uniqueness of $\boldsymbol{Q}_{\boldsymbol{y}}(\boldsymbol{u})$ cannot be guaranteed for all $\boldsymbol{y}$ and $\boldsymbol{u}$. If multiple solutions $\boldsymbol{Q}_{\boldsymbol{y}}(\boldsymbol{u})$ exist, one can simply select one of them using a possibly random rule. Some guidance of such selection can be found in Hannig (2013). In fact, the uncertainty due to multiple solutions will only introduce a second-order effect on the statistical inference in many parametric problems (Hannig et al., 2016). Thus, in these parametric problems, the GFD, which is defined in the following, is not sensitive to the choice among multiple solutions as n grows.

Therefore, in principle, one can generate a fiducial sample of $\boldsymbol{\theta}$ by first generating a series of independent $\{\boldsymbol{u}^*_{(k)}\}_{k=1}^N$ from $\boldsymbol{U}$'s distribution truncated to $\mathcal{U}_{\boldsymbol{y},0}$, i.e., generating $\boldsymbol{u}^*_{(k)}$ conditional on the event that they fall into $\mathcal{U}_{\boldsymbol{y},0}$. The sample from the GFD is then any sequence of $\{\boldsymbol{\theta}^*_{(k)}\}_{k=1}^N$ satisfying $\boldsymbol{y} = \boldsymbol{F}(\boldsymbol{\theta}^*_{(1)}, \boldsymbol{u}^*_{(1)}), \ldots, \boldsymbol{y} = \boldsymbol{F}(\boldsymbol{\theta}^*_{(N)}, \boldsymbol{u}^*_{(N)})$.

However, due to Borel paradox (Casella and Berger, 2024, sec. 4.9.3), the above conditional distribution is ill-defined when $P(\boldsymbol{U} \in \mathcal{U}_{\boldsymbol{y},0}) = 0$. To address this, we enlarge this set by adding a small tolerance, defining

$$\mathcal{U}_{\boldsymbol{y},\epsilon} := \{\boldsymbol{u}^* : \rho(\boldsymbol{y}, \boldsymbol{F}(\boldsymbol{Q}_{\boldsymbol{y}}(\boldsymbol{u}^*), \boldsymbol{u}^*)) \leq \epsilon\}.$$

As this tolerance vanishes, it leads to the following limit definition of GFD (Hannig et al., 2016).

Definition 2.1.1. *Let $\boldsymbol{U}^*_\epsilon$ follows the distribution of $\boldsymbol{U}$ truncated to $\mathcal{U}_{\boldsymbol{y},\epsilon}$, i.e., having density $f_{\boldsymbol{U}}(\boldsymbol{u}) I_{\mathcal{U}_{\boldsymbol{y},\epsilon}}(\boldsymbol{u}) / P(\boldsymbol{U} \in \mathcal{U}_{\boldsymbol{y},\epsilon})$, where $f_{\boldsymbol{U}}(\boldsymbol{u})$ is the density of $\boldsymbol{U}$. Denote the distribution of $\boldsymbol{Q}_{\boldsymbol{y}}(\boldsymbol{U}^*_\epsilon)$ by μ_ϵ. If the weak limit $\lim_{\epsilon \to 0} \mu_\epsilon$ exists, the limit is called a generalized fiducial distribution (GFD).*

In practice, one could select some $\epsilon > 0$, and the approximate GFD of $\boldsymbol{\theta}$ would then be $\boldsymbol{Q}_{\boldsymbol{y}}(\boldsymbol{U}^*_\epsilon)$. Essentially, one can use the following process to generate a realization of the approximate fiducial sample: First, generate $\boldsymbol{u}^*$ from an independent copy $\boldsymbol{U}^*$ of $\boldsymbol{U}$ and optimize (2.2) to obtain the best-fitting $\boldsymbol{\theta}^*$. Then, compute $\boldsymbol{y}^* = \boldsymbol{F}(\boldsymbol{\theta}^*, \boldsymbol{u}^*)$ and compare $\boldsymbol{y}^*$ with the observed data $\boldsymbol{y}$. The candidate $\boldsymbol{\theta}^*$ is accepted if $\rho(\boldsymbol{y}, \boldsymbol{y}^*) \leq \epsilon$; otherwise, it is rejected. It is important to note that this process closely resembles Approximate Bayesian Computations (ABC, Beaumont et al., 2002). While both methods involve comparing $\boldsymbol{y}^* = \boldsymbol{F}(\boldsymbol{\theta}^*, \boldsymbol{u}^*)$ with the observed data $\boldsymbol{y}$, the main difference lies in how $\boldsymbol{\theta}^*$ is generated. ABC derives $\boldsymbol{\theta}^*$ from a prior distribution, whereas the GFD method obtains $\boldsymbol{\theta}^*$ as the solution to the optimization problem defined in (2.2).

2.2. GFI for Continuous Data

Under some smoothness assumptions, see Assumptions A.1-A.4 in Hannig et al. (2016), the limiting distribution in Definition 2.1.1 has a density

$$r_{\boldsymbol{y}}(\boldsymbol{\theta}) = \frac{f(\boldsymbol{y}, \boldsymbol{\theta}) J(\boldsymbol{y}, \boldsymbol{\theta})}{\int_\Theta f(\boldsymbol{y}, \boldsymbol{\theta}') J(\boldsymbol{y}, \boldsymbol{\theta}') d\boldsymbol{\theta}'}, \tag{2.5}$$

where $f(\boldsymbol{y}, \boldsymbol{\theta})$ is the likelihood function, and

$$J(\boldsymbol{y}, \boldsymbol{\theta}) = D\left(\nabla_{\boldsymbol{\theta}} \boldsymbol{F}(\boldsymbol{u}, \boldsymbol{\theta}) \Big|_{\boldsymbol{u}=\boldsymbol{F}^{-1}(\boldsymbol{y},\boldsymbol{\theta})} \right). \tag{2.6}$$

When using squared ℓ_2 norm as ρ in (2.2), it is showed in Hannig et al. (2016) that $D(\nabla_{\boldsymbol{\theta}} \boldsymbol{F}) = (\det \nabla_{\boldsymbol{\theta}} \boldsymbol{F}^\top \nabla_{\boldsymbol{\theta}} \boldsymbol{F})^{1/2}$. Also, $\boldsymbol{u} = \boldsymbol{F}^{-1}(\boldsymbol{y}, \boldsymbol{\theta})$ is the value of $\boldsymbol{u}$ such that $\boldsymbol{y} = \boldsymbol{F}(\boldsymbol{\theta}, \boldsymbol{u})$.

Equations (2.5) and (2.6) present an interesting and intriguing connection between GFI and Bayesian methodology: the density $r_{\boldsymbol{y}}(\boldsymbol{\theta})$ in (2.5) behaves like a posterior density in the Bayesian context with $J(\boldsymbol{y}, \boldsymbol{\theta})$ being the "prior", except that the data $\boldsymbol{y}$ also appear in $J(\boldsymbol{y}, \boldsymbol{\theta})$, so strictly speaking it is not a prior density. Note also that $J(\boldsymbol{y}, \boldsymbol{\theta})$ shares the invariance to reparametrization property with the Jeffreys prior.

In practice, when using (2.5) and (2.6) for GFI applications, typically there are three possibilities:

(1) A closed-form expression for $r_{\boldsymbol{y}}(\boldsymbol{\theta})$ can be obtained,

(2) $r_{\boldsymbol{y}}(\boldsymbol{\theta})$ is known up to a normalizing constant.

(3) The term $J(\boldsymbol{y}, \boldsymbol{\theta})$ cannot be analytically calculated so (2.5) cannot be applied.

The first possibility often happens only for simple problems where alternative inference solutions are known. For the second possibility, MCMC methods are required to generate a fiducial sample from $r_{\boldsymbol{y}}(\boldsymbol{\theta})$, which could be computationally demanding. More complex problems fall into the third category, where working with $J(\boldsymbol{y}, \boldsymbol{\theta})$ is not straightforward.

In addition, recall that the formula (2.5) was derived for scenarios with low dimensionality and, as such, cannot be directly applied to the high-dimensional context. To accommodate for the challenges introduced by high dimensionality, model selection is required. As noted by Hannig and Lee (2009), this can be achieved by adding extra penalty structural equations to (2.5). Interesting readers can refer to Hannig and Lee (2009); Hannig et al. (2016) for more details about model selection in GFI. However, this method involves a complex sampling process to emulate the GFD.

Therefore, the closed form of the GFD is not always practical. To make GFI more widely applicable, a more user-friendly approach is necessary.

2.3. GFI for Discrete Data

The closed form (2.5) of the GFD is derived for the case when the observations following continuous distributions. It can no longer be used when working with discrete distributions. When the observations are discrete, there is no problem with the Borel paradox, i.e., $P(\boldsymbol{U} \in \mathcal{U}_{\boldsymbol{y},0}) > 0$. Therefore, one must find the exact inverse of the data generating equation to get the GFD defined in (2.1.1).

Note that the minimizer $\boldsymbol{Q}_{\boldsymbol{y}}(\boldsymbol{u})$ in (2.2) for $\boldsymbol{u} \in \mathcal{U}_{\boldsymbol{y},0}$ is not unique; it is actually a set in the discrete model context. Therefore, a selection rule is needed. Let $V[A]$ denote the rule for selecting a (possibly random) element from the closure of set A. The GFD of $\boldsymbol{\theta}$ is then defined as the distribution of

$$V[\boldsymbol{Q}_{\boldsymbol{y}}(\boldsymbol{U}_0^*)], \tag{2.7}$$

where $\boldsymbol{U}_0^*$ has the density $f_{\boldsymbol{U}}(\boldsymbol{u}) I_{\mathcal{U}_{\boldsymbol{y},0}}(\boldsymbol{u}) / P(\boldsymbol{U} \in \mathcal{U}_{\boldsymbol{y},0})$, i.e., it follows the distribution of $\boldsymbol{U}$ truncated to $\mathcal{U}_{\boldsymbol{y},0}$ as defined in (2.4). If $A = (a, b)$ is a finite interval, Hannig (2009) recommends using a "half-correction" rule for $V[A]$, which selects one of the endpoints a or b randomly and independently of $\boldsymbol{U}_0^*$. Such a selection maximizes the variance of the GFD. More details can be found in Hannig (2009).

For some common discrete models, the exact inverse $\boldsymbol{Q}_{\boldsymbol{y}}(\boldsymbol{u})$ can be computed straightforwardly; see examples in Hannig et al. (2016); Murph et al. (2023). However, for more complex models, deriving the exact inverse set of the data generating equation and sampling from it can be challenging, often involving the solution of multiple inequalities and the use of MCMC techniques. Therefore, this approach is not always feasible. Applying GFI to more complex discrete models requires a simplified sampling process.

CHAPTER 3

Streamlined GFI for Additive Noise Models

In this chapter, we focus on applying GFI to the additive noise model, which has the data generating equation of the following form:

$$\boldsymbol{Y} = \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{\theta}) + \boldsymbol{U}, \tag{3.1}$$

where $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^\top$ denotes the observable variables, $\boldsymbol{X}$ represents the feature information, $\boldsymbol{\theta}$ is the unknown parameter, and $\boldsymbol{U} = (U_1, \ldots, U_n)^\top$ denotes the noise vector, following a completely known distribution. Many popular statistical and machine learning models fit this framework, for example, linear regression, tensor regression (Zhou et al., 2013), noise matrix completion (Candès and Recht, 2009), and regression incorporating network cohesion (Li et al., 2019). When the structural function $\mathbf{G}(\boldsymbol{X}, \boldsymbol{\theta})$ is complex, computing the GFD (2.6) can be highly challenging or even impractical, limiting the application of GFI for these contemporary models. Therefore, this chapter introduces a method, termed AutoGFI, to simplify the implementation of GFI in such additive noise models, thereby broadening its applicability.

The remainder of this chapter is structured as follows: Sections 3.1 and 3.2 introduce the basic form of AutoGFI, as well as its regularized and debiased counterparts. The theoretical properties of AutoGFI are then explored in Section 3.3, followed by application examples to tensor regression in Section 3.4, matrix completion in Section 3.5, and regression with network cohesion in Section 3.6 to demonstrate its wide applicability and excellent empirical performance. Concluding remarks are presented in Section 3.7, while technical details are deferred to Appendix A.

3.1. AutoGFI for Additive Noise Models

First, we examine the algorithm described at the end of Section 2.1, which generates approximate fiducial samples for the model (3.1):

(1) Generate an independent copy $\boldsymbol{U}^*$ of $\boldsymbol{U}$.

(2) Solve for $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \rho(\boldsymbol{Y}, \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{\theta}) + \boldsymbol{U}^*)$.

(3) Accept $\boldsymbol{\theta}^*$ if $\rho(\boldsymbol{Y}, \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{\theta}^*) + \boldsymbol{U}^*) \leq \epsilon$; otherwise, reject and return to Step (1).

This algorithm can be viewed as a fiducial version of Approximate Bayesian Computation (ABC) (Beaumont et al., 2002). We assume $\boldsymbol{U}$ follows a fully known distribution, and a commonly used choice for that would be the normal distribution, e.g., $U_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma)$. However, σ is often not available in practice. In such cases, it is advisable for computational reasons to replace σ with a consistent plug-in estimator. ϵ can be set to $\varepsilon\delta_0$, where $\delta_0 = \min_{\boldsymbol{\theta}} \rho(\boldsymbol{Y}, \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{\theta}))$ and $0 < \varepsilon < 1$. For some simple models, such as linear regression without regularization, the optimization problem in Step (2) can be solved in closed form, making the algorithm very easy to implement.

Building upon the basic GFI algorithm, we propose several generalizations to extend its applicability to more complex models. These generalizations include the introduction of a penalty term and a de-biasing operation. First, in Step (2) of the algorithm, we add a penalty term $\Lambda(\boldsymbol{\theta})$ to introduce regularization or shrinkage. This addition is particularly useful for handling model selection problems. Next, when a penalty is applied, we propose including a de-biasing operation $\boldsymbol{d}(\boldsymbol{\theta})$ to counteract the bias introduced by the penalty. A general approach to finding such a de-biasing function will be provided in Section 3.2. The resulting easy-to-use AutoGFI algorithm, which incorporates these generalizations, is summarized in Algorithm 1.

The proposed AutoGFI algorithm is versatile and can be applied to a wide range of modern applications as long as there is a fitting procedure for the parameters, i.e., Step (2) can be executed. In the following sections, we will use AutoGFI to perform the uncertainty

Algorithm 1 AutoGFI: Generating fiducial sample for additive noise models.

```
Input: Data $(\boldsymbol{X}, \boldsymbol{Y})$; penalty parameter of $\Lambda(\boldsymbol{\theta})$; tolerance $\epsilon$
Output: De-biased fiducial sample of $\boldsymbol{\theta}$: $\boldsymbol{\theta}^*_{\text{de}}$
(1) Generate an independent copy $\boldsymbol{U}^*$ of $\boldsymbol{U}$.
(2) Solve
        $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \rho(\boldsymbol{Y}, \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{\theta}) + \boldsymbol{U}^*) + \Lambda(\boldsymbol{\theta}).$
(3) Debias $\boldsymbol{\theta}^*$ with $\boldsymbol{\theta}^*_{\text{de}} = \boldsymbol{d}(\boldsymbol{\theta}^*)$.
(4) Accept $\boldsymbol{\theta}^*_{\text{de}}$ if $\rho(\boldsymbol{Y}, \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{\theta}^*_{\text{de}}) + \boldsymbol{U}^*) \leq \epsilon$; otherwise, reject and return to Step (1).
```

quantification task in three different problems. Through a series of simulation studies, we will showcase its exceptional empirical performance.

In the end of this section, we compare AutoGFI to the celebrated bootstrap (Efron and Tibshirani, 1994), highlighting their distinctions and similarities. Non-parametric bootstrap is based on the "re-sampling" idea, for example, re-sampling pairs in regression setup. In contrast, parametric bootstrap must rely on an "initial" fitted model, and hence the inference made is sensitive to the initial fitting. On the other hand, the key of AutoGFI is re-sampling the random component $\boldsymbol{U}$ acting similarly to a "pivot" and then refitting the perturbed data as in Step (2) of the algorithm. Additionally, AutoGFI incorporates a rejection step to ensure that the data is perturbed in a manner considered reasonable. It is worth mentioning that although one might expect this step to increase computational demands, in practice, AutoGFI maintains a computation cost comparable to that of the bootstrap methods. Theoretical results, presented in Section 3.3, indicate that the inference remains valid even if the rejection step (Step (4)) is omitted. This allows us to relax the choice of ϵ to speed up the calculation. Consequently, we choose a relatively large ϵ and exclude only samples that appear to be outliers. As a result, the effect of the rejection step on the computational efficiency of AutoGFI is minimal.

3.2. Debiasing for AutoGFI

As hinted before, if a penalty term is added in Step (2), the AutoGFI estimates necessarily suffer from non-negligible bias. Inspired by the idea proposed in van de Geer et al. (2014)

and Janková and van de Geer (2018), this section provides a general approach to correct such estimation bias.

Under differentiability conditions, a potential fiducial sample generated by Step (2) can often be viewed as solving the estimating equation

$$\nabla_{\boldsymbol{\theta}} \rho(\boldsymbol{Y}, \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{\theta}) + \boldsymbol{U}^*)|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} + \boldsymbol{\xi}(\boldsymbol{\theta}^*) = 0, \tag{3.2}$$

where $\boldsymbol{\xi}(\boldsymbol{\theta}^*)$ is a (sub-)gradient of the penalty term $\Lambda(\boldsymbol{\theta})$ evaluated at $\boldsymbol{\theta}^*$. The idea of removing the bias associated with the penalty is to modify the fiducial sample $\boldsymbol{\theta}^*$ such that the first term of (3.2) is closer to zero. To achieve this, we implement a one-step modification and define the de-biased fiducial sample $\boldsymbol{\theta}^*_{\text{de}}$ by solving the equation

$$-\boldsymbol{\xi}(\boldsymbol{\theta}^*) + \mathbf{H}(\boldsymbol{\theta}^*)(\boldsymbol{\theta}^*_{\text{de}} - \boldsymbol{\theta}^*) = \mathbf{0}, \tag{3.3}$$

where $\mathbf{H}(\boldsymbol{\theta}^*)$ represents the Hessian matrix of second partial derivatives of $\rho(\boldsymbol{Y}, \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{\theta}) + \boldsymbol{U}^*)$ with respect to $\boldsymbol{\theta}$, evaluated at $\boldsymbol{\theta}^*$. When the j-th coordinate of the sub-gradient, $\xi(\boldsymbol{\theta}^*)_j$, is an interval, we recommend using $-\xi(\boldsymbol{\theta}^*)_j = \nabla_{\boldsymbol{\theta}} \rho(\boldsymbol{Y}, \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{\theta}) + \boldsymbol{U}^*)_j|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}$ in (3.3), unless the sub-gradient interval includes zero. If zero is included, we do not de-bias the j-th coordinate of the fiducial sample; instead, we set $\theta^*_{j,\text{de}} = \theta^*_j$ and exclude the corresponding coordinates in (3.3), also removing the relevant rows and columns from the matrix $\mathbf{H}(\boldsymbol{\theta}^*)$. This approach ensures that our de-biasing process does not interfere with model selection. For instance, if an ℓ_1 penalty shrinks some coefficients to zero, these coefficients will remain zero.

In high-dimensional settings, the matrix $\mathbf{H}(\boldsymbol{\theta}^*)$ is often rank-deficient and poorly conditioned. As a consequence, the solution to (3.3) can be numerically unstable and highly variable. To manage this variability, we employ two strategies. First, when penalty-induced sparsity is utilized, as mentioned previously, we treat Step (2) as a model selection step and perform de-biasing only for the non-zero coordinates where $\boldsymbol{\xi}(\boldsymbol{\theta}^*)$ does not form an interval containing zero. Second, to further reduce variability, we use the pseudo-inverse $\mathbf{H}(\boldsymbol{\theta}^*)^{\text{pinv}}$ in place of the actual inverse $\mathbf{H}(\boldsymbol{\theta}^*)^{-1}$ when solving the equation (3.3).

For a square matrix $\mathbf{H} \in \mathbb{R}^{n\times n}$ with singular value decomposition as $\mathbf{H} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{W}^\top$, where $\boldsymbol{\Sigma}$ is the diagonal matrix containing all non-zero singular values, denote by $\zeta_i(\mathbf{H})$ the i-th largest singular value of $\mathbf{H}$. Let $\mathbf{S}$ be the diagonal matrix containing all the singular values greater than $c\,\zeta_1(\mathbf{H})$ for some threshold constant c. We define the pseudo inverse of $\mathbf{H}$ as

$$\mathbf{H}^{\text{pinv}} := \mathbf{W} \begin{pmatrix} \mathbf{S}^{-1} & 0 \\ 0 & 0 \end{pmatrix} \mathbf{V}^\top.$$

In other words, we only use those singular vectors corresponding to significant singular values of $\mathbf{H}(\boldsymbol{\theta}^*)$ to perform de-biasing. The threshold c can be a small constant or determined in a data-dependent manner. For example, c can be chosen as any value between $\zeta_i(\mathbf{H})/\zeta_1(\mathbf{H})$ and $\zeta_{i+1}(\mathbf{H})/\zeta_1(\mathbf{H})$ where $i = \arg\max_k \zeta_k(\mathbf{H})/\zeta_{k+1}(\mathbf{H})$. This implies that c is located at the point where there is a large jump in the magnitude of the singular values of $\mathbf{H}$. With these, the de-biasing function is defined as

$$\boldsymbol{\theta}^*_{\text{de}} := \boldsymbol{\theta}^* + \mathbf{H}(\boldsymbol{\theta}^*)^{\text{pinv}}\boldsymbol{\xi}(\boldsymbol{\theta}^*). \tag{3.4}$$

The remaining problem is calculating the first and second derivatives of ρ, which could be quite challenging for complex models. To tackle this, we advocate using a set of new techniques in mathematical computation called *Automatic Differentiation* (Autodiff). Autodiff provides a robust framework for evaluating derivatives of functions formulated as computer programs. At its essence, Autodiff takes advantage of the fact that every computer program, regardless of complexity, ultimately breaks down into a sequence of elementary arithmetic operations (such as addition and subtraction) and basic functions (like exponential, logarithmic, sine, and cosine functions). Similar to backpropagation in neural networks, Autodiff applies the chain rule systematically to these operations, enabling the computation of derivatives at any desired order. The precision of these derivatives matches the computer's operational accuracy, and the technique requires only a modest increase in computational effort compared to the original program execution. Therefore, with such techniques, we can accurately and efficiently evaluate the first and second derivatives of ρ even for complex models. Numerous software packages have been developed to implement Autodiff in fields such as machine learning and scientific computing. In our work, we use the `Python` package `JAX` (Frostig et al., 2018) to carry out the calculations.

3.3. Theoretical Properties

This section presents some theoretical properties of AutoGFI. The proofs are delayed to Appendix A.

Let $Y_i = G(\boldsymbol{X}_i, \boldsymbol{\theta}_0) + U_i$ for all $i = 1, \ldots, n$ and $\boldsymbol{U}^*$ be an independent copy of $\boldsymbol{U} = (U_1, \ldots, U_n)^\top$, i.e., $\boldsymbol{U}$ and $\boldsymbol{U}^*$ are i.i.d.. Assume $\rho(\boldsymbol{y}, \boldsymbol{y}^*) = \|\boldsymbol{y} - \boldsymbol{y}^*\|^2/2 = \sum_{i=1}^n (y_i - y_i^*)^2/2$, the gradient of the penalty function $\boldsymbol{\xi}_n(\boldsymbol{\theta})$ is a monotone increasing and differentiable function, and the data generating function $\boldsymbol{G}(\boldsymbol{X}, \boldsymbol{\theta})$ is twice continuously differentiable in $\boldsymbol{\theta}$. Consequently (3.2) becomes

$$\sum_{i=1}^n -\nabla_{\boldsymbol{\theta}} G(\boldsymbol{X}_i, \boldsymbol{\theta})(Y_i - G(\boldsymbol{X}_i, \boldsymbol{\theta}) - U_i^*) + \boldsymbol{\xi}_n(\boldsymbol{\theta}) = 0.$$

Denote by $\hat{\boldsymbol{\theta}}$ the solution of

$$\sum_{i=1}^n -\nabla_{\boldsymbol{\theta}} G(\boldsymbol{X}_i, \hat{\boldsymbol{\theta}})(Y_i - G(\boldsymbol{X}_i, \hat{\boldsymbol{\theta}})) + \boldsymbol{\xi}_n(\hat{\boldsymbol{\theta}}) = 0, \tag{3.5}$$

by $\boldsymbol{\theta}^*$ the solution of

$$\sum_{i=1}^n -\nabla_{\boldsymbol{\theta}} G(\boldsymbol{X}_i, \boldsymbol{\theta}^*)(Y_i - G(\boldsymbol{X}_i, \boldsymbol{\theta}^*) - U_i^*) + \boldsymbol{\xi}_n(\boldsymbol{\theta}^*) = 0,$$

and by $\hat{\boldsymbol{\theta}}_0$ the solution of

$$\sum_{i=1}^n -\nabla_{\boldsymbol{\theta}} G(\boldsymbol{X}_i, \hat{\boldsymbol{\theta}}_0)(G(\boldsymbol{X}_i, \boldsymbol{\theta}_0) - G(\boldsymbol{X}_i, \hat{\boldsymbol{\theta}}_0)) + \boldsymbol{\xi}_n(\hat{\boldsymbol{\theta}}_0) = 0. \tag{3.6}$$

We assume the following:

Assumption 3.3.1. *There exists a compact $K \subset \Theta$ so that $\boldsymbol{\theta}_0 \in K^o$ the interior of K.*

(a) The probability that there exist $\hat{\boldsymbol{\theta}}_0 \in K$, $\hat{\boldsymbol{\theta}} \in K$, and $\boldsymbol{\theta}^* \in K$ converges to 1.

(b) There exists a monotone increasing function h so that $h(0) = 0$ and for all $\boldsymbol{\theta} \in K$

$$P\Big(\|n^{-1}\sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} G(\boldsymbol{X}_i, \boldsymbol{\theta})(G(\boldsymbol{X}_i, \boldsymbol{\theta}_0) - G(\boldsymbol{X}_i, \boldsymbol{\theta}))\| \geq h(\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|)\Big) \to 1.$$

(c) There exists a full rank, continuous covariance matrix $\boldsymbol{S}(\boldsymbol{\theta})$ so that

$$n^{-1/2} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} G(\boldsymbol{X}_i, \boldsymbol{\theta}) U_i \xrightarrow{\mathcal{D}} N(\mathbf{0}, \boldsymbol{S}(\boldsymbol{\theta}))$$

uniformly in $\boldsymbol{\theta} \in K$.

(d) The penalty satisfies $n^{-1} \sup_{\boldsymbol{\theta} \in K} \boldsymbol{\xi}_n(\boldsymbol{\theta}) \to 0$.

Theorem 3.3.2. *Under Assumption 3.3.1, $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\| \xrightarrow{P} 0$ and $\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_0\| \xrightarrow{P} 0$.*

Assumption 3.3.3. *Consider the Taylor series approximation at $\hat{\boldsymbol{\theta}}_0$*

$$n^{-1} \left(\sum_{i=1}^{n} -\nabla_{\boldsymbol{\theta}} G(\boldsymbol{X}_i, \boldsymbol{\theta})(G(\boldsymbol{X}_i, \boldsymbol{\theta}_0) - G(\boldsymbol{X}_i, \boldsymbol{\theta})) + \boldsymbol{\xi}_n(\boldsymbol{\theta}) \right) = \boldsymbol{T}_n(\hat{\boldsymbol{\theta}}_0)(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_0) + \boldsymbol{R}_n(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_0). \tag{3.7}$$

Assume

(a) $\boldsymbol{T}_n(\hat{\boldsymbol{\theta}}_0) \xrightarrow{P} \boldsymbol{T}_\infty$, where $\boldsymbol{T}_\infty$ is invertible.

(b) There is a continuous function R, so that $R(0) = 0$, and $\epsilon > 0$ so that for all $\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_0\| \leq \epsilon$,

$$P\left(\|\boldsymbol{R}_n(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_0)\| \leq \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_0\| R(\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_0\|)\right) \to 1.$$

Theorem 3.3.4. *Under the Assumptions 3.3.1 and 3.3.3,*

$$n^{1/2}(\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}_0) \xrightarrow{\mathcal{D}} N(\mathbf{0}, \boldsymbol{T}_\infty^{-1} \mathbf{S}(\boldsymbol{\theta}_0) \boldsymbol{T}_\infty^{-1\top}),$$

$$n^{1/2}(\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}) \xrightarrow{\mathcal{D}} N(\mathbf{0}, \boldsymbol{T}_\infty^{-1}\mathbf{S}(\boldsymbol{\theta}_0)\boldsymbol{T}_\infty^{-1\top}).$$

Notice that if $n^{-1/2}\boldsymbol{\xi}_n(\boldsymbol{\theta}) \to \mathbf{0}$, Theorem 3.3.4 implies that confidence intervals based on the generalized fiducial distribution will be asymptotically correct. Otherwise, recall the debiasing procedure (3.4) and notice that under assumptions of this section

$$\mathbf{H}(\boldsymbol{\theta}^*) = \sum_{i=1}^{n} \left[-(\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}}^\top G(\boldsymbol{X}_i, \boldsymbol{\theta}^*))(Y_i - G(\boldsymbol{X}_i, \boldsymbol{\theta}^*) - U_i^*) + \nabla_{\boldsymbol{\theta}} G(\boldsymbol{X}_i, \boldsymbol{\theta}^*)\nabla_{\boldsymbol{\theta}}^\top G(\boldsymbol{X}_i, \boldsymbol{\theta}^*)\right].$$

Next define

$$\hat{\mathbf{H}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \left[-(\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}}^\top G(\boldsymbol{X}_i, \boldsymbol{\theta}))(Y_i - G(\boldsymbol{X}_i, \boldsymbol{\theta})) + \nabla_{\boldsymbol{\theta}} G(\boldsymbol{X}_i, \boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}^\top G(\boldsymbol{X}_i, \boldsymbol{\theta})\right],$$

$$\mathbf{H}_0(\boldsymbol{\theta}) = \sum_{i=1}^{n} \left[-(\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}}^\top G(\boldsymbol{X}_i, \boldsymbol{\theta}))(G(\boldsymbol{X}_i, \boldsymbol{\theta}_0) - G(\boldsymbol{X}_i, \boldsymbol{\theta})) + \nabla_{\boldsymbol{\theta}} G(\boldsymbol{X}_i, \boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}^\top G(\boldsymbol{X}_i, \boldsymbol{\theta})\right],$$

and set

$$\hat{\boldsymbol{\theta}}_{\text{de}} = \hat{\boldsymbol{\theta}} + \hat{\mathbf{H}}(\hat{\boldsymbol{\theta}})^{\text{pinv}}\boldsymbol{\xi}_n(\hat{\boldsymbol{\theta}}), \quad \hat{\boldsymbol{\theta}}_{0,\text{de}} = \hat{\boldsymbol{\theta}}_0 + \mathbf{H}_0(\hat{\boldsymbol{\theta}}_0)^{\text{pinv}}\boldsymbol{\xi}_n(\hat{\boldsymbol{\theta}}_0).$$

Assumption 3.3.5. *The de-biasing procedure satisfies*

(a) $n^{1/2}(\mathbf{H}(\boldsymbol{\theta}^*)^{pinv} - \mathbf{H}_0(\boldsymbol{\theta}^*)^{pinv})\boldsymbol{\xi}_n(\boldsymbol{\theta}^*) \xrightarrow{P} \mathbf{0}$ *and* $n^{1/2}(\hat{\mathbf{H}}(\hat{\boldsymbol{\theta}})^{pinv} - \mathbf{H}_0(\hat{\boldsymbol{\theta}})^{pinv})\boldsymbol{\xi}_n(\hat{\boldsymbol{\theta}}) \xrightarrow{P} \mathbf{0}$.

(b) There exist $C_n \xrightarrow{P} 0$ *so that*

$$\|\mathbf{H}_0(\boldsymbol{\theta}_1)^{pinv}\boldsymbol{\xi}_n(\boldsymbol{\theta}_1) - \mathbf{H}_0(\boldsymbol{\theta}_2)^{pinv}\boldsymbol{\xi}_n(\boldsymbol{\theta}_2)\| \le C_n\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|$$

for all $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in K$.

(c) $R_n = \sup_{\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\| \le \|\hat{\boldsymbol{\theta}}_0 - \boldsymbol{\theta}_0\|} \|\mathbf{I} - \mathbf{H}_0(\hat{\boldsymbol{\theta}}_0)^{pinv}\mathbf{H}_0(\boldsymbol{\theta})\|_2 \xrightarrow{P} 0.$

The following theorem shows that under our assumptions the de-biasing procedure reduces bias without affecting the asymptotic normality of the fiducial samples.

Theorem 3.3.6. *Under the Assumptions 3.3.1, 3.3.3, 3.3.5*

$$n^{1/2}(\hat{\boldsymbol{\theta}}_{de} - \hat{\boldsymbol{\theta}}_{0,de}) \xrightarrow{\mathcal{D}} N(\mathbf{0}, \boldsymbol{T}_\infty^{-1}\mathbf{S}(\boldsymbol{\theta}_0)\boldsymbol{T}_\infty^{-1\top}),$$

$$n^{1/2}(\boldsymbol{\theta}^*_{de} - \hat{\boldsymbol{\theta}}_{de}) \xrightarrow{\mathcal{D}} N(\mathbf{0}, \boldsymbol{T}_\infty^{-1}\mathbf{S}(\boldsymbol{\theta}_0)\boldsymbol{T}_\infty^{-1\top}),$$

and

$$\hat{\boldsymbol{\theta}}_{0,de} - \boldsymbol{\theta}_0 = o_P(\hat{\boldsymbol{\theta}}_0 - \boldsymbol{\theta}_0).$$

3.4. Tensor Regression

In this section, we introduce tensor regression as the first example of a complex modern model. By applying AutoGFI to this model, we aim to evaluate its performance in parameter estimation and uncertainty quantification within such a complex model structure.

3.4.1. Background. In the field of biomedical sciences, technological advances have led to the generation of vast multi-way array data in areas such as genomics (Tao et al., 2017) and medical imaging (Zhou et al., 2013). These multi-way array data can be naturally represented as tensors. For instance, gene-gene and protein-protein interaction networks can be expressed as second-order tensors (i.e., 2D adjacency matrices), while anatomical MRI scans can be represented as third-order tensors. Understanding the relationships between these complex tensor data and clinical outcomes is crucial for uncovering the biological mechanisms underlying various diseases. To address this challenge, tensor regression models are employed, linking clinical outcomes, represented as continuous criterion variables, with multi-way data serving as predictors. While tensor regression models have proven particularly valuable in biomedical research, their utility extends far beyond this field. These models can handle a tensor predictor and a real-valued response, making them suitable for a wide range of applications where high-dimensional data need to be correlated with quantitative outcomes.

Tensor decomposition is a useful method for exploring the low-rank structure of a tensor, as the major component is often governed by a small number of latent factors (Kolda and Bader, 2009; Shang et al., 2014). Several tensor regression models have been proposed based on tensor decomposition, such as the CANDECOMP/PARAFAC and Tucker decompositions. While many studies have focused on estimating the tensor coefficient and selecting effective regions (Guo et al., 2011; Li et al., 2018; Ou-Yang et al., 2020; Zhou et al., 2013), few have addressed quantifying the uncertainty of estimates. Notable exceptions include the Bayesian approaches introduced in Guhaniyogi et al. (2017) and Papadogeorgou et al. (2021).

This section applies AutoGFI to the tensor regression model and compares it with the two Bayesian methods mentioned above. Results from simulation experiments show that AutoGFI can provide a robust estimate of the tensor coefficient and, at the same time, can offer preferable uncertainty quantification.

3.4.2. Problem Definition. Let $Y_i \in \mathbb{R}$ be a response variable and $\boldsymbol{X}_i \in \otimes_{d=1}^{D} \mathbb{R}^{p_d}$ be a tensor predictor of order D for $i = 1, \ldots, n$. We consider the Gaussian linear model for tensor regression, given by

$$Y_i = \langle \boldsymbol{X}_i, \boldsymbol{B} \rangle + U_i, \quad U_i \overset{iid}{\sim} N(0, \sigma^2), \tag{3.8}$$

where $\boldsymbol{B}$ is the tensor coefficient, a D-mode tensor with $\prod_{d=1}^{D} p_d$ unknown parameters. The inner product of two tensors, denoted by $\langle \boldsymbol{X}_i, \boldsymbol{B} \rangle$, is defined as $\mathrm{vec}(\boldsymbol{X}_i)^\top \mathrm{vec}(\boldsymbol{B})$, where $\mathrm{vec}(\boldsymbol{X}_i)$ represents the vectorization of $\boldsymbol{X}_i$. Our objective is to provide robust estimation and uncertainty quantification for the tensor coefficient $\boldsymbol{B}$.

As the dimensionality of $\boldsymbol{B}$ usually exceeds the sample size, low-rank approximation and regularization techniques are needed to reduce the number of parameters. We follow Zhou et al. (2013) and assume a rank-R CP decomposition of the tensor coefficient, i.e. $\boldsymbol{B} = \sum_{r=1}^{R} \boldsymbol{\beta}_1^{(r)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r)}$, where $\boldsymbol{\beta}_d^{(r)} \in \mathbb{R}^{p_d}$ for $r = 1, \ldots, R$ and $d = 1, \ldots, D$. With this, model (3.8) becomes

$$\begin{aligned} Y_i &= \langle \boldsymbol{X}_i, \boldsymbol{B} \rangle + U_i = \langle \boldsymbol{X}_i, \sum_{r=1}^{R} \boldsymbol{\beta}_1^{(r)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r)} \rangle + U_i \\ &= \sum_{r=1}^{R} \mathrm{vec}(\boldsymbol{X}_i)^\top \boldsymbol{\beta}_D^{(r)} \otimes \cdots \otimes \boldsymbol{\beta}_1^{(r)} + U_i. \end{aligned} \tag{3.9}$$

In the above, the coefficient $\boldsymbol{B}$ is determined by the factors $\{\boldsymbol{\beta}_d^r\}_{d=1\dots D}^{r=1\dots R}$. Therefore, the number of unknown parameters of $\boldsymbol{B}$ decreases from $\prod_{d=1}^D p_d$ to $R\sum_{d=1}^D p_d$.

3.4.3. AutoGFI for Tensor Regression.

Model (3.9) can be re-expressed as

$$Y_i = G(\boldsymbol{X}_i, \boldsymbol{\theta}) + U_i$$

with $U_i \overset{iid}{\sim} N(0, \sigma^2)$ and

$$G(\boldsymbol{X}_i, \boldsymbol{\theta}) = \langle \boldsymbol{X}_i, \sum_{r=1}^R \boldsymbol{\beta}_1^{(r)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r)} \rangle.$$

Choosing the squared ℓ_2 norm as the semi-metric ρ, we have

$$\rho(\boldsymbol{Y}, G(\boldsymbol{X}, \boldsymbol{\theta}) + \boldsymbol{U}^*) = \sum_{i=1}^n (Y_i - \langle \boldsymbol{X}_i, \sum_{r=1}^R \boldsymbol{\beta}_1^{(r)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r)} \rangle - U_i^*)^2, \tag{3.10}$$

where $\boldsymbol{Y} = (Y_1, \dots, Y_n)$, $\boldsymbol{X} = (\boldsymbol{X}_1, \dots, \boldsymbol{X}_n)$ and $\boldsymbol{U}^*$ is an independent copy of $\boldsymbol{U}$.

Although the low-rank structure assumption reduces the number of parameters significantly, further regularization is required to ensure the number of parameters is less than the number of observations. Here we employ an ℓ_1 penalty to introduce sparsity into the model: $\Lambda(\boldsymbol{\beta}) = \lambda \sum_{d,r} \|\boldsymbol{\beta}_d^{(r)}\|_1$.

Plugging the above ρ and Λ in the AutoGFI algorithm, Steps (2) and (3) of Algorithm 1 for tensor regression become:

(2). Solve

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \left[\sum_{i=1}^n (Y_i - \langle \boldsymbol{X}_i, \sum_{r=1}^R \boldsymbol{\beta}_1^{(r)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r)} \rangle - U_i^*)^2 + \lambda \sum_{d,r} \|\boldsymbol{\beta}_d^{(r)}\|_1 \right].$$

(3). De-bias $\boldsymbol{\theta}^*$ as described in Section 3.2.

Note that the optimization problem in Step (2) is readily solvable using the block relaxation algorithm proposed by Zhou et al. (2013). The gradient and Hessian matrix of ρ, required in Step (3), can be efficiently and accurately obtained using the automatic differentiation technique. Therefore, when the noise scale σ is known, one can generate a de-biased fiducial sample for the unknown parameters $\boldsymbol{\theta} = \{\boldsymbol{\beta}_d^r\}_{d=1\ldots D}^{r=1\ldots R}$ by AutoGFI.

In practice, when σ is unknown, we first obtain its MLE $\hat{\sigma}$ before generating the fiducial sample and then replace σ by $\hat{\sigma}$ in the algorithm.

A de-biased fiducial sample of N copies of $\{\boldsymbol{\beta}_d^{r*}\}_{d=1\ldots D}^{r=1\ldots R}$ can be generated by repeating the above algorithm. Each copy forms a sample of $\boldsymbol{B}$, denoted by $\boldsymbol{B}^*$. The entries of $\boldsymbol{B}$ can then be estimated by taking the element-wise mean or median of the $\boldsymbol{B}^*$s. Additionally, the $(1-\alpha)$ confidence interval can be constructed by using the percentiles of the sample.

3.4.4. Empirical Performance. This subsection evaluates the practical performance of AutoGFI by comparing it to two Bayesian methods in Guhaniyogi et al. (2017) and Papadogeorgou et al. (2021). The former utilizes CP decomposition for dimension reduction and assumes a multiway shrinkage prior in the model, while the latter softens the CP decomposition by introducing entry-specific variability to the row contributions. We refer to these two methods as Bayesian-hard and Bayesian-soft, respectively.

Throughout the simulation study, we set $\boldsymbol{X}_i$ as a 32×32 matrix with standard normal entries and the corresponding coefficient $\boldsymbol{B}$ as an image varying from low-rank to no low-rank structure, with different degrees of sparsity. Setting $\sigma^2 = 0.5$, we generated 100 replicated datasets with $n = 400$ according to (3.8). The images $\boldsymbol{B}$ considered are shown in Figure 3.1.

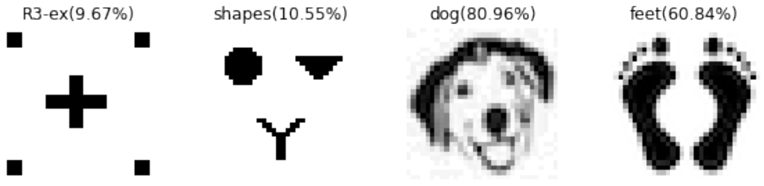


Figure 3.1. Four 32×32 2D images $\boldsymbol{B}$ used in the tensor regression simulation study. Their sparsity levels, defined as % of non-zero pixels, are displayed in parentheses.

The methods are evaluated by their pixel-wise average biases and root-mean-squared errors (rMSEs), as well as frequentist coverages (and widths) of 90%, 95%, and 99% confidence/credible intervals. By default, we set $R = 10$ for AutoGFI and Bayesian-hard and $R = 3$ for Bayesian-soft as recommended by Papadogeorgou et al. (2021). The regularization parameter λ in AutoGFI was chosen by 10-fold cross-validation. The accepting threshold ϵ was automatically chosen as $Q_3 + 1.5(Q_3 - Q_1)$ to remove potential outliers of the losses $\rho(\boldsymbol{Y}, G(\boldsymbol{X}, \boldsymbol{\theta}^*_{\text{de}}) + \boldsymbol{U}^*)$, where the Q_1 and Q_3 are, respectively, the first and third quartiles of these losses. We chose 0.05 as the threshold constant c for de-biasing based on our experiment results, which indicated that the singular values of $\mathbf{H}(\boldsymbol{\theta}^*)$ experienced a significant drop near $0.05\,\zeta_1(\mathbf{H}(\boldsymbol{\theta}^*))$.

Table 3.1 reports the relative performance of the methods on point estimation, which shows that AutoGFI outperformed the two Bayesian methods in both bias and rMSE.

		AutoGFI	Bayesian-hard	Bayesian-soft
R3-ex	bias	0.0024	0.0141	0.0372
	rMSE	0.0079	0.0198	0.1155
shapes	bias	0.0384	0.0675	0.1431
	rMSE	0.0730	0.1008	0.3150
dog	bias	0.1130	0.1310	0.2260
	rMSE	0.1561	0.1736	0.3235
feet	bias	0.1039	0.1226	0.2957
	rMSE	0.1469	0.1625	0.4321

Table 3.1. Average biases and rMSEs of the three methods compared in tensor regression.

Table 3.2 compares the empirical frequentist coverages and widths of 90%, 95%, and 99% confidence/credible intervals for truly zero and truly non-zero pixel entries in $\boldsymbol{B}$. For non-zero entries, when the low-rank assumption is true (R3-ex), AutoGFI and Bayesian-hard performed similarly, with coverages close to the target level. When $\boldsymbol{B}$ is approximately low-rank, AutoGFI outperformed other methods, with coverage levels much closer to the target levels and much higher than the others when sharing the same order of interval widths. For truly zero elements, AutoGFI provided higher coverages (and very often smaller interval widths), indicating that they are better at distinguishing significant regions from zeros. This is because shrinkage causes a significant portion of the relevant coordinates in the fiducial sample to be exactly zero. The suboptimal performance of Bayesian-soft may stem from hyperparameters related to the prior that have not been optimally tuned. Although we adhered to the parameter settings suggested in the original paper, refining these hyperparameters to better align with our specific dataset could potentially improve the model's effectiveness and reliability.

3.5. Matrix Completion

In this section, we present matrix completion, another modern machine learning problem, as our second example. We apply AutoGFI to this problem to further assess its performance in uncertainty quantification.

3.5.1. Background. Matrix completion is a fundamental problem in machine learning, encountered in many applications, with the Netflix movie rating challenge (Bennett and Lanning, 2007) being perhaps the most well-known example. Here the dataset is a large movie rating matrix consisting of 17770 movies (columns) and 480189 customers (rows), with less than 1% of the data matrix (customer-movie pairs) observed. The challenge participants were asked to develop methods to impute the unobserved movie ratings, which is an ill-specified problem and requires additional constraints on the unknown full matrix to make it well-defined. Rank constraints are the most popular choices, with many solutions assuming the full matrix is of low rank. This low-rank assumption is well-empirically motivated in many applications; for example, in the Netflix challenge, it corresponds to the belief that users' movie ratings are based on a few factors. Other applications areas for matrix completion include computer vision (Chen and Suter, 2004), medical imaging (Haldar and Liang, 2010), collaborative filtering (Rennie and Srebro, 2005), and others.

			AutoGFI	Bayesian-hard	Bayesian-soft
R3-ex	non-zero	90%	88.02% (0.0730)	90.77% (0.1032)	88.03% (0.3090)
		95%	93.87% (0.0869)	95.33% (0.1231)	91.32% (0.3725)
		99%	98.49% (0.1131)	98.91% (0.1610)	93.42% (0.5004)
	zero	90%	99.98% (0.0030)	94.99% (0.0649)	97.29% (0.1648)
		95%	99.99% (0.0046)	97.84% (0.0790)	98.35% (0.2105)
		99%	99.99% (0.0094)	99.64% (0.1074)	99.05% (0.3075)
shapes	non-zero	90%	89.48% (0.4225)	77.16% (0.4282)	41.51% (0.4036)
		95%	94.15% (0.5075)	84.33% (0.5072)	45.44% (0.4913)
		99%	97.83% (0.6597)	91.77% (0.6485)	49.57% (0.6721)
	zero	90%	98.52% (0.1422)	94.92% (0.2889)	92.65% (0.3189)
		95%	99.16% (0.1867)	97.22% (0.3464)	93.58% (0.3981)
		99%	99.79% (0.2827)	98.99% (0.4541)	94.24% (0.5683)
dog	non-zero	90%	90.19% (0.5138)	82.98% (0.4757)	70.32% (0.5806)
		95%	94.29% (0.6313)	88.58% (0.5680)	75.86% (0.6993)
		99%	98.35% (0.8853)	94.51% (0.7409)	84.72% (0.9405)
	zero	90%	97.09% (0.3795)	90.96% (0.4394)	87.71% (0.5281)
		95%	99.39% (0.4882)	95.32% (0.5267)	90.78% (0.6405)
		99%	100.0% (0.7434)	98.81% (0.6923)	94.23% (0.8720)
feet	non-zero	90%	84.35% (0.4654)	79.64% (0.4604)	50.57% (0.4993)
		95%	90.96% (0.5709)	86.00% (0.5484)	56.01% (0.5991)
		99%	97.62% (0.7945)	93.45% (0.7102)	64.39% (0.7989)
	zero	90%	96.13% (0.3088)	91.97% (0.4060)	79.68% (0.4175)
		95%	98.62% (0.4004)	95.79% (0.4864)	82.39% (0.5051)
		99%	99.95% (0.6155)	98.64% (0.6375)	84.53% (0.6842)

TABLE 3.2. Empirical frequentist coverages and widths (in parentheses) of 90%, 95%, and 99% confidence/credible intervals among truly zero and truly non-zero pixel entries for different methods in tensor regression.

3.5.2. Problem Definition. Let $\boldsymbol{M}$ be a real matrix of size $m \times n$. Only a small fraction of the elements M_{ij}s of $\boldsymbol{M}$ are observed. Denote the index set of the observed elements as $\Omega = \{1, \ldots, m\} \times \{1, \ldots, n\}$; that is, $(i, j) \in \Omega$ if and only if M_{ij} is observed. For simplicity, we collect all the observed elements in a matrix $\boldsymbol{Y}$, through a projection function $\boldsymbol{f}(\boldsymbol{M})$ of $\boldsymbol{M}$; that is,

$$Y_{ij} = \begin{cases} M_{ij} & \text{if } (i,j) \in \Omega \\ 0 & \text{otherwise.} \end{cases}$$

The goal is to, given $\boldsymbol{Y}$, estimate the unknown elements of $\boldsymbol{M}$. Under the low-rank assumption and in the absence of noise, the matrix completion problem can be formulated as a rank-minimization problem:

$$\min_{\boldsymbol{M}} \operatorname{rank}(\boldsymbol{M}) \quad \text{s.t. } \boldsymbol{Y} = \boldsymbol{f}(\boldsymbol{M}). \tag{3.11}$$

Although (3.11) guarantees the exact recovery of $\boldsymbol{M}$ under some regularity conditions (Candès et al., 2011; Candès and Recht, 2009), it is a non-convex and NP-hard problem (Srebro and Jaakkola, 2003) so no known polynomial-time solutions exist. Various computationally feasible reformulations have been proposed to overcome this limitation, e.g., (Candès and Plan, 2010; Chen and Wainwright, 2015; Gross, 2011; Keshavan et al., 2010; Koltchinskii et al., 2011; Negahban and Wainwright, 2012).

This section focuses on a reformulation that allows for the observed entries to be corrupted by additive noise and assumes the true matrix $\boldsymbol{M}$ can be factorized into two factors, i.e., $\boldsymbol{M} = \boldsymbol{A}\boldsymbol{B}^\top$. The optimization problem for this reformulation is given by:

$$\min_{\boldsymbol{A},\boldsymbol{B}} \|\boldsymbol{Y} - \boldsymbol{f}(\boldsymbol{A}\boldsymbol{B}^\top)\|_F^2 + \lambda\|\boldsymbol{A}\|_F^2 + \lambda\|\boldsymbol{B}\|_F^2, \tag{3.12}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and λ is the regularization parameter. This reformulation has been studied, for example, in Chen et al. (2020, 2019); Yuchi et al. (2022).

3.5.3. AutoGFI for Matrix Completion. Here we apply AutoGFI to solve the noise matrix completion task. With the assumptions behind (3.12), the noisy matrix observation model is

$$Y_{ij} = (\boldsymbol{A}\boldsymbol{B}^\top)_{ij} + U_{ij} \tag{3.13}$$

for all $(i, j) \in \Omega$, where U_{ij} denotes independently generated normal noise at the location (i, j). We assume the data are missing uniformly at random. That is, each index (i, j) is included in Ω independently with the same probability. The data generating equation of (3.13) is

$$\boldsymbol{Y} = \boldsymbol{f}(\boldsymbol{A}\boldsymbol{B}^\top) + \boldsymbol{U},$$

where $\boldsymbol{U}$ is a matrix with $U_{ij} \overset{iid}{\sim} N(0, \sigma^2)$ for all $(i, j) \in \Omega$ and $U_{ij} = 0$ for all $(i, j) \notin \Omega$. Choosing the squared ℓ_2 norm as the semi-metric ρ, we have

$$\rho(\boldsymbol{Y}, \boldsymbol{G}(\boldsymbol{\theta}) + \boldsymbol{U}^*) = \|\boldsymbol{Y} - \boldsymbol{f}(\boldsymbol{A}\boldsymbol{B}^T) - \boldsymbol{U}^*\|_F^2, \tag{3.14}$$

where $\boldsymbol{\theta} = (\boldsymbol{A}, \boldsymbol{B})$. With penalty chosen as $\Lambda(\boldsymbol{\theta}) = \lambda\|\boldsymbol{A}\|_F^2 + \lambda\|\boldsymbol{B}\|_F^2$, this leads to an optimization problem that is similar to (3.12) and can be efficiently solved by the 2-stage algorithm proposed in Chen et al. (2020). Assuming the noise scale σ is known, the fiducial samples of $\boldsymbol{A}$ and $\boldsymbol{B}$ can be generated by Algorithm 1, where Steps (1) to (3) become:

(1) Generate $U_{ij}^* \sim N(0, \sigma^2)$ for $(i, j) \in \Omega$ and leave the other entries as 0.

(2) Solve

$$\boldsymbol{\theta}^* = (\boldsymbol{A}^*, \boldsymbol{B}^*) = \arg\min_{\boldsymbol{A},\boldsymbol{B}} \|\boldsymbol{Y} - f(\boldsymbol{A}\boldsymbol{B}^T) - \boldsymbol{U}^*\|_F^2 + \lambda(\|\boldsymbol{A}\|_F^2 + \|\boldsymbol{B}\|_F^2)$$

using the 2-stage algorithm of Chen et al. (2020).

(3) Debias $\boldsymbol{\theta}^*$ as described in Section 3.2 using ρ defined in (3.14) and $\boldsymbol{\xi}$ the gradient of the penalty term.

In practice, if the matrix rank R and the noise scale σ are not known, they can often be reliably estimated in a data-dependent manner (e.g., Chen et al., 2020).

Using the procedure introduced above, one can obtain a debiased fiducial sample of size N for $\boldsymbol{A}$ and $\boldsymbol{B}$, and consequently, for $\boldsymbol{M} = \boldsymbol{A}\boldsymbol{B}^\top$. The point estimates and confidence intervals of the missing entries of $\boldsymbol{M}$ can then be obtained from these samples.

3.5.4. Empirical Performance. This section compares the empirical performance of AutoGFI for matrix completion with the frequentist method proposed in Chen et al. (2019), referred to as Freq-MC below, and the Bayesian method BayeSMG proposed in Yuchi et al. (2022).

Similar to the experimental setting of Chen et al. (2019), we generated a rank-R matrix $\boldsymbol{M} = \boldsymbol{A}\boldsymbol{B}^\top$, where $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{R}^{n \times R}$ are random orthonormal matrices. We added noise with $\sigma = 0.001$ to obtain $\boldsymbol{Y} = \boldsymbol{M} + \boldsymbol{U}$, where $U_{ij} \sim N(0, \sigma^2)$ are i.i.d. Each entry of $\boldsymbol{Y}$ is observed with probability p independently. We considered two values for n: (500, 1000), two values for R: (2, 5), and two values for p: (20%, 40%), resulting in eight simulation settings.

The methods are evaluated by estimation errors measured in Frobenuis norm and frequentist coverages (and widths) of 90%, 95%, and 99% confidence/credible intervals. We used 10-fold cross-validation to choose the penalty parameter λ for AutoGFI and set $\lambda = 2.5\sigma\sqrt{np}$ for Freq-MC as suggested by the authors. Same as tensor regression, the accepting threshold ϵ was chosen as $Q_3 + 1.5(Q_3 - Q_1)$, where the Q_1 and Q_3 are the, respectively, first and third quartiles of the losses $\rho(\boldsymbol{Y}, G(\boldsymbol{\theta}^*_{\text{de}}) + \boldsymbol{U}^*)$. The threshold constant c used in debiasing was chosen as 0.05, the same as tensor regression.

For each simulation setting, we randomly generated 100 matrices with missing entries, and for each matrix, we applied the above three methods to estimate its missing entries and construct confidence intervals. Table 3.3 reports the estimation errors of the methods, while Tables 3.4 shows the empirical coverage rates. Notice that AutoGFI gave the lowest estimation errors and produced comparable confidence interval coverages when compared to the other two methods.

3.6. Regression with Network Cohesion

In this section, we introduce a regression model which incorporates the network cohesion. We use it our last example to evaluate AutoGFI's performance.

		AutoGFI	BayeSMG	Freq-MC
$n = 500,\ R = 2$	$p = 0.2$	0.1035	0.1449	0.1125
	$p = 0.4$	0.0724	0.1002	0.0738
$n = 500,\ R = 5$	$p = 0.2$	0.1707	0.2335	0.1944
	$p = 0.4$	0.1156	0.1611	0.1207
$n = 1000,\ R = 2$	$p = 0.2$	0.1469	0.2027	0.1500
	$p = 0.4$	0.1040	0.1417	0.1020
$n = 1000,\ R = 5$	$p = 0.2$	0.2373	0.3214	0.2457
	$p = 0.4$	0.1629	0.2248	0.1625

TABLE 3.3. Average estimation errors of the three methods compared in matrix completion.

3.6.1. Background. As modern communication technology advances, network data are becoming increasingly popular. One longstanding problem is community detection, as in identifying friendship circles in a social network. Also, combining information from node features and the network structure has gained interest among researchers, such as using node covariates to assist in inferring the network structure (e.g., Binkiewicz et al., 2017; Su et al., 2020; Zhang et al., 2016).

One can also leverage network structures to assist inference on the node covariates, which is the main focus of this section. We consider a linear regression model with observations connected in a network, where each node is associated with some node covariates and a response variable of interest. To incorporate network effects into traditional predictive models, Li et al. (2019) proposed an regression with network cohesion (RNC) estimator that uses a graph-based regularization to assert similar individual effects for those who are connected in the network. They showed that by adding the network cohesion, the out-of-sample prediction error was significantly improved. This idea of adding graph-regularization has also been applied to many other problems, such as graph-regularized matrix completion (Ma et al., 2011; Rao et al., 2015).

			AutoGFI	BayeSMG	Freq-MC
$n = 500$, $R = 2$	$p = 0.2$	90%	89.08% (0.0006)	89.82% (0.0006)	86.27% (0.0006)
		95%	94.30% (0.0007)	94.85% (0.0008)	92.18% (0.0007)
		99%	98.59% (0.001)	98.90% (0.0010)	97.73% (0.0010)
	$p = 0.4$	90%	89.31% (0.0004)	89.92% (0.0004)	88.44% (0.0004)
		95%	94.46% (0.0005)	94.93% (0.0005)	93.89% (0.0005)
		99%	98.67% (0.0007)	98.96% (0.0007)	98.57% (0.0007)
$n = 500$, $R = 5$	$p = 0.2$	90%	87.73% (0.0010)	89.78% (0.0011)	81.94% (0.0010)
		95%	93.34% (0.0012)	94.83% (0.0013)	88.78% (0.0012)
		99%	98.24% (0.0016)	98.90% (0.0017)	96.04% (0.0015)
	$p = 0.4$	90%	88.64% (0.0007)	89.90% (0.0007)	86.69% (0.0007)
		95%	94.00% (0.0008)	94.92% (0.0009)	92.62% (0.0008)
		99%	98.50% (0.0011)	98.96% (0.0011)	98.07% (0.0011)
$n = 1000$, $R = 2$	$p = 0.2$	90%	89.54% (0.0004)	89.79% (0.0004)	88.15% (0.0004)
		95%	94.57% (0.0005)	94.81% (0.0005)	93.62% (0.0005)
		99%	98.70% (0.0007)	98.89% (0.0007)	98.43% (0.0007)
	$p = 0.4$	90%	89.57% (0.0003)	90.10% (0.0003)	89.46% (0.0003)
		95%	94.67% (0.0004)	95.05% (0.0004)	94.62% (0.0004)
		99%	98.76% (0.0005)	98.95% (0.0005)	98.83% (0.0005)
$n = 1000$, $R = 5$	$p = 0.2$	90%	89.01% (0.0007)	89.84% (0.0007)	86.44% (0.0007)
		95%	94.22% (0.0009)	94.86% (0.0009)	92.38% (0.0008)
		99%	98.60% (0.0011)	98.90% (0.0011)	97.94% (0.0011)
	$p = 0.4$	90%	89.66% (0.0005)	90.06% (0.0005)	88.96% (0.0005)
		95%	94.68% (0.0006)	95.01% (0.0006)	94.26% (0.0006)
		99%	98.71% (0.0008)	98.97% (0.0008)	98.69% (0.0008)

TABLE 3.4. Empirical frequentist coverages and widths (in parentheses) of 90%, 95%, and 99% confidence/credible intervals for the missing entries in matrix completion.

This section demonstrates the use of AutoGFI in the RNC problem to provide uncertainty quantification of the model parameters. We note that, when the noise is additive, AutoGFI can be straightforwardly extended to other graph-regularized methods.

3.6.2. Problem Definition. In linear regression with network cohesion (Li et al., 2019), the model is

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\alpha} + \boldsymbol{U}, \tag{3.15}$$

where $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$ is the response vector, $\boldsymbol{X}$ is an $n \times p$ design matrix, $\boldsymbol{\alpha}$ is an $n \times 1$ vector of individual effects, $\boldsymbol{\beta}$ is a $p \times 1$ vector of fixed effects, and $\boldsymbol{U} \sim N(\mathbf{0}, \sigma \mathbf{I}_n)$ is the error term.

We assume these n samples are connected in a network $\boldsymbol{\mathcal{G}} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V} = \{1, \ldots, n\}$ is the node set, and $\boldsymbol{E} \subset \boldsymbol{V} \times \boldsymbol{V}$ is the edge set. The Laplacian of $\boldsymbol{\mathcal{G}}$ is defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ where $\boldsymbol{A}$ is the adjacency matrix and $\boldsymbol{D} = \text{diag}(d_1, \ldots, d_n)$ with d_i being the degree of the i-th node. The RNC estimator is defined as the minimizer of

$$L(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{\alpha}\|_2^2 + \lambda \boldsymbol{\alpha}^T \boldsymbol{L} \boldsymbol{\alpha}, \tag{3.16}$$

where $\lambda > 0$ is a tuning parameter selected by cross-validation.

In general, the $n + p$ parameters ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) cannot be estimated from n observations without further assumptions. In this network regression setting, we assume that the design matrix $\boldsymbol{X}$ is centered and has full column rank. Li et al. (2019) proved that when the network contains additional information beyond what is contained in $\boldsymbol{X}$, the RNC estimator defined in equation (3.16) always exists. We will make the same assumption here.

Li et al. (2019) also derived the asymptotic bounds for the bias and the variance of the RNC estimator. In particular, the bias term depends on $\boldsymbol{L}\boldsymbol{\alpha}$, and the norm of bias grows with $\|\boldsymbol{L}\boldsymbol{\alpha}\|$. Under the condition that $\|\boldsymbol{L}\boldsymbol{\alpha}\| = 0$, the RNC estimator is unbiased. This occurs only when the individual effect on node i is simply the average of those of its neighboring nodes; otherwise, there is bias involved.

3.6.3. AutoGFI for Network Regression. The data generating equation of network regression is given in (3.15). Using our notations from Section 3.1, we can express it in the form of $\boldsymbol{Y} = \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{\theta}) + \boldsymbol{U}$ with $\boldsymbol{G}(\boldsymbol{X}, \boldsymbol{\theta}) = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\alpha}$. The unknown parameters are

$\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, and the random component is $\boldsymbol{U}$. Also, the semi-metric ρ (the squared ℓ_2 norm) for this model is

$$\rho(\boldsymbol{Y}, \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{\theta}) + \boldsymbol{U}^*) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{\alpha} - \boldsymbol{U}^*\|_2^2, \tag{3.17}$$

where $\boldsymbol{U}^*$ is an independent copy of $\boldsymbol{U}$. According to (3.16), the penalty term is

$$\Lambda(\boldsymbol{\theta}) = \lambda \boldsymbol{\alpha}^T \boldsymbol{L} \boldsymbol{\alpha}. \tag{3.18}$$

Unlike the penalty terms used in Sections 3.4 and 3.5, formula (3.18) only penalizes the component of $\boldsymbol{\alpha}$ that lies in the column space of $\boldsymbol{L}$. The component of $\boldsymbol{\alpha}$ in the null space does not contribute to the penalty term. Consequently, we only need to debias the component in the column space of $\boldsymbol{L}$, i.e., we only debias the part that was penalized.

The optimization problem in Step (2) of AutoGFI can be solved by the Newton-Raphson or another appropriate convex optimization algorithm. To de-bias the penalized parameters, we separate $\boldsymbol{\alpha}$ as $\boldsymbol{\alpha} = (\mathbf{I} - \boldsymbol{P}_{\boldsymbol{L}})\boldsymbol{\alpha} + \boldsymbol{L}^{-1/2}\boldsymbol{\eta}$ and transform (3.17) to

$$\rho_{\boldsymbol{\eta}}(\boldsymbol{Y}, \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{\theta}) + \boldsymbol{U}^*) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} - (\mathbf{I} - \boldsymbol{P}_{\boldsymbol{L}})\boldsymbol{\alpha} - \boldsymbol{L}^{-1/2}\boldsymbol{\eta} - \boldsymbol{U}^*\|^2, \tag{3.19}$$

where $\boldsymbol{P}_{\boldsymbol{L}}$ is the project matrix of $\boldsymbol{L}$ and $\boldsymbol{\eta} = \boldsymbol{L}^{1/2}\boldsymbol{\alpha}$.

From (3.19), we can see that $\boldsymbol{\eta}$ is the term that we penalize. Thus, we first debias $\boldsymbol{\eta}$, then use $\boldsymbol{d}(\boldsymbol{\eta})$ to rebuild a debiased $\boldsymbol{\alpha}$, and finally refit the model to debias $\boldsymbol{\beta}$. Therefore, Steps (2) and (3) of Algorithm 1 for network regression become:

(2). Solve

$$\boldsymbol{\theta}^* = (\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = \arg\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{\alpha} - \boldsymbol{U}^*\|^2 + \lambda \boldsymbol{\alpha}^T L \boldsymbol{\alpha}.$$

(3). Let $\boldsymbol{\eta}^* = \boldsymbol{L}^{1/2}\boldsymbol{\alpha}^*$ and plug it in (3.19) so that

$$\rho_{\boldsymbol{\eta}}(\boldsymbol{Y}, \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{\theta}) + \boldsymbol{U}^*) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}^* - (\mathbf{I} - \boldsymbol{P}_{\boldsymbol{L}})\boldsymbol{\alpha}^* - \boldsymbol{L}^{-1/2}\boldsymbol{\eta}^* - \boldsymbol{U}^*\|^2.$$

(a) Debias $\boldsymbol{\eta}^*$ by $\boldsymbol{\eta}^*_{\text{de}} = \boldsymbol{\eta}^* + \boldsymbol{H}(\boldsymbol{\eta}^*)^{\text{pinv}}\boldsymbol{\xi}(\boldsymbol{\eta}^*)$.

(b) Calculate the debiased $\boldsymbol{\alpha}^*$ as $\boldsymbol{\alpha}^*_{\text{de}} = \boldsymbol{P}_{\boldsymbol{L}}\boldsymbol{\alpha}^* + \boldsymbol{L}^{-1/2}\boldsymbol{\eta}^*_{\text{de}}$.

(c) Obtain the debiased $\boldsymbol{\beta}^*$ by refitting the model, i.e.,

$$\boldsymbol{\beta}^*_{\text{de}} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top(\boldsymbol{Y} - \boldsymbol{L}^{-1/2}\boldsymbol{\eta}^*_{\text{de}}).$$

Set $\boldsymbol{\theta}^*_{\text{de}} = (\boldsymbol{\beta}^*_{\text{de}}, \boldsymbol{\alpha}^*_{\text{de}})$.

In the above, as the Laplacian matrix $\boldsymbol{L}$ is not of full rank, the generalized inverse of $\boldsymbol{L}^{1/2}$ is used instead. This de-biased fiducial sample enables the construction of a $(1-\alpha)$ fiducial confidence interval for $\boldsymbol{\beta}$ using the $\alpha/2$ and $1-\alpha/2$ quantiles of $\boldsymbol{\beta}^*_{\text{de}}$.

3.6.4. Empirical Performance. We evaluated the practical performance of AutoGFI using a stochastic block model with 3 blocks. The probability of a connection between two nodes in the same block was p_w, and the probability of a connection between nodes in different blocks was p_b. The adjacency matrix $\boldsymbol{A} \in \{0,1\}^{n\times n}$ was generated independently with $A_{ij} = A_{ji}$ and $A_{ii} = 0$ from the probability matrix $\boldsymbol{P} = \boldsymbol{B} \otimes \boldsymbol{J}_{n/3}$, where $\boldsymbol{B}$ is given by

$$\boldsymbol{B} = \begin{pmatrix} p_w & p_b & p_b \\ p_b & p_w & p_b \\ p_b & p_b & p_w \end{pmatrix}$$

and $\boldsymbol{J}_k$ is a $k \times k$ matrix with all ones.

We generated $\boldsymbol{\beta}$ from $N(\mathbf{1}, \mathbf{I}_p)$, where $p = 10$ in our experiments, and the covariate matrix $\boldsymbol{X}$ was generated independently from the standard normal. Therefore, the columns are uncorrelated and have 0 means. The $\boldsymbol{\alpha}$ was generated independently from a normal distribution with the mean determined by the node's block assignment $N(\gamma_k, s^2)$, where $\gamma_1 = -1, \gamma_2 = 0, \gamma_3 = 1$. Finally, $\boldsymbol{Y} = \boldsymbol{\alpha} + \boldsymbol{X}^T\boldsymbol{\beta} + \boldsymbol{U}$ with $\boldsymbol{U} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$.

As mentioned before, the bias of the RNC estimator depends on $\|\boldsymbol{L}\boldsymbol{\alpha}\|$. We tested AutoGFI under both unbiased (i.e., $\|\boldsymbol{L}\boldsymbol{\alpha}\| = 0$) and biased (i.e., $\|\boldsymbol{L}\boldsymbol{\alpha}\| > 0$) conditions. Specifically, we set $p_w = 0.2$, $\sigma = 0.5$ and used three choices of (p_b, s): (i) (0, 0) which gives $\|\boldsymbol{L\alpha}\| = 0$, (ii) (0.01, 0.1) which gives $\|\boldsymbol{L\alpha}\| \approx 65$, and (iii) (0.02, 0.1) which gives $\|\boldsymbol{L\alpha}\| \approx 108$.

Once the network $\boldsymbol{A}$, design matrix $\boldsymbol{X}$ and parameters ($\boldsymbol{\alpha}$, $\boldsymbol{\beta}$) were generated, we fixed them. We then simulated 500 datasets by re-generating $\boldsymbol{U}$ and generating a fiducial sample of size 1000 for ($\boldsymbol{\alpha}$, $\boldsymbol{\beta}$) for each dataset using AutoGFI. These samples were then used to form point estimates and confidence intervals.

We compared AutoGFI with two other methods. The first was ordinary linear regression (OLR) without using network information. We used the true σ^2 and the classical way of constructing confidence intervals. Since the network is correlated with individual effects, ignoring them would lose information and worsen the estimation results. The second method was based on a fixed effects linear model, for which we assumed the true group assignments of nodes are known. That is, there were 3 known groups, and the individuals within each group shared a common intercept. When $s \neq 0$, the randomness in $\boldsymbol{\alpha}$ was simply combined into $\boldsymbol{U}$, i.e., $U_i \sim N(0, s^2 + \sigma^2)$. Note that this second method could be seen as an oracle method since it has access to the usually unknown group assignment. For the AutoGFI, the σ^2 was estimated by the MSE of the RNC estimator before the fiducial sample was generated. We compared the rMSEs of the methods and the coverages and widths of their confidence intervals for the fixed effect parameters $\boldsymbol{\beta}$. The results are summarized in Table 3.5 for different settings.

We can see AutoGFI outperforms the OLS method and is very similar to the oracle in terms of rMSEs when estimating $\boldsymbol{\beta}$. The coverage of AutoGFI based confidence intervals is also very close to the target level under all cases. As the value of $\|\boldsymbol{L\alpha}\|$ increases, all confidence intervals tend to be wider due to more noise in the data. Compared to the oracle, the AutoGFI confidence intervals are often slightly wider because σ^2 is slightly overestimated by the MSE of the RNC estimator, resulting in more dispersed fiducial samples and thus wider confidence intervals. However, under the unbiased case (i.e., $\|\boldsymbol{L\alpha}\| = 0$), the AutoGFI confidence intervals are almost identical to the oracle ones. On the other hand, the confidence intervals of OLR have dramatically lower coverage than the target. This shows the necessity of taking the network information into consideration.

		AutoGFI	oracle	OLR
$p_b = 0$, $s = 0$, $\|\boldsymbol{L\alpha}\| = 0$	rMSE	0.0292	0.0290	0.0537
	90%	89.32% (0.0992)	89.98% (0.0981)	62.90% (0.0980)
	95%	94.80% (0.1180)	95.38% (0.1170)	71.46% (0.1169)
	99%	98.68% (0.1535)	99.10% (0.1542)	83.68% (0.1540)
$p_b = 0.01$, $s = 0.1$, $\|\boldsymbol{L\alpha}\| \approx 65$	rMSE	0.0298	0.0295	0.0571
	90%	90.04% (0.1055)	90.00% (0.1002)	54.46% (0.0979)
	95%	94.68% (0.1255)	95.04% (0.1195)	64.66% (0.1167)
	99%	98.66% (0.1632)	99.12% (0.1574)	80.72% (0.1538)
$p_b = 0.02$, $s = 0.1$, $\|\boldsymbol{L\alpha}\| \approx 108$	rMSE	0.0318	0.0299	0.0606
	90%	91.62% (0.1157)	89.58% (0.1008)	55.16% (0.0984)
	95%	95.78% (0.1377)	95.04% (0.1203)	64.00% (0.1173)
	99%	99.12% (0.1791)	99.12% (0.1585)	77.46% (0.1546)

TABLE 3.5. Average rMSEs of the parameter estimates and empirical frequentist coverages and widths (in parentheses) of various confidence intervals in network regression.

3.7. Conclusion

In this chapter, we discussed a new form of GFI, which can be applied to a broad range of high-dimensional and/or nonlinear additive noise problems. We introduced a practical and straightforward-to-implement algorithm, AutoGFI, for generating fiducial samples of the parameters of interest. This approach is particularly useful in situations where uncertainty quantification has been difficult or impossible using traditional inference methods. Numerical results of applying AutoGFI to three challenging problems demonstrate its highly competitive performance compared to tailor-made competitors. Overall, the result has shown that GFI is a promising alternative to traditional methods for addressing important and practical inference problems.

CHAPTER 4

Fiducial Selector for High-Dimensional Linear Regression

High-dimensional linear regression is a pivotal tool in numerous scientific fields. In response to its importance, we introduce Fiducial Selector, a novel approach aimed at enhancing the application of GFI to such models. Built upon insights from the sampling framework detailed in Chapter 3, Fiducial Selector is tailored to address the challenges of high-dimensional linear regression. Notably, it incorporates a specialized de-biasing procedure designed to mitigate the bias introduced by regularization terms within the model. This new approach, Fiducial Selector, has the following attractive properties:

- It enjoys model selection consistency under certain conditions.
- It provides unbiased estimates for the significant parameters.
- It is straightforward to implement.
- It is computationally fast.
- It demonstrates highly competitive empirical performance.

In this chapter, we introduce Fiducial Selector in the following order. Section 4.1 provides a brief review of recent work on the high-dimensional linear regression problem. Section 4.2 presents Fiducial Selector in detail. Sections 4.3 and 4.4 examine the theoretical and empirical properties of Fiducial Selector, respectively. Concluding remarks are offered in Section 4.5, and technical details are provided in Appendix B.

4.1. Introduction

The high-dimensional linear regression problem has attracted enormous research attention in the statistics literature. Consider the following linear model:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{e}, \tag{4.1}$$

where $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$ is a vector of n response variables, $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$ is a design matrix of size $n \times p$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ is a size p vector of parameters, and $\boldsymbol{e} = (e_1, \ldots, e_n)^T$ is a size n vector of i.i.d. normal random errors with zero mean and unknown variance σ^2. The error term $\boldsymbol{e}$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are assumed to be independent.

In high-dimensional settings where $p \gg n$, we often assume the true model is sparse. Specifically, let $\mathcal{A} = \{j : \beta_j \neq 0, j = 1, \ldots, p\}$ be the support of $\boldsymbol{\beta}$, with cardinality $s = |\mathcal{A}|$ then $s \ll p$. Identifying the significant variables becomes the major task, which we call the variable selection problem. This problem has been well studied in the past decade, and regularized methods are the most popular approach, including the l_1 regularized method lasso (Tibshirani, 1996) and its variations, such as non-concave penalized method (SCAD) (Fan and Li, 2001), elastic net (Zou and Hastie, 2005), the adaptive lasso (Zou, 2006), and the Dantzig selector (Candes and Tao, 2007).

In recent years, much attention has been given to the statistical inference problem in high-dimensional regression. For example, Bühlmann (2013), Zhang and Zhang (2014), van de Geer et al. (2014), and Javanmard and Montanari (2014) proposed different versions of "de-biased" lasso and applied them to perform statistical inference in the high-dimensional linear regression problem. These methods require a sparse estimate of the precision matrix. However, due to the singularity of the sample covariance matrix, obtaining a satisfactory solution is challenging. Typically, the computational workload associated with these methods is substantial. Furthermore, specific regularity conditions must be met to ensure that the estimated precision matrix is feasible and stable.

Post-selection inference is another class of methods for statistical inference in high-dimensional linear regression problems. As implied by its name, post-selection inference only performs statistical inference for the selected coefficients. Such inference is conditioning on the selection event. Multiple methods have been proposed to conduct a valid post-model selection inference. Among those, one kind is the sample splitting method (Wasserman and Roeder, 2009; Meinshausen et al., 2009) that splits data to do model selection and inference separately. Such methods are sensitive to the manner in which the sample is split, and several methods, including Rasines and Young (2022) and James Leiner and Ramdas (2023), are proposed to provide more elaborate splitting strategies. Another kind, including but not limited to Lockhart et al. (2014), Lee et al. (2016), and Tian and Taylor (2018), is to correct inferential statements for variable selection by conditioning on the region of the sample space that led to the selection being made. Resampling methods (Jessica Minnier and Cai, 2011) and bootstrap-based methods (Chatterjee and Lahiri, 2011; Liu and Yu, 2013; Tibshirani et al., 2018) are also proposed to solve this problem. Interested readers can refer to Zhang et al. (2022) for a comprehensive review of post-selection inference. Many other new tools are combining the above techniques to provide more valid inferences for high-dimensional problems. Some of the most recent ones include the simultaneous post-model selection inference method proposed by Wang et al. (2021), the bootstrap lasso and partial ridge method proposed by Liu et al. (2020), the data splitting-based method proposed by Fei and Li (2021), and the method proposed by Battey and Reid (2023).

GFI has also been previously applied to the high-dimensional linear regression problem. In Lai et al. (2015), a sampling method was introduced that allows direct sampling from the GFD as delineated in equations (2.5) and (2.6). This method incorporates additional penalty structural equations into equation (2.5) to manage high dimensionality. It involves a complex sampling process to emulate the GFD. Furthermore, the general approach AutoGFI, introduced in Chapter 3, is designed to facilitate the application of GFI across various problems, including high-dimensional regression. However, the broad applicability of AutoGFI sometimes compromises its precision, leading to undesirable empirical outcomes. To mitigate the complexities of sampling and enhance problem-specific performance, we propose a new approach, termed Fiducial Selector. This method retains the simplicity of sampling characteristic of AutoGFI but is specifically tailored for high-dimensional linear regression.

4.2. Methodology

4.2.1. Fiducial Selector. We start by aligning the notations used in the linear model (4.1) with those introduced in the fiducial framework in Chapter 2. Assuming the noise scale σ is known, the only unknown parameter in the linear model is $\boldsymbol{\beta}$, thus $\boldsymbol{\theta} = \boldsymbol{\beta}$. The error term $\boldsymbol{e}$ can be expressed as $\sigma \boldsymbol{U}$, where $\boldsymbol{U}$ represents the random component in the model following a known distribution, specifically $\mathcal{N}(\mathbf{0}, \mathbf{I}_n)$. After aligning the notations, it becomes evident that (4.1) corresponds to the data generating equation for the linear models.

Since we are aiming at a sparse solution, we propose to select the $\boldsymbol{\beta}$ with the smallest l_1 norm as the solution. In particular, we generalize the optimization problem (2.2), used for producing fiducial samples, to the following form:

$$\boldsymbol{Q}_{\boldsymbol{y}}(\boldsymbol{U}) = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} - \sigma \boldsymbol{U}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1, \tag{4.2}$$

where $\boldsymbol{y}$ is the observed value of $\boldsymbol{Y}$ and the distribution of $\boldsymbol{U}$ is truncated to the set

$$\mathcal{U}_{\boldsymbol{y},\epsilon} = \{\boldsymbol{u} : \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} - \sigma \boldsymbol{u}\|_2^2 \leq \epsilon\}. \tag{4.3}$$

We term the limiting distribution as $\epsilon \to 0$ the raw fiducial selector.

4.2.2. Practical Generation of Fiducial Sample from (4.2). In this subsection, we propose a practical procedure to generate a fiducial sample of $\boldsymbol{\beta}$ from the GFD defined by equation (4.2). The resulting fiducial sample, de-biased using the method described in Section 4.2.3, can be used for statistical estimation and inference, which will be discussed in Section 4.2.5.

By definition, a realization of the fiducial selector, denoted as $\boldsymbol{\beta}^*$, is obtained by solving the optimization problem (4.2), where the truncated random component $\boldsymbol{U}$ is replaced by one of its realizations, $\boldsymbol{u}^*$. This process begins by generating a $\boldsymbol{u}^*$ from the original unbounded distribution, specifically $\mathcal{N}(\mathbf{0}, \sigma \mathbf{I}_n)$. Then with $\boldsymbol{U}$ replaced by $\boldsymbol{u}^*$, we solve the optimization problem (4.2) for $\boldsymbol{\beta}^*$, which is equivalent to fitting a lasso regression from $\boldsymbol{y} - \sigma \boldsymbol{u}^*$ to $\boldsymbol{X}$.

Next, we check whether $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^* - \sigma\boldsymbol{u}^*\|_2^2 \leq \epsilon$ and only accept $\boldsymbol{\beta}^*$ as a valid realization of the fiducial selector if the condition is met. If the condition is violated, a new $\boldsymbol{u}^*$ is generated and the process repeats. This accept/rejection step is to ensure $\boldsymbol{u}^*$ is from the truncated set defined in (4.3). In summary, a fiducial sample of size N can be generated by repeating the following procedure N times:

(1) Generate $\boldsymbol{u}^*$ from $\mathcal{N}(\mathbf{0}, \sigma\boldsymbol{I}_n)$.

(2) Fit a lasso regression on $(\boldsymbol{y} - \sigma\boldsymbol{u}^*, \boldsymbol{X})$ to obtain $\boldsymbol{\beta}^*$.

(3) Accept $\boldsymbol{\beta}^*$ if $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^* - \sigma\boldsymbol{u}^*\|_2^2 \leq \epsilon$; otherwise, return to Step (1).

4.2.3. De-biasing Method for Fiducial Selector. Due to the bias of the lasso estimator, the raw fiducial selector defined by (4.2) and the fiducial sample generated from Section 4.2.2 is biased. Here, we develop a de-biasing method for the fiducial samples of the fiducial selector.

The idea is based on the following facts of the standardized score function $\boldsymbol{s}(\boldsymbol{\beta})$,

$$\boldsymbol{s}(\boldsymbol{\beta}) = \boldsymbol{I}(\boldsymbol{\beta})^{-1/2} \frac{\partial}{\partial \boldsymbol{\beta}} \log f, \tag{4.4}$$

where f is the likelihood function and $\boldsymbol{I}(\boldsymbol{\beta})$ is the Fisher information. This standardized score function satisfies the following two equations,

$$\mathbf{E}\left(\boldsymbol{I}(\boldsymbol{\beta})^{-1/2} \frac{\partial}{\partial \boldsymbol{\beta}} \log f\right) = \mathbf{0}, \tag{4.5}$$

$$\mathbf{Var}\left(\boldsymbol{I}(\boldsymbol{\beta})^{-1/2} \frac{\partial}{\partial \boldsymbol{\beta}} \log f\right) = \mathbf{I}_{p\times p}, \tag{4.6}$$

For the linear model, we can calculate (4.4) as follows,

$$\boldsymbol{s}(\boldsymbol{\beta}) = \boldsymbol{I}(\boldsymbol{\beta})^{-1/2} \frac{\partial}{\partial \boldsymbol{\beta}} \log f = \frac{1}{\sigma}(\boldsymbol{X}^T\boldsymbol{X})^{-1/2}(\boldsymbol{X}^T\boldsymbol{y} - \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}). \tag{4.7}$$

An unbiased estimator of $\boldsymbol{\beta}$ should make (4.7) satisfy (4.5) and (4.6). Thus, we define the de-biased fiducial sample as follows.

Definition 4.2.1. *De-biased fiducial sample: A de-biased fiducial sample is defined by*

$$\hat{\boldsymbol{\beta}} = -\boldsymbol{S}_{\boldsymbol{\beta}^*}^{-1/2}(\boldsymbol{\beta}^* - \mathbf{E}[\boldsymbol{\beta}^*]), \tag{4.8}$$

where $\boldsymbol{\beta}^*$ *is the original fiducial sample,* $\mathbf{E}[\boldsymbol{\beta}^*]$ *and* $\boldsymbol{S}_{\boldsymbol{\beta}^*}$ *are the mean and variance-covariance matrix of the original fiducial distribution respectively.*

In practice, $\mathbf{E}[\boldsymbol{\beta}^*]$ and $\boldsymbol{S}_{\boldsymbol{\beta}^*}$ are replaced with the average and sample variance-covariance matrix of the all the original fiducial samples.

Lemma 4.2.2. *The de-biased fiducial sample for the linear regression problem has the following closed-form expression:*

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} + \sigma(\boldsymbol{X}^T\boldsymbol{X})^{-1/2}\boldsymbol{S}_{\boldsymbol{\beta}^*}^{-1/2}(\boldsymbol{\beta}^* - \mathbf{E}[\boldsymbol{\beta}^*]). \tag{4.9}$$

The proof of (4.9) is a direct result of (4.7) and (4.8).

In high-dimensional settings, the true parameter vector $\boldsymbol{\beta}$ is often assumed to be sparse. When performing de-biasing, it is crucial to maintain this sparsity to avoid losing the benefits of the penalty. Therefore, we need to decide which β_js should be de-biased. Additionally, when $p > n$, $\boldsymbol{X}^\top\boldsymbol{X}$ is not full-rank, and we need to obtain an approximate inverse of $\boldsymbol{X}^\top\boldsymbol{X}$. A simple approach to address these issues is to perform the de-biasing procedure only on the significant β_js.

To identify those significant β_js, we use the following simple procedure. For each β_j, we count the percentage of zero values in the fiducial sample. If it is more than 50%, we say that this particular β_j is not significant. Otherwise, we treat this β_j as a significant parameter and perform de-biasing. Barbieri and Berger (2004) applied a similar idea to determine the significance of a parameter in the Bayesian inference context. The choice of the threshold (50%) is the same as the thresholds in Barbieri and Berger (2004) and Lai et al. (2015). We denote the identified significant set by $\hat{\boldsymbol{\mathcal{A}}}$ and define it as

$$\hat{\mathcal{A}} = \{j : \sum_{k=1}^{N} \mathbb{1}(\beta^*_{(k),j} \neq 0) \geq \lfloor \frac{N}{2} \rfloor\}, \tag{4.10}$$

where $\beta^*_{(k),j}$ denotes the j-th element of the k-th realization of the fiducial selector and N is the fiducial sample size.

To sum up, upon decomposing $\boldsymbol{X}$ into $\boldsymbol{X} = (\boldsymbol{X}_{\hat{\boldsymbol{\mathcal{A}}}}, \boldsymbol{X}_{\hat{\boldsymbol{\mathcal{A}}}^c})$, where $\boldsymbol{X}_{\hat{\boldsymbol{\mathcal{A}}}}$ comprises columns with coordinates in $\hat{\mathcal{A}}$ and $\boldsymbol{X}_{\hat{\boldsymbol{\mathcal{A}}}^c}$ includes the remaining columns, the de-biased fiducial sample $\hat{\boldsymbol{\beta}} = d(\boldsymbol{\beta}^*)$ is characterized as follows:

$$\hat{\boldsymbol{\beta}}_{\hat{\boldsymbol{\mathcal{A}}}} = (\boldsymbol{X}^{\boldsymbol{T}}_{\hat{\boldsymbol{\mathcal{A}}}} \boldsymbol{X}_{\hat{\boldsymbol{\mathcal{A}}}})^{-1} \boldsymbol{X}^{\boldsymbol{T}}_{\hat{\boldsymbol{\mathcal{A}}}} \boldsymbol{y} + \sigma (\boldsymbol{X}^{T}_{\hat{\boldsymbol{\mathcal{A}}}} \boldsymbol{X}_{\hat{\boldsymbol{\mathcal{A}}}})^{-1/2} \hat{\boldsymbol{S}}^{-1/2}_{\boldsymbol{\beta}^*_{\hat{\boldsymbol{\mathcal{A}}}}} (\boldsymbol{\beta}^*_{\hat{\boldsymbol{\mathcal{A}}}} - \overline{\boldsymbol{\beta}}^*_{\hat{\mathcal{A}}}), \tag{4.11}$$

$$\hat{\boldsymbol{\beta}}_{\hat{\boldsymbol{\mathcal{A}}}^c} = \mathbf{0}. \tag{4.12}$$

Here, $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}}$ denotes the vector composed of $\hat{\beta}_j$ for indices $j \in \hat{\mathcal{A}}$, and similarly, $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}^c}$ refers to the coefficients corresponding to the complementary set of indices. The notation $\overline{\boldsymbol{\beta}}^*_{\hat{\mathcal{A}}}$ represents the average of all realizations of $\boldsymbol{\beta}^*_{\hat{\mathcal{A}}}$ within the fiducial sample, while $\hat{\boldsymbol{S}}_{\boldsymbol{\beta}^*_{\hat{\mathcal{A}}}}$ denotes the sample variance-covariance matrix of $\boldsymbol{\beta}^*_{\hat{\mathcal{A}}}$. In the following, we will use $\boldsymbol{\beta}^*$ to represent a copy in the original fiducial sample of the fiducial selector and use $\hat{\boldsymbol{\beta}}$ to represent a copy in the de-biased fiducial sample of the fiducial selector. It is important to note that all copies in the de-biased fiducial sample of the fiducial selector have the same set of nonzero parameters.

4.2.4. Implementation Details. In this subsection, we clarify some implementation details and summarize the steps of the proposed method in Algorithm 2.

Firstly, the processes introduced in both Section 4.2.2 and 4.2.3 are based on the assumption that the noise scale σ is known, which is often not the case in practice. When σ is unavailable, we recommend obtaining an estimate, $\hat{\sigma}$, prior to applying the method. This $\hat{\sigma}$ can then be used in place of σ for both the sampling and de-biasing procedures. In the statistics literature, several methods have been proposed to estimate σ in the high-dimensional regression setting. Examples include Fan et al. (2012), Lai et al. (2015), and Sun and Zhang (2012). In our numerical experiments, we used the R package `selectiveInference`, introduced in Taylor and Tibshirani (2015), to obtain $\hat{\sigma}$.

Secondly, a well-tuned penalty parameter λ for the lasso regression is important in practice. The classical cross-validation method can be used to tune this parameter. However, considering the potential computational cost of conducting full cross-validation for the fiducial selector, such as tuning λ for every copy of the fiducial sample, we investigated a quicker approach that provides similar coverage results to the full cross-validation approach. First, we apply the cross-validation procedure for lasso on the observed data $(\boldsymbol{y}, \boldsymbol{X})$ to obtain the cross-validated λ_{CV}. Then, this λ_{CV} is used throughout the entire fiducial sample generation procedure.

Another hyper-parameter in our analysis is the tolerance, denoted by ϵ. Although, according to Definition 2.1.1, it is preferable to set ϵ as small as possible, we have observed in practice that the specific value of this tolerance is not critically stringent. Even with a relatively large tolerance, the fiducial selector demonstrates excellent empirical performance. Consequently, throughout our experiments, we chose $\epsilon = \|\boldsymbol{y} - \overline{\boldsymbol{y}}\|_2^2$ to effectively exclude extremely poor fiducial samples.

In summary, the implementation steps of the proposed method are outlined in Algorithm 2.

It is worth noting that Fiducial Selector, Algorithm 2, is an overall fast method due to the efficiency of the lasso procedure and the closed-form nature of the de-biasing formula. The computational time of Fiducial Selector and its competitors is reported in Section 4.4.

4.2.5. Point Estimates and Confidence Intervals. Applying Algorithm 2, one can obtain a de-biased fiducial sample consisting of multiple copies of $\hat{\boldsymbol{\beta}}$. Similar to the Bayesian posterior sample, we can use this fiducial sample to form point estimates and confidence intervals for $\boldsymbol{\beta}$. For example, the average of all copies in the fiducial sample can be used as

Algorithm 2 Generating fiducial sample for high-dimensional linear models.

```
1: Input: Data $(\boldsymbol{X}, \boldsymbol{y})$; noise scale $\sigma$; penalty parameter $\lambda_{CV}$; tolerance $\epsilon$; fiducial sample size $N$
2: Output: $N$ realizations of the de-biased fiducial sample: $\{\hat{\boldsymbol{\beta}}_{(k)}\}_{k=1}^{N}$
3: for $k = 1$ to $N$ do
4:     Generate $\boldsymbol{u}_{(k)}$ from $\mathcal{N}(\mathbf{0}, \sigma \mathbf{I}_n)$.
5:     Fit a lasso regression to $\{\boldsymbol{y} - \sigma \boldsymbol{u}_{(k)}, \boldsymbol{X}\}$ using $\lambda_{CV}$ to obtain $\boldsymbol{\beta}^*_{(k)}$.
6:     if $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*_{(k)} - \sigma \boldsymbol{u}_{(k)}\|_2^2 \leq \epsilon$ then
7:         Accept $\boldsymbol{\beta}^*_{(k)}$.
8:     else
9:         Reject $\boldsymbol{\beta}^*_{(k)}$ and restart the iteration at line 4.
10: Determine the significant set $\hat{\mathcal{A}}$ as defined in (4.10).
11: De-bias $\boldsymbol{\beta}^*_{(k)}$ by (4.11) and (4.12) to obtain $\hat{\boldsymbol{\beta}}_{(k)}$, for $k = 1, \ldots, N$.
```

the point estimator of $\boldsymbol{\beta}$, and the 5% quantile and 95% quantile can be used to form the 90% confidence interval.

4.3. Theoretical Properties

In this section, we demonstrate that the de-biased fiducial selector exhibits model selection consistency and mean unbiasedness for the significant parameters under certain conditions. We begin by introducing the necessary notations and conditions from Zhao and Yu (2006).

Without loss of generality, let $\boldsymbol{\beta}^n = (\beta_1^n, \ldots, \beta_s^n, \beta_{s+1}^n, \ldots, \beta_p^n)^T$ represent the parameter vector, where the superscript n denotes the number of observations. Assume the true signal set $\mathcal{A} = \{1, \ldots, s\}$, such that $\beta_j^n \neq 0$ for $j = 1, \ldots, s$ and $\beta_j^n = 0$ for $j = s+1, \ldots, p$. Define $\boldsymbol{\beta}_{\mathcal{A}}^n = (\beta_1^n, \ldots, \beta_s^n)^T$ and $\boldsymbol{\beta}_{\mathcal{A}^c}^n = (\beta_{s+1}^n, \ldots, \beta_p^n)^T$. Let $\boldsymbol{X}_{\mathcal{A}}^n$ and $\boldsymbol{X}_{\mathcal{A}^c}^n$ represent the first s and last $p - s$ columns of the design matrix $\boldsymbol{X}^n$, respectively. Define $\boldsymbol{C}^n = \frac{1}{n}\boldsymbol{X}^{n\top}\boldsymbol{X}^n$ and partition $\boldsymbol{C}^n$ into four blocks as follows:

$$\boldsymbol{C}^n = \begin{pmatrix} \boldsymbol{C}_{11}^n & \boldsymbol{C}_{12}^n \\ \boldsymbol{C}_{21}^n & \boldsymbol{C}_{22}^n \end{pmatrix},$$

where $\boldsymbol{C}_{11}^n = \frac{1}{n}\boldsymbol{X}_{\mathcal{A}}^{n\top}\boldsymbol{X}_{\mathcal{A}}^n$, $\boldsymbol{C}_{12}^n = \frac{1}{n}\boldsymbol{X}_{\mathcal{A}}^{n\top}\boldsymbol{X}_{\mathcal{A}^c}^n$, $\boldsymbol{C}_{21}^n = \frac{1}{n}\boldsymbol{X}_{\mathcal{A}^c}^{n\top}\boldsymbol{X}_{\mathcal{A}}^n$, and $\boldsymbol{C}_{22}^n = \frac{1}{n}\boldsymbol{X}_{\mathcal{A}^c}^{n\top}\boldsymbol{X}_{\mathcal{A}^c}^n$. Assume $\boldsymbol{C}_{11}^n$ is invertible. The strong irrepresentable condition is defined as follows.

Definition 4.3.1 (Strong Irrepresentable Condition)**.** *There exists a positive constant vector* $\boldsymbol{\eta}$ *of size* $(p-s)\times 1$ *s.t. the following equation hold,*

$$|\boldsymbol{C}_{21}^n(\boldsymbol{C}_{11}^n)^{-1} sign(\boldsymbol{\beta}_{\mathcal{A}}^n)| \leq \mathbf{1}_{p-s} - \boldsymbol{\eta},$$

where the inequality holds element-wise.

The strong irrepresentable condition states that the correlation between the significant parameters and insignificant parameters should not be too large. Section 2.3 of Zhao and Yu (2006) provides some examples of correlation designs that satisfy the strong irrepresentable condition.

Now, we cite the regularity conditions that are necessary for the selection consistency theorems of the lasso. For the 'small p and s' cases (i.e., fixed p and s), we need the following two conditions

$$\boldsymbol{C}^n \to \boldsymbol{C}, \text{ as } n \to \infty, \tag{4.13}$$

where $\boldsymbol{C}$ is a positive definite matrix, and

$$\frac{1}{n}\max_{1\leq i\leq n}((\boldsymbol{x}_i^n)^T\boldsymbol{x}_i^n) \to 0, \text{ as } n \to \infty, \tag{4.14}$$

where $\boldsymbol{x}_i^n$ is the ith row of the design matrix $\boldsymbol{X}^n$.

For the 'large p and s' case (i.e., $p = p_n$ and $s = s_n$ are allowed to grow with n), we assume there exists $0 \leq c_1 \leq c_2 \leq 1$ and $M_1, M_2, M_3 > 0$ s.t. the following holds:

$$\frac{1}{n}(\boldsymbol{x}_i^n)^\top\boldsymbol{x}_i^n \leq M_1 \text{ for } \forall i, \tag{4.15}$$

$$\alpha^\top\boldsymbol{C}_{11}^n\alpha \geq M_2, \text{ for } \forall||\alpha||_2^2 = 1, \tag{4.16}$$

$$s_n = O(n^{c_1}), \tag{4.17}$$

$$n^{\frac{1-c_2}{2}} \min_{j=1,\dots,s} |\beta_j^n| \geq M_3. \tag{4.18}$$

THEOREM 4.3.2 (Selection Consistency). *For both "small p and s" and "large p and s" cases and arbitrary fixed fiducial sample size N, under the strong irrepresentable condition (Definition 4.3.1) and regularity conditions (4.13) and (4.14) (for "small p and s" case) and (4.15), (4.16), (4.17) and (4.18) (for "large p and s" case), the probability of the de-biased fiducial selector (4.11) and (4.12) identifying the correct set of significant parameters tends to 1 as the number of observations n goes to infinity. That is, For arbitrary fixed fiducial sample size m, we have*

$$P(\hat{\mathcal{A}} = \mathcal{A}) \to 1 \quad as \quad n \to \infty,$$

where $\hat{\mathcal{A}}$ is defined in equation (4.10) and $\mathcal{A}$ is the true signal set.

The proof of the theorem is delayed to Appendix A.

THEOREM 4.3.3 (Mean Unbiasedness). *If for $\forall j \in \hat{\mathcal{A}}^c$, we have $\beta_j = 0$. Then the mean of the de-biased fiducial sample (4.11) and (4.12) is an unbiased estimator of β_j for $\forall j \in \hat{\mathcal{A}}$.*

The proof of the theorem is delayed to Appendix A.

The above two theorems show that the de-biased fiducial selector can guarantee the model selection consistency and provide unbiased estimation for significant parameters at the same time, which is stronger than lasso's strong and general sign consistency defined in Zhao and Yu (2006).

4.4. Empirical Performance

4.4.1. Simulation Study. This section presents the results of a simulation study designed to evaluate the performance of Fiducial Selector and to compare it with existing methods. We employed the following model to generate simulated noisy data:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{e}, \tag{4.19}$$

where $\boldsymbol{e} \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I})$. In this setup, σ is set to 0.5. Let s represent the number of parameters in the signal set $\mathcal{A}$, which is a randomly sampled subset of $\{1, \cdots, p\}$. The signal parameters within $\mathcal{A}$ are drawn from a uniform distribution $U(0.2, 0.5)$, while the remaining null signal parameters are fixed at 0. Each row of $\boldsymbol{X}$ is generated independently from a multivariate normal distribution $\mathcal{N}(\mathbf{0}, 0.5\boldsymbol{R}(\rho))$, where $\boldsymbol{R}(\rho)$ is a first-order autoregressive correlation matrix with correlation parameter ρ. Furthermore, each column of $\boldsymbol{X}$ is normalized to have an ℓ_2-norm of 1.

We evaluated the following scenarios of combinations: $n \in \{300, 500\}$, $p \in \{400, 600, 1000\}$, $s \in \{5, 15\}$ and $\rho \in \{0.3, 0.5\}$. In total, we conducted experiments under 24 different settings. For each of these settings, 500 datasets were simulated using the linear model (4.19), with independent error terms. For each dataset, a fiducial sample of size 1000 was generated to estimate the model coefficients and construct confidence intervals. Our proposed method was compared with several contemporary approaches, including "MOCE", a post-model selection method from Wang et al. (2021); "pBLPR", a bootstrap lasso+partial ridge estimator by Liu et al. (2020); "LDPE" and "SSLasso", two well-known de-biased lasso methods introduced by Zhang and Zhang (2014) and Javanmard and Montanari (2014), respectively; and "AutoGFI" introduced by Du et al. (2024). Additionally, we employed the classical maximum likelihood estimate of the true model as a benchmark (termed the "Oracle" method), even though it is impractical to obtain in real-world scenarios.

To evaluate the performance of different methods, we use bias as a measure for both the signal and non-signal parameters to assess the accuracy of point estimators. Additionally, we evaluate the inferential performance of the methods by examining the confidence intervals they provide for the signal parameters. Specifically, we calculate the coverage probability and average width of the confidence intervals. These metrics are defined as follows:

$$\text{Bias}_{\mathcal{A}} = \frac{1}{s}\sum_{j\in\mathcal{A}}|\tilde{\beta}_j-\beta_j|, \quad \text{Bias}_{\mathcal{A}^c} = \frac{1}{p-s}\sum_{j\notin\mathcal{A}^c}|\tilde{\beta}_j-\beta_j|,$$

$$\text{CP}^{\alpha}_{\mathcal{A}} = \frac{1}{s}\sum_{j\in\mathcal{A}}\mathbb{1}\{\beta_j\in CI_j(\alpha)\}, \quad \text{Wid}^{\alpha}_{\mathcal{A}} = \frac{1}{s}\sum_{j\in\mathcal{A}}\text{width}(CI_j(\alpha)),$$

where $\tilde{\beta}_j$ is the point estimate of β_j, and $CI_j(\alpha)$ is the confidence interval for β_j under the confidence level $1-\alpha$.

The metric Bias for both signal and non-signal parameters is reported across 500 simulation replicates in Tables 4.1 to 4.4, which cover different experimental setups and methods. It is evident that, apart from the Oracle method, Fiducial Selector consistently shows the lowest $\text{Bias}_{\mathcal{A}}$ and $\text{Bias}_{\mathcal{A}^c}$. Furthermore, Figure 4.1 illustrates the absolute bias $|\tilde{\beta}_j-\beta_j|$ for each signal parameter β_j, which varies in magnitude, in cases with $s=5$ and 10. Regardless of signal magnitude, Fiducial Selector consistently achieves the lowest bias for all parameters. Note that we present only the configuration with $n=300$, $p=1000$, and $\rho=0.5$ here to limit the scope; similar trends in other settings are detailed in the Appendix B.

n	p		Oracle	Fiducial Selector	AutoGFI	LDPE	SSLasso	pBLPR	MOCE
300	400	$\mathcal{A}$	$0.0227_{0.0005}$	$0.0228_{0.0005}$	$0.0238_{0.0004}$	$0.0272_{0.0015}$	$0.0275_{0.0014}$	$0.0255_{0.0009}$	$0.0297_{0.0016}$
		$\mathcal{A}^c$	–	$0.0000_{0.0001}$	$0.0005_{0.0005}$	$0.0275_{0.0018}$	$0.0249_{0.0016}$	$0.0086_{0.0003}$	$0.0477_{0.0028}$
	600	$\mathcal{A}$	$0.0236_{0.0006}$	$0.0237_{0.0010}$	$0.0262_{0.0011}$	$0.0276_{0.0010}$	$0.0285_{0.0014}$	$0.0265_{0.0008}$	$0.0341_{0.0019}$
		$\mathcal{A}^c$	–	$0.0000_{0.0001}$	$0.0007_{0.0005}$	$0.0278_{0.0024}$	$0.0229_{0.0013}$	$0.0073_{0.0003}$	$0.0182_{0.0011}$
	1000	$\mathcal{A}$	$0.0230_{0.0011}$	$0.0235_{0.0008}$	$0.0276_{0.0005}$	$0.0286_{0.0016}$	$0.0316_{0.0030}$	$0.0257_{0.0011}$	$0.0393_{0.0031}$
		$\mathcal{A}^c$	–	$0.0000_{0.0001}$	$0.0006_{0.0004}$	$0.0280_{0.0027}$	$0.0195_{0.0012}$	$0.0054_{0.0003}$	$0.0092_{0.0011}$
500	400	$\mathcal{A}$	$0.0174_{0.0007}$	$0.0174_{0.0009}$	$0.0174_{0.0010}$	$0.0213_{0.0010}$	$0.0232_{0.0012}$	$0.0194_{0.0007}$	$0.0214_{0.0011}$
		$\mathcal{A}^c$	–	$0.0000_{0.0000}$	$0.0001_{0.0001}$	$0.0211_{0.0009}$	$0.0228_{0.0011}$	$0.0076_{0.0003}$	$0.0332_{0.0017}$
	600	$\mathcal{A}$	$0.0177_{0.0007}$	$0.0177_{0.0004}$	$0.0177_{0.0005}$	$0.0216_{0.0018}$	$0.0212_{0.0009}$	$0.0195_{0.0007}$	$0.0212_{0.0007}$
		$\mathcal{A}^c$	–	$0.0000_{0.0000}$	$0.0001_{0.0001}$	$0.0213_{0.0011}$	$0.0200_{0.0009}$	$0.0069_{0.0003}$	$0.0632_{0.0038}$
	1000	$\mathcal{A}$	$0.0181_{0.0007}$	$0.0180_{0.0008}$	$0.0182_{0.0007}$	$0.0213_{0.0007}$	$0.0215_{0.0013}$	$0.0203_{0.0011}$	$0.0227_{0.0007}$
		$\mathcal{A}^c$	–	$0.0000_{0.0000}$	$0.0001_{0.0001}$	$0.0214_{0.0013}$	$0.0181_{0.0008}$	$0.0057_{0.0002}$	$0.0141_{0.0007}$

TABLE 4.1. In the scenario of $s=5$ and $\rho=0.3$, averages of the biases $\text{Bias}_{\mathcal{A}}$ and $\text{Bias}_{\mathcal{A}^c}$ for different methods across 500 simulation rounds, with standard errors as subscripts.

Tables 4.5 to 4.8 present the metrics $\text{CP}^{\alpha}_{\mathcal{A}}$ and $\text{Wid}^{\alpha}_{\mathcal{A}}$ for confidence intervals constructed by various methods across different scenarios, with α values of 0.1, 0.05, and 0.01. Generally,

n	p		Oracle	Fiducial Selector	AutoGFI	LDPE	SSLasso	pBLPR	MOCE
300	400	$\mathcal{A}$	$0.0231_{0.0005}$	$0.0233_{0.0004}$	$0.0253_{0.0021}$	$0.0298_{0.0013}$	$0.0302_{0.0006}$	$0.0265_{0.0006}$	$0.0323_{0.0021}$
		$\mathcal{A}^c$	–	$0.0000_{0.0001}$	$0.0007_{0.0006}$	$0.0309_{0.0024}$	$0.0280_{0.0019}$	$0.0084_{0.0004}$	$0.0580_{0.0036}$
	600	$\mathcal{A}$	$0.0225_{0.0007}$	$0.0233_{0.0012}$	$0.0258_{0.0017}$	$0.0318_{0.0037}$	$0.0304_{0.0012}$	$0.0262_{0.0005}$	$0.0332_{0.0026}$
		$\mathcal{A}^c$	–	$0.0000_{0.0003}$	$0.0005_{0.0008}$	$0.0308_{0.0023}$	$0.0253_{0.0018}$	$0.0072_{0.0003}$	$0.0202_{0.0013}$
	1000	$\mathcal{A}$	$0.0236_{0.0007}$	$0.0243_{0.0007}$	$0.0311_{0.0018}$	$0.0345_{0.0067}$	$0.0348_{0.0058}$	$0.0271_{0.0003}$	$0.0408_{0.0030}$
		$\mathcal{A}^c$	–	$0.0000_{0.0002}$	$0.0006_{0.0006}$	$0.0312_{0.0031}$	$0.0231_{0.0016}$	$0.0054_{0.0003}$	$0.0096_{0.0008}$
500	400	$\mathcal{A}$	$0.0181_{0.0002}$	$0.0182_{0.0004}$	$0.0186_{0.0005}$	$0.0236_{0.0008}$	$0.0259_{0.0002}$	$0.0206_{0.0003}$	$0.0234_{0.0009}$
		$\mathcal{A}^c$	–	$0.0000_{0.0001}$	$0.0001_{0.0004}$	$0.0238_{0.0011}$	$0.0252_{0.0012}$	$0.0073_{0.0003}$	$0.0389_{0.0024}$
	600	$\mathcal{A}$	$0.0184_{0.0008}$	$0.0184_{0.0008}$	$0.0189_{0.0008}$	$0.0248_{0.0009}$	$0.0253_{0.0004}$	$0.0212_{0.0004}$	$0.0249_{0.0015}$
		$\mathcal{A}^c$	–	$0.0000_{0.0001}$	$0.0001_{0.0004}$	$0.0240_{0.0013}$	$0.0230_{0.0013}$	$0.0067_{0.0003}$	$0.0752_{0.0050}$
	1000	$\mathcal{A}$	$0.0176_{0.0010}$	$0.0175_{0.0004}$	$0.0177_{0.0005}$	$0.0245_{0.0017}$	$0.0232_{0.0003}$	$0.0208_{0.0006}$	$0.0238_{0.0010}$
		$\mathcal{A}^c$	–	$0.0000_{0.0000}$	$0.0001_{0.0002}$	$0.0240_{0.0015}$	$0.0203_{0.0013}$	$0.0056_{0.0002}$	$0.0156_{0.0008}$

TABLE 4.2. In the scenario of $s = 5$ and $\rho = 0.5$, averages of the biases $\text{Bias}_{\mathcal{A}}$ and $\text{Bias}_{\mathcal{A}^c}$ for different methods across 500 simulation rounds, with standard errors as subscripts.

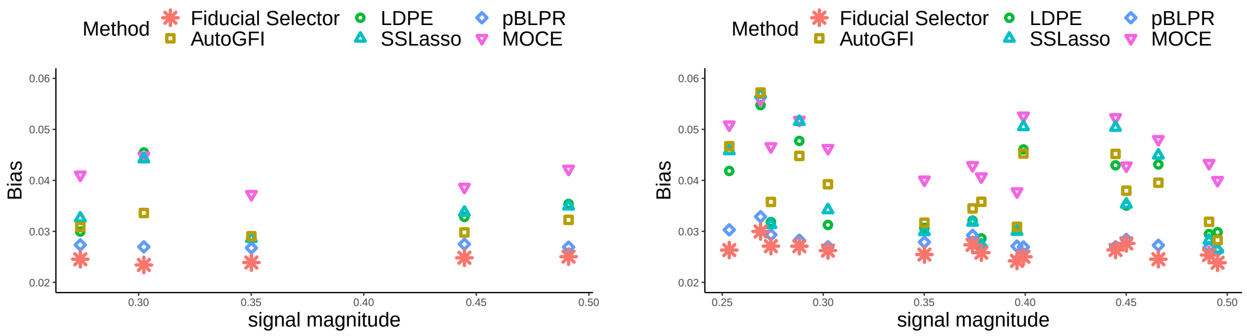
n	p		Oracle	Fiducial Selector	AutoGFI	LDPE	SSLasso	pBLPR	MOCE
300	400	$\mathcal{A}$	$0.0235_{0.0008}$	$0.0241_{0.0007}$	$0.0273_{0.0017}$	$0.0302_{0.0038}$	$0.0306_{0.0035}$	$0.0263_{0.0011}$	$0.0341_{0.0025}$
		$\mathcal{A}^c$	–	$0.0003_{0.0011}$	$0.0017_{0.0019}$	$0.0275_{0.0030}$	$0.0258_{0.0026}$	$0.0087_{0.0005}$	$0.0751_{0.0052}$
	600	$\mathcal{A}$	$0.0233_{0.0009}$	$0.0241_{0.0007}$	$0.0282_{0.0025}$	$0.0314_{0.0051}$	$0.0310_{0.0042}$	$0.0267_{0.0007}$	$0.0355_{0.0032}$
		$\mathcal{A}^c$	–	$0.0002_{0.0006}$	$0.0015_{0.0015}$	$0.0276_{0.0034}$	$0.0239_{0.0025}$	$0.0073_{0.0005}$	$0.0187_{0.0012}$
	1000	$\mathcal{A}$	$0.0235_{0.0011}$	$0.0247_{0.0011}$	$0.0321_{0.0038}$	$0.0297_{0.0050}$	$0.0312_{0.0055}$	$0.0262_{0.0011}$	$0.0385_{0.0038}$
		$\mathcal{A}^c$	–	$0.0002_{0.0007}$	$0.0012_{0.0014}$	$0.0278_{0.0043}$	$0.0205_{0.0028}$	$0.0054_{0.0005}$	$0.0087_{0.0011}$
500	400	$\mathcal{A}$	$0.0181_{0.0006}$	$0.0183_{0.0005}$	$0.0183_{0.0005}$	$0.0217_{0.0012}$	$0.0239_{0.0009}$	$0.0196_{0.0006}$	$0.0231_{0.0008}$
		$\mathcal{A}^c$	–	$0.0000_{0.0002}$	$0.0003_{0.0006}$	$0.0212_{0.0013}$	$0.0229_{0.0012}$	$0.0076_{0.0003}$	$0.0289_{0.0013}$
	600	$\mathcal{A}$	$0.0179_{0.0006}$	$0.0180_{0.0006}$	$0.0184_{0.0006}$	$0.0221_{0.0013}$	$0.0225_{0.0010}$	$0.0199_{0.0004}$	$0.0243_{0.0015}$
		$\mathcal{A}^c$	–	$0.0000_{0.0001}$	$0.0003_{0.0006}$	$0.0214_{0.0017}$	$0.0203_{0.0012}$	$0.0069_{0.0003}$	$0.1047_{0.0087}$
	1000	$\mathcal{A}$	$0.0183_{0.0008}$	$0.0185_{0.0008}$	$0.0199_{0.0015}$	$0.0232_{0.0025}$	$0.0232_{0.0021}$	$0.0203_{0.0010}$	$0.0282_{0.0023}$
		$\mathcal{A}^c$	–	$0.0000_{0.0001}$	$0.0004_{0.0006}$	$0.0215_{0.0022}$	$0.0186_{0.0015}$	$0.0057_{0.0002}$	$0.0144_{0.0008}$

TABLE 4.3. In the scenario of $s = 15$ and $\rho = 0.3$, averages of the biases $\text{Bias}_{\mathcal{A}}$ and $\text{Bias}_{\mathcal{A}^c}$ for different methods across 500 simulation rounds, with standard errors as subscripts.

Fiducial Selector maintains an empirical coverage probability close to the target confidence level for the significant parameters, while also offering the narrowest confidence intervals. However, all methods struggle with the challenging scenario of $n = 300$, $p = 1000$, $\rho = 0.5$, and $s = 15$, significantly under-covering. Still, the coverage probabilities of Fiducial Selector's confidence intervals remain closest to the target level. Turning to the other methods, AutoGFI demonstrates competitive performance, particularly in cases with larger sample

n	p		Oracle	Fiducial Selector	AutoGFI	LDPE	SSLasso	pBLPR	MOCE
300	400	$\mathcal{A}$	$0.0236_{0.0006}$	$0.0248_{0.0011}$	$0.0280_{0.0025}$	$0.0313_{0.0022}$	$0.0317_{0.0021}$	$0.0276_{0.0012}$	$0.0353_{0.0022}$
		$\mathcal{A}^c$	–	$0.0003_{0.0010}$	$0.0016_{0.0019}$	$0.0307_{0.0033}$	$0.0286_{0.0026}$	$0.0085_{0.0008}$	$0.0807_{0.0064}$
	600	$\mathcal{A}$	$0.0235_{0.0008}$	$0.0272_{0.0060}$	$0.0350_{0.0091}$	$0.0350_{0.0065}$	$0.0352_{0.0067}$	$0.0290_{0.0020}$	$0.0398_{0.0036}$
		$\mathcal{A}^c$	–	$0.0003_{0.0009}$	$0.0016_{0.0019}$	$0.0307_{0.0041}$	$0.0265_{0.0032}$	$0.0074_{0.0010}$	$0.0212_{0.0015}$
	1000	$\mathcal{A}$	$0.0235_{0.0011}$	$0.0262_{0.0016}$	$0.0390_{0.0077}$	$0.0370_{0.0083}$	$0.0383_{0.0105}$	$0.0281_{0.0017}$	$0.0461_{0.0056}$
		$\mathcal{A}^c$	–	$0.0002_{0.0008}$	$0.0014_{0.0017}$	$0.0311_{0.0053}$	$0.0245_{0.0036}$	$0.0054_{0.0007}$	$0.0094_{0.0013}$
500	400	$\mathcal{A}$	$0.0186_{0.0012}$	$0.0191_{0.0013}$	$0.0197_{0.0012}$	$0.0250_{0.0014}$	$0.0265_{0.0012}$	$0.0210_{0.0011}$	$0.0251_{0.0017}$
		$\mathcal{A}^c$	–	$0.0001_{0.0003}$	$0.0005_{0.0010}$	$0.0238_{0.0014}$	$0.0253_{0.0013}$	$0.0073_{0.0003}$	$0.0341_{0.0020}$
	600	$\mathcal{A}$	$0.0181_{0.0006}$	$0.0188_{0.0013}$	$0.0197_{0.0015}$	$0.0252_{0.0023}$	$0.0260_{0.0024}$	$0.0208_{0.0010}$	$0.0271_{0.0016}$
		$\mathcal{A}^c$	–	$0.0001_{0.0006}$	$0.0004_{0.0012}$	$0.0240_{0.0019}$	$0.0232_{0.0015}$	$0.0067_{0.0004}$	$0.1259_{0.0114}$
	1000	$\mathcal{A}$	$0.0185_{0.0012}$	$0.0188_{0.0012}$	$0.0196_{0.0013}$	$0.0247_{0.0012}$	$0.0232_{0.0012}$	$0.0213_{0.0014}$	$0.0263_{0.0008}$
		$\mathcal{A}^c$	–	$0.0000_{0.0002}$	$0.0003_{0.0006}$	$0.0241_{0.0022}$	$0.0207_{0.0016}$	$0.0056_{0.0002}$	$0.0160_{0.0009}$

TABLE 4.4. In the scenario of $s = 15$ and $\rho = 0.5$, averages of the biases $\text{Bias}_{\mathcal{A}}$ and $\text{Bias}_{\mathcal{A}^c}$ for different methods across 500 simulation rounds, with standard errors as subscripts.



(a)

(b)

FIGURE 4.1. Scenario: $n = 300$, $p = 1000$, $\rho = 0.5$. Relationship between signal magnitude of β_j and absolute bias $|\beta_j - \tilde{\beta}_j|$ for all signal parameters, i.e. $j \in \mathcal{A}$. (a) presents the case when the number of signals is $s = 5$, while (b) corresponds to $s = 15$.

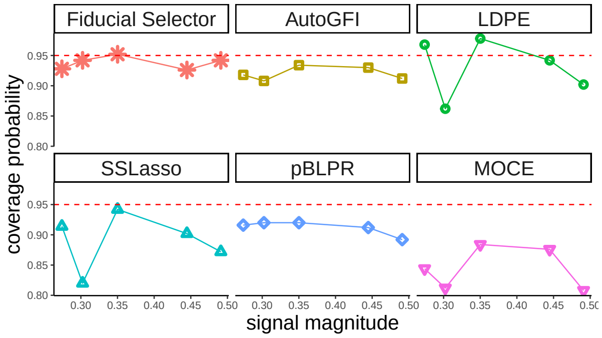
sizes, such as $n = 500$, or in lower dimensional scenarios with p smaller than 1000. However, it encounters severe under-coverage issues in more challenging environments characterized by smaller sample sizes and higher dimensionality, for example, $n = 300$ and $p = 1000$. This limitation arises because AutoGFI is designed for broad applicability in complex models, which may compromise its de-biasing effectiveness. The general approach employed by AutoGFI can lead to missing signals or the excessive de-biasing of noise. Consequently, AutoGFI tends to underperform in these demanding situations. LDPE also displays strong coverage performance, with rates nearly at the target level. However, compared to Fiducial Selector, LDPE tends to produce wider confidence intervals, occasionally leading to over-coverage. This is due to LDPE's strategy of optimizing global coverage across all parameters, without distinguishing between significant and null signals, thereby incurring larger biases and wider intervals. SSLasso faces similar issues but exhibits more instability; it tends to over-cover in scenarios with large n but under-cover with large p. Both pBLPR and MOCE generally underperform, predominantly showing under-coverage in most scenarios.

To examine the impact of the signal magnitude on empirical coverage probabilities and interval widths, Figure 4.2a and Figure 4.3a displays the coverage probability versus signal magnitude for each signal parameter in the cases with $s = 5$ and $s = 15$, respectively. It is evident that Fiducial Selector's coverage probabilities remain close to the target level across varying signal magnitudes, whereas other methods tend to under-cover smaller signals. Similarly, Figure 4.2b and Figure 4.3b illustrate how interval widths vary with signal magnitude for the cases with $s = 5$ and $s = 15$. The results indicate that Fiducial Selector consistently provides the narrowest confidence intervals for all signal parameters. Figure 4.4 further illustrates this finding by plotting coverage probabilities against interval widths for all signal parameters. Note that we display only the most challenging scenarios with $n = 300$, $p = 1000$, $\rho = 0.5$ for $s = 5$ and $s = 15$ here to conserve space. Similar trends observed in other cases are detailed in Appendix B.

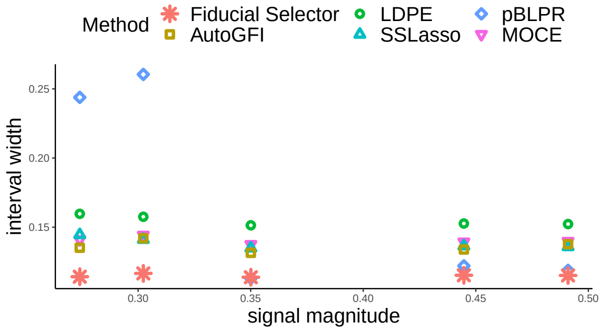
Another advantage of Fiducial Selector is its computational efficiency. Tables 4.9 and 4.10 report the average computation time per replicate across different values of n and p, for $s = 5$ and $s = 15$, respectively. The results clearly demonstrate that Fiducial Selector is the most efficient method. While we have presented results for $\rho = 0.3$, similar performance is observed the case with $\rho = 0.5$. It is noteworthy that the average computation time for some methods increases with smaller p values. For instance, in Table 4.9, the computation time for MOCE is greater at $n = 500$ and $p = 600$ than at $n = 500$ and $p = 1000$. This counter-intuitive result can be explained by the increased sparsity of the target parameter

n	p	$1-\alpha$	Oracle	Fiducial Selector	AutoGFI	LDPE	SSLasso	pBLPR	MOCE
300	400	90%	$0.907_{0.096}$	$0.907_{0.096}$	$0.920_{0.103}$	$0.924_{0.120}$	$0.908_{0.118}$	$0.876_{0.133}$	$0.843_{0.104}$
		95%	$0.956_{0.114}$	$0.951_{0.114}$	$0.957_{0.123}$	$0.958_{0.143}$	$0.946_{0.137}$	$0.936_{0.155}$	$0.907_{0.124}$
		99%	$0.989_{0.150}$	$0.988_{0.148}$	$0.991_{0.163}$	$0.993_{0.189}$	$0.989_{0.175}$	$0.981_{0.198}$	$0.969_{0.163}$
	600	90%	$0.894_{0.096}$	$0.889_{0.096}$	$0.888_{0.106}$	$0.917_{0.118}$	$0.884_{0.112}$	$0.864_{0.125}$	$0.790_{0.107}$
		95%	$0.946_{0.114}$	$0.943_{0.114}$	$0.944_{0.126}$	$0.958_{0.141}$	$0.930_{0.129}$	$0.922_{0.144}$	$0.868_{0.128}$
		99%	$0.990_{0.150}$	$0.987_{0.148}$	$0.985_{0.165}$	$0.994_{0.185}$	$0.979_{0.163}$	$0.975_{0.186}$	$0.957_{0.168}$
	1000	90%	$0.897_{0.096}$	$0.886_{0.095}$	$0.880_{0.108}$	$0.900_{0.119}$	$0.849_{0.111}$	$0.843_{0.139}$	$0.763_{1426.525}$
		95%	$0.952_{0.114}$	$0.940_{0.113}$	$0.935_{0.129}$	$0.953_{0.142}$	$0.894_{0.126}$	$0.907_{0.162}$	$0.850_{1699.809}$
		99%	$0.991_{0.150}$	$0.986_{0.147}$	$0.984_{0.169}$	$0.991_{0.187}$	$0.952_{0.155}$	$0.967_{0.206}$	$0.949_{2233.928}$
500	400	90%	$0.908_{0.074}$	$0.904_{0.074}$	$0.906_{0.075}$	$0.905_{0.090}$	$0.905_{0.097}$	$0.880_{0.102}$	$0.848_{0.077}$
		95%	$0.950_{0.088}$	$0.953_{0.088}$	$0.956_{0.089}$	$0.955_{0.107}$	$0.953_{0.114}$	$0.933_{0.121}$	$0.909_{0.092}$
		99%	$0.990_{0.116}$	$0.989_{0.114}$	$0.986_{0.116}$	$0.994_{0.140}$	$0.990_{0.148}$	$0.981_{0.154}$	$0.974_{0.121}$
	600	90%	$0.911_{0.074}$	$0.906_{0.074}$	$0.911_{0.075}$	$0.899_{0.090}$	$0.899_{0.088}$	$0.879_{0.118}$	$0.866_{0.078}$
		95%	$0.954_{0.088}$	$0.952_{0.088}$	$0.959_{0.089}$	$0.946_{0.107}$	$0.943_{0.104}$	$0.932_{0.138}$	$0.924_{0.093}$
		99%	$0.988_{0.116}$	$0.987_{0.114}$	$0.988_{0.116}$	$0.990_{0.141}$	$0.985_{0.133}$	$0.982_{0.169}$	$0.978_{0.123}$
	1000	90%	$0.894_{0.074}$	$0.901_{0.074}$	$0.904_{0.076}$	$0.909_{0.091}$	$0.898_{0.088}$	$0.858_{0.103}$	$0.842_{0.081}$
		95%	$0.951_{0.088}$	$0.943_{0.088}$	$0.948_{0.090}$	$0.953_{0.108}$	$0.937_{0.102}$	$0.914_{0.119}$	$0.912_{0.096}$
		99%	$0.991_{0.115}$	$0.988_{0.114}$	$0.991_{0.118}$	$0.993_{0.142}$	$0.986_{0.130}$	$0.972_{0.147}$	$0.973_{0.127}$

TABLE 4.5. In scenario of $s = 5$ and $\rho = 0.3$, average empirical coverage probabilities $\text{CP}_{\mathcal{A}}^{\alpha}$, with interval widths $\text{Wid}_{\mathcal{A}}^{\alpha}$ as subscripts, for the signal parameters across 500 simulation rounds. The table includes results for target levels $\alpha = 0.1, 0.05$, and 0.01.



(a)



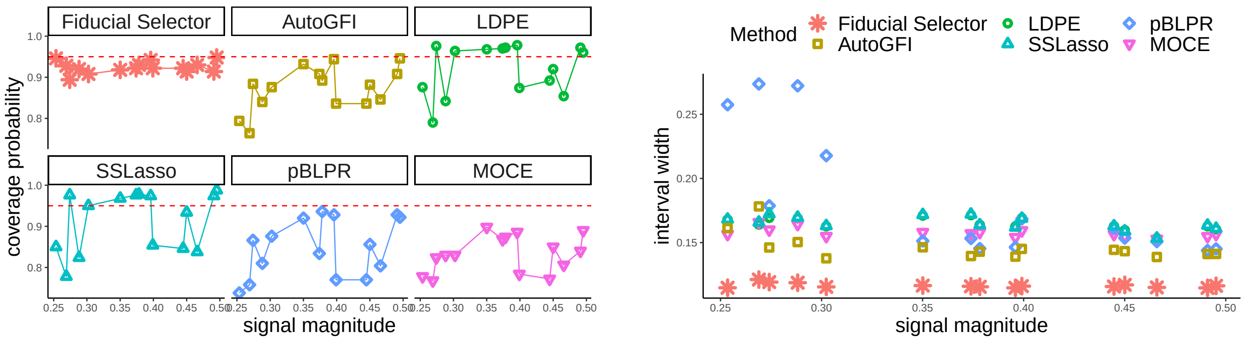
(b)

FIGURE 4.2. Scenario: $n = 300$, $p = 1000$, $\rho = 0.5$ and $s = 5$. Relationship between signal magnitude of β_j and empirical coverage probability / interval width for all signal parameters, i.e. $j \in \mathcal{A}$.

$\boldsymbol{\beta}$ at $p = 1000$ compared to $p = 600$. The higher sparsity results in a larger penalty term

n	p	$1-\alpha$	Oracle	Fiducial Selector	AutoGFI	LDPE	SSLasso	pBLPR	MOCE
300	400	90%	$0.891_{0.096}$	$0.899_{0.095}$	$0.914_{0.111}$	$0.925_{0.132}$	$0.911_{0.129}$	$0.870_{0.126}$	$0.816_{0.108}$
		95%	$0.952_{0.114}$	$0.951_{0.113}$	$0.959_{0.134}$	$0.964_{0.157}$	$0.953_{0.151}$	$0.928_{0.148}$	$0.890_{0.129}$
		99%	$0.992_{0.150}$	$0.987_{0.146}$	$0.991_{0.177}$	$0.993_{0.206}$	$0.988_{0.193}$	$0.981_{0.191}$	$0.966_{0.169}$
	600	90%	$0.905_{0.096}$	$0.894_{0.096}$	$0.909_{0.109}$	$0.895_{0.131}$	$0.896_{0.127}$	$0.872_{0.125}$	$0.812_{0.110}$
		95%	$0.950_{0.114}$	$0.944_{0.114}$	$0.950_{0.130}$	$0.952_{0.156}$	$0.944_{0.148}$	$0.925_{0.144}$	$0.888_{0.131}$
		99%	$0.991_{0.150}$	$0.988_{0.149}$	$0.990_{0.171}$	$0.990_{0.205}$	$0.984_{0.188}$	$0.978_{0.182}$	$0.962_{0.172}$
	1000	90%	$0.899_{0.096}$	$0.880_{0.097}$	$0.857_{0.114}$	$0.869_{0.130}$	$0.840_{0.121}$	$0.844_{0.150}$	$0.747_{0.118}$
		95%	$0.948_{0.114}$	$0.938_{0.115}$	$0.920_{0.136}$	$0.930_{0.155}$	$0.890_{0.139}$	$0.912_{0.172}$	$0.845_{0.140}$
		99%	$0.988_{0.150}$	$0.982_{0.150}$	$0.982_{0.179}$	$0.982_{0.203}$	$0.956_{0.173}$	$0.971_{0.211}$	$0.941_{0.185}$
500	400	90%	$0.898_{0.074}$	$0.886_{0.074}$	$0.898_{0.078}$	$0.902_{0.101}$	$0.897_{0.108}$	$0.870_{0.103}$	$0.829_{0.081}$
		95%	$0.947_{0.088}$	$0.939_{0.088}$	$0.953_{0.094}$	$0.950_{0.120}$	$0.944_{0.127}$	$0.928_{0.121}$	$0.903_{0.097}$
		99%	$0.991_{0.115}$	$0.987_{0.114}$	$0.990_{0.124}$	$0.992_{0.158}$	$0.984_{0.165}$	$0.980_{0.152}$	$0.974_{0.127}$
	600	90%	$0.898_{0.074}$	$0.888_{0.074}$	$0.906_{0.080}$	$0.905_{0.102}$	$0.901_{0.103}$	$0.878_{0.108}$	$0.820_{0.083}$
		95%	$0.951_{0.088}$	$0.935_{0.088}$	$0.950_{0.096}$	$0.952_{0.122}$	$0.946_{0.121}$	$0.923_{0.125}$	$0.890_{0.099}$
		99%	$0.992_{0.115}$	$0.984_{0.114}$	$0.989_{0.125}$	$0.989_{0.160}$	$0.986_{0.156}$	$0.980_{0.158}$	$0.964_{0.130}$
	1000	90%	$0.904_{0.074}$	$0.906_{0.074}$	$0.910_{0.079}$	$0.900_{0.100}$	$0.916_{0.099}$	$0.861_{0.108}$	$0.831_{0.083}$
		95%	$0.955_{0.088}$	$0.951_{0.088}$	$0.962_{0.094}$	$0.950_{0.120}$	$0.956_{0.115}$	$0.927_{0.125}$	$0.906_{0.099}$
		99%	$0.992_{0.116}$	$0.990_{0.115}$	$0.994_{0.123}$	$0.990_{0.157}$	$0.987_{0.147}$	$0.976_{0.155}$	$0.974_{0.131}$

TABLE 4.6. In scenario of $s = 5$ and $\rho = 0.5$, average empirical coverage probabilities $\mathrm{CP}_{\mathcal{A}}^{\alpha}$, with interval widths $\mathrm{Wid}_{\mathcal{A}}^{\alpha}$ as subscripts, for the signal parameters across 500 simulation rounds. The table includes results for target levels $\alpha = 0.1, 0.05$, and 0.01.



(a)

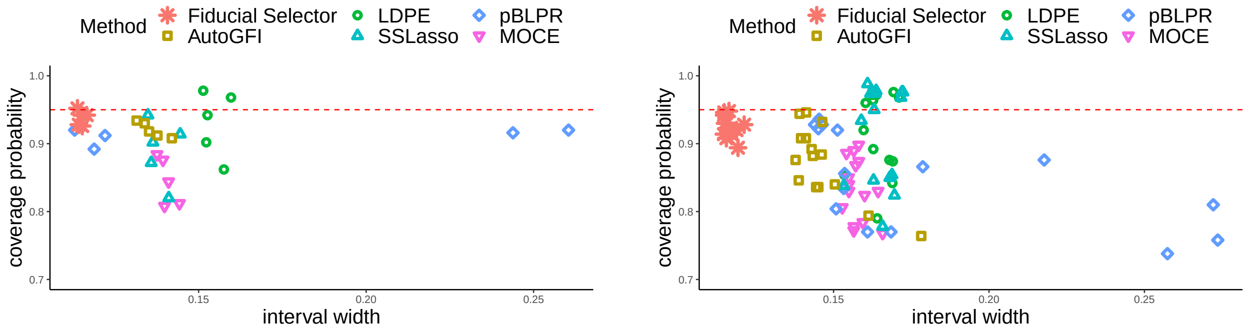
(b)

FIGURE 4.3. Scenario: $n = 300$, $p = 1000$, $\rho = 0.5$ and $s = 15$. Relationship between signal magnitude of β_j and empirical coverage probability / interval width for all signal parameters, i.e. $j \in \mathcal{A}$.

during the regression process, which in turn accelerates computation despite the increase in dimensionality.

n	p	$1-\alpha$	Oracle	Fiducial Selector	AutoGFI	LDPE	SSLasso	pBLPR	MOCE
300	400	90%	$0.903_{0.098}$	$0.890_{0.099}$	$0.887_{0.110}$	$0.889_{0.122}$	$0.911_{0.131}$	$0.878_{0.141}$	$0.811_{0.112}$
		95%	$0.952_{0.117}$	$0.941_{0.117}$	$0.942_{0.132}$	$0.944_{0.146}$	$0.950_{0.151}$	$0.933_{0.169}$	$0.886_{0.134}$
		99%	$0.991_{0.153}$	$0.988_{0.152}$	$0.986_{0.174}$	$0.988_{0.192}$	$0.986_{0.190}$	$0.981_{0.224}$	$0.956_{0.175}$
	600	90%	$0.901_{0.097}$	$0.891_{0.098}$	$0.886_{0.112}$	$0.889_{0.124}$	$0.914_{0.132}$	$0.859_{0.141}$	$0.806_{0.116}$
		95%	$0.951_{0.116}$	$0.943_{0.116}$	$0.940_{0.133}$	$0.945_{0.148}$	$0.950_{0.151}$	$0.918_{0.169}$	$0.875_{0.138}$
		99%	$0.991_{0.152}$	$0.987_{0.151}$	$0.987_{0.175}$	$0.987_{0.194}$	$0.986_{0.187}$	$0.978_{0.221}$	$0.961_{0.181}$
	1000	90%	$0.902_{0.097}$	$0.880_{0.098}$	$0.849_{0.116}$	$0.899_{0.124}$	$0.902_{0.129}$	$0.842_{0.149}$	$0.806_{284.052}$
		95%	$0.953_{0.116}$	$0.936_{0.116}$	$0.916_{0.138}$	$0.949_{0.148}$	$0.936_{0.145}$	$0.904_{0.178}$	$0.880_{338.469}$
		99%	$0.992_{0.152}$	$0.983_{0.151}$	$0.977_{0.180}$	$0.989_{0.194}$	$0.974_{0.175}$	$0.969_{0.229}$	$0.961_{444.824}$
500	400	90%	$0.902_{0.075}$	$0.901_{0.075}$	$0.905_{0.077}$	$0.905_{0.092}$	$0.911_{0.103}$	$0.884_{0.088}$	$0.838_{0.081}$
		95%	$0.948_{0.089}$	$0.948_{0.090}$	$0.951_{0.092}$	$0.954_{0.109}$	$0.952_{0.121}$	$0.936_{0.105}$	$0.904_{0.096}$
		99%	$0.991_{0.117}$	$0.988_{0.116}$	$0.990_{0.120}$	$0.991_{0.144}$	$0.990_{0.157}$	$0.982_{0.136}$	$0.970_{0.127}$
	600	90%	$0.904_{0.075}$	$0.900_{0.075}$	$0.908_{0.079}$	$0.909_{0.093}$	$0.921_{0.099}$	$0.875_{0.099}$	$0.827_{0.083}$
		95%	$0.955_{0.089}$	$0.951_{0.089}$	$0.956_{0.094}$	$0.953_{0.111}$	$0.958_{0.115}$	$0.934_{0.118}$	$0.895_{0.099}$
		99%	$0.992_{0.117}$	$0.990_{0.116}$	$0.992_{0.122}$	$0.990_{0.146}$	$0.989_{0.146}$	$0.983_{0.153}$	$0.969_{0.130}$
	1000	90%	$0.906_{0.075}$	$0.900_{0.075}$	$0.900_{0.082}$	$0.887_{0.093}$	$0.907_{0.097}$	$0.862_{0.095}$	$0.781_{0.087}$
		95%	$0.952_{0.090}$	$0.950_{0.090}$	$0.950_{0.098}$	$0.943_{0.111}$	$0.947_{0.111}$	$0.924_{0.114}$	$0.859_{0.103}$
		99%	$0.990_{0.118}$	$0.989_{0.117}$	$0.989_{0.129}$	$0.989_{0.146}$	$0.985_{0.139}$	$0.977_{0.150}$	$0.952_{0.136}$

TABLE 4.7. In scenario of $s = 15$ and $\rho = 0.3$, average empirical coverage probabilities $\text{CP}^{\alpha}_{\mathcal{A}}$, with interval widths $\text{Wid}^{\alpha}_{\mathcal{A}}$ as subscripts, for the signal parameters across 500 simulation rounds. The table includes results for target levels $\alpha = 0.1, 0.05$, and 0.01.



(a)

(b)

FIGURE 4.4. Scenario: $n = 300$, $p = 1000$, $\rho = 0.5$. Relationship between empirical coverage probability and interval width for each signal parameters. (a) presents the case when the number of signals is $s = 5$, while (b) corresponds to $s = 15$.

n	p	$1-\alpha$	Oracle	Fiducial Selector	AutoGFI	LDPE	SSLasso	pBLPR	MOCE
300	400	90%	$0.905_{0.097}$	$0.888_{0.100}$	$0.892_{0.114}$	$0.912_{0.134}$	$0.930_{0.144}$	$0.880_{0.123}$	$0.815_{0.116}$
		95%	$0.951_{0.116}$	$0.941_{0.119}$	$0.943_{0.136}$	$0.959_{0.160}$	$0.963_{0.167}$	$0.931_{0.148}$	$0.885_{0.138}$
		99%	$0.991_{0.152}$	$0.985_{0.154}$	$0.987_{0.178}$	$0.991_{0.210}$	$0.991_{0.210}$	$0.982_{0.198}$	$0.967_{0.181}$
	600	90%	$0.904_{0.097}$	$0.876_{0.097}$	$0.855_{0.126}$	$0.879_{0.136}$	$0.906_{0.147}$	$0.824_{0.140}$	$0.776_{0.123}$
		95%	$0.952_{0.116}$	$0.930_{0.115}$	$0.922_{0.151}$	$0.934_{0.162}$	$0.943_{0.168}$	$0.898_{0.167}$	$0.855_{0.146}$
		99%	$0.990_{0.152}$	$0.973_{0.148}$	$0.981_{0.195}$	$0.983_{0.214}$	$0.982_{0.208}$	$0.970_{0.219}$	$0.954_{0.192}$
	1000	90%	$0.899_{0.097}$	$0.865_{0.098}$	$0.782_{0.122}$	$0.859_{0.138}$	$0.871_{0.146}$	$0.759_{0.154}$	$0.747_{0.132}$
		95%	$0.948_{0.116}$	$0.924_{0.117}$	$0.873_{0.146}$	$0.921_{0.165}$	$0.914_{0.165}$	$0.848_{0.181}$	$0.833_{0.158}$
		99%	$0.990_{0.152}$	$0.976_{0.151}$	$0.957_{0.192}$	$0.981_{0.216}$	$0.968_{0.202}$	$0.947_{0.234}$	$0.940_{0.207}$
500	400	90%	$0.900_{0.077}$	$0.891_{0.077}$	$0.905_{0.084}$	$0.904_{0.103}$	$0.917_{0.114}$	$0.881_{0.089}$	$0.832_{0.086}$
		95%	$0.949_{0.091}$	$0.945_{0.092}$	$0.955_{0.100}$	$0.953_{0.123}$	$0.958_{0.134}$	$0.937_{0.106}$	$0.902_{0.103}$
		99%	$0.989_{0.120}$	$0.987_{0.120}$	$0.990_{0.132}$	$0.990_{0.162}$	$0.990_{0.172}$	$0.982_{0.141}$	$0.971_{0.135}$
	600	90%	$0.899_{0.075}$	$0.890_{0.076}$	$0.897_{0.082}$	$0.901_{0.105}$	$0.917_{0.113}$	$0.879_{0.097}$	$0.807_{0.088}$
		95%	$0.947_{0.090}$	$0.944_{0.090}$	$0.950_{0.098}$	$0.951_{0.125}$	$0.953_{0.131}$	$0.932_{0.117}$	$0.879_{0.105}$
		99%	$0.990_{0.118}$	$0.986_{0.118}$	$0.989_{0.128}$	$0.989_{0.164}$	$0.988_{0.167}$	$0.981_{0.154}$	$0.961_{0.138}$
	1000	90%	$0.894_{0.076}$	$0.890_{0.076}$	$0.906_{0.083}$	$0.908_{0.105}$	$0.941_{0.110}$	$0.871_{0.096}$	$0.828_{0.089}$
		95%	$0.950_{0.090}$	$0.942_{0.090}$	$0.953_{0.099}$	$0.955_{0.125}$	$0.968_{0.126}$	$0.930_{0.115}$	$0.898_{0.106}$
		99%	$0.991_{0.118}$	$0.988_{0.118}$	$0.991_{0.130}$	$0.993_{0.164}$	$0.993_{0.157}$	$0.981_{0.152}$	$0.965_{0.139}$

Table 4.8. In scenario of $s = 15$ and $\rho = 0.5$, average empirical coverage probabilities $\mathrm{CP}^{\alpha}_{\mathcal{A}}$, with interval widths $\mathrm{Wid}^{\alpha}_{\mathcal{A}}$ as subscripts, for the signal parameters across 500 simulation rounds. The table includes results for target levels $\alpha = 0.1, 0.05$, and 0.01.

n	p	Fiducial Selector	AutoGFI	LDPE	SSLasso	pBLPR	MOCE
300	400	5.82	6.34	636.68	91.73	91.76	6.67
	600	7.72	8.67	508.80	370.94	289.29	11.18
	1000	12.86	14.92	892.79	828.21	789.41	37.27
500	400	18.76	21.03	1363.77	242.84	376.35	16.24
	600	19.60	22.26	909.13	576.93	469.85	87.27
	1000	46.41	53.53	1459.78	1095.29	1004.13	65.27

Table 4.9. In scenario of $s = 5$ and $\rho = 0.3$, average execution times (in seconds) for different methods with different values of n and p.

Overall, the empirical results strongly indicate that Fiducial Selector is the superior method. It consistently demonstrated the smallest biases and achieved empirical coverage

n	p	Fiducial Selector	AutoGFI	LDPE	SSLasso	pBLPR	MOCE
300	400	6.62	7.98	670.63	104.57	104.94	8.97
	600	9.42	11.47	510.36	434.40	321.51	15.55
	1000	16.03	20.24	873.98	807.42	804.74	48.42
500	400	21.17	24.22	1334.49	237.83	344.27	42.27
	600	28.14	34.03	909.11	541.38	485.37	110.67
	1000	56.40	66.95	1444.84	1075.12	984.83	90.80

Table 4.10. In scenario of $s = 15$ and $\rho = 0.3$, average execution times (in seconds) for different methods with different values of n and p.

rates that closely aligned with the target levels, all while maintaining the narrowest confidence intervals. Moreover, Fiducial Selector ranked as the fastest method among those tested, further solidifying its preference for high-dimensional regression analysis.

4.4.2. Misspecified Models. In this section, we evaluate the performance of Fiducial Selector in situations where the model is misspecified. Specifically, we consider two situations: one where the noise distribution is misspecified and one where the model structure is misspecified.

Misspecified noise distributions. For this scenario, we generated the noise term $\boldsymbol{e}$ from two non-Gaussian distributions: a heavy-tailed Student's t-distribution and a non-symmetric chi-square distribution. Specifically, we assumed the true model structure follows model (4.19). For the Student's t-distribution, we assumed the noise term $\boldsymbol{e}_i \overset{i.i.d.}{\sim} c_0 t_3$, where the scale c_0 is set to 0.3. For the chi-square distribution, we assumed the noise term $\boldsymbol{e}_i \overset{i.i.d.}{\sim} c_0(\chi_4 - 4)$, with the scale c_0 set to 0.4. The design matrix $\boldsymbol{X}$ and the true coefficient vector $\boldsymbol{\beta}$ were generated as described in Section 4.4.1, with $\rho = 0.5$ and all signal parameters equal to 0.5. We evaluated cases with $n = 300$, $p = 1000$, and $s = 5, 15$. For each case, we simulated 500 datasets using the linear model (4.19) with independent error terms. For each dataset, a fiducial sample of size 1000 was generated to estimate the model coefficients and construct confidence intervals. The empirical results are summarized in Table 4.11 and

Table 4.12. It is clear that fiducial methods, Fiducial Selector and AutoGFI, are robust to the misspecified noise distributions.

	s		Oracle	Fiducial Selector	AutoGFI	LDPE	SSLasso	pBLPR	MOCE
t_3	5	$\mathcal{A}$	$0.0237_{0.0010}$	$0.0251_{0.0010}$	$0.0254_{0.0010}$	$0.0302_{0.0017}$	$0.0325_{0.0029}$	$0.0272_{0.0014}$	$0.0384_{0.0022}$
		$\mathcal{A}^c$	–	$0.0000_{0.0000}$	$0.0000_{0.0001}$	$0.0332_{0.0044}$	$0.0235_{0.0015}$	$0.0056_{0.0003}$	$0.0095_{0.0007}$
	15	$\mathcal{A}$	$0.0241_{0.0007}$	$0.0261_{0.0007}$	$0.0275_{0.0016}$	$0.0376_{0.0089}$	$0.0378_{0.0085}$	$0.0279_{0.0012}$	$0.0454_{0.0046}$
		$\mathcal{A}^c$	–	$0.0000_{0.0002}$	$0.0002_{0.0007}$	$0.0336_{0.0069}$	$0.0252_{0.0038}$	$0.0055_{0.0003}$	$0.0096_{0.0013}$
χ_4	5	$\mathcal{A}$	$0.0257_{0.0008}$	$0.0258_{0.0008}$	$0.0258_{0.0008}$	$0.0327_{0.0009}$	$0.0341_{0.0008}$	$0.0297_{0.0004}$	$0.0604_{0.0057}$
		$\mathcal{A}^c$	–	$0.0000_{0.0000}$	$0.0000_{0.0002}$	$0.0357_{0.0038}$	$0.0260_{0.0017}$	$0.0062_{0.0003}$	$0.0087_{0.0012}$
	15	$\mathcal{A}$	$0.0257_{0.0008}$	$0.0258_{0.0008}$	$0.0258_{0.0008}$	$0.0327_{0.0009}$	$0.0427_{0.0097}$	$0.0297_{0.0004}$	$0.0604_{0.0057}$
		$\mathcal{A}^c$	–	$0.0000_{0.0000}$	$0.0000_{0.0002}$	$0.0357_{0.0038}$	$0.0278_{0.0041}$	$0.0062_{0.0003}$	$0.0087_{0.0012}$

TABLE 4.11. In the scenario of misspecified noise distribution, averages of the biases $\text{Bias}_{\mathcal{A}}$ and $\text{Bias}_{\mathcal{A}^c}$ for different methods across 500 simulation rounds, with standard errors as subscripts.

	s	$1-\alpha$	Oracle	Fiducial Selector	AutoGFI	LDPE	SSLasso	pBLPR	MOCE
t_3	5	90%	$0.892_{0.098}$	$0.886_{0.096}$	$0.894_{0.101}$	$0.929_{0.140}$	$0.880_{0.125}$	$0.883_{0.206}$	$0.777_{0.116}$
		95%	$0.943_{0.116}$	$0.938_{0.115}$	$0.946_{0.121}$	$0.973_{0.167}$	$0.928_{0.143}$	$0.934_{0.251}$	$0.869_{0.139}$
		99%	$0.988_{0.153}$	$0.981_{0.149}$	$0.984_{0.158}$	$0.995_{0.220}$	$0.973_{0.179}$	$0.979_{0.348}$	$0.951_{0.182}$
	15	90%	$0.897_{0.100}$	$0.889_{0.099}$	$0.894_{0.107}$	$0.882_{0.149}$	$0.889_{0.151}$	$0.860_{0.229}$	$0.771_{1084.239}$
		95%	$0.947_{0.119}$	$0.940_{0.118}$	$0.945_{0.128}$	$0.938_{0.177}$	$0.934_{0.170}$	$0.922_{0.288}$	$0.852_{1291.950}$
		99%	$0.989_{0.157}$	$0.983_{0.153}$	$0.985_{0.169}$	$0.985_{0.233}$	$0.975_{0.208}$	$0.980_{0.393}$	$0.947_{1697.910}$
χ_4	5	90%	$0.894_{0.108}$	$0.893_{0.107}$	$0.899_{0.109}$	$0.931_{0.150}$	$0.893_{0.138}$	$0.877_{0.254}$	$0.745_{0.179}$
		95%	$0.950_{0.128}$	$0.947_{0.128}$	$0.952_{0.131}$	$0.970_{0.179}$	$0.936_{0.158}$	$0.933_{0.312}$	$0.827_{0.213}$
		99%	$0.987_{0.169}$	$0.988_{0.167}$	$0.989_{0.171}$	$0.996_{0.235}$	$0.978_{0.197}$	$0.978_{0.407}$	$0.922_{0.280}$
	15	90%	$0.895_{0.111}$	$0.892_{0.111}$	$0.902_{0.121}$	$0.879_{0.158}$	$0.879_{0.167}$	$0.847_{0.247}$	$0.627_{0.156}$
		95%	$0.947_{0.132}$	$0.944_{0.132}$	$0.953_{0.147}$	$0.938_{0.189}$	$0.926_{0.188}$	$0.909_{0.298}$	$0.719_{0.185}$
		99%	$0.989_{0.173}$	$0.989_{0.171}$	$0.991_{0.199}$	$0.986_{0.248}$	$0.974_{0.231}$	$0.976_{0.402}$	$0.851_{0.244}$

TABLE 4.12. In scenario of misspecified noise distribution, average empirical coverage probabilities $\text{CP}^{\alpha}_{\mathcal{A}}$, with interval widths $\text{Wid}^{\alpha}_{\mathcal{A}}$ as subscripts, for the signal parameters across 500 simulation rounds. The table includes results for target levels $\alpha = 0.1, 0.05$, and 0.01.

Misspecified linear model. In this scenario, we examine the performance of Fiducial Selector when the linear model structure is misspecified. We simulated the response vector

$\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ from the following model:

$$Y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \sum_{j \in \mathcal{A}} \alpha_j x_{ij}^2 + \sum_{j,k \in \mathcal{A}, j<k} \alpha_{jk} x_{ij} x_{ik} + e_i, \quad e_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma), \tag{4.20}$$

where $\boldsymbol{x}_i$ is the i-th row of the design matrix $\boldsymbol{X}$, and $\mathcal{A}$ is a randomly selected subset of $1, \ldots, p$ with size $s = 3$. The coefficients α_j and α_{jk} were independently generated from a uniform distribution $U(0, 0.1)$. The true coefficients β_j were set to 0.5 for $j \in \mathcal{A}$ and 0 for $j \notin \mathcal{A}$, and the error scale σ was set to 0.5.

Since the quadratic and interaction terms are not included in the design matrix $\boldsymbol{X}$, the linear model (4.1) is misspecified. In this misspecified linear model, the parameter vector $\boldsymbol{\beta}_0$ we are interested in is the projection coefficient of $E(\boldsymbol{Y}|\boldsymbol{X})$ onto the subspace spanned by the relevant predictors:

$$\boldsymbol{\beta}^0_{\mathcal{A}} = (\boldsymbol{X}_{\mathcal{A}}^\top \boldsymbol{X}_{\mathcal{A}})^{-1} \boldsymbol{X}_{\mathcal{A}}^\top E(\boldsymbol{Y}|\boldsymbol{X}); \quad \boldsymbol{\beta}^0_{\mathcal{A}^c} = \mathbf{0}.$$

Again, we considered the case when $n = 300$ and $p = 1000$. The design matrix $\boldsymbol{X}$ was generated as described in Section 4.4.1, with the correlation parameter $\rho = 0.5$. We simulated 500 datasets from model (4.20). For each dataset, a fiducial sample of size 1000 was generated to estimate the projection coefficient $\boldsymbol{\beta}_0$ and construct confidence intervals. The empirical results are summarized in Table 4.13 and Table 4.14, demonstrating that Fiducial Selector remains robust even when the linear model is misspecified.

	Oracle	Fiducial Selector	AutoGFI	LDPE	SSLasso	pBLPR	MOCE
$\mathcal{A}$	$0.0218_{0.0008}$	$0.0252_{0.0053}$	$0.0303_{0.0059}$	$0.0296_{0.0012}$	$0.0388_{0.0060}$	$0.0328_{0.0069}$	$0.0508_{0.0156}$
$\mathcal{A}^c$	–	$0.0000_{0.0008}$	$0.0004_{0.0013}$	$0.0336_{0.0057}$	$0.0276_{0.0063}$	$0.0065_{0.0015}$	$0.0117_{0.0032}$

Table 4.13. In the scenario of misspecified linear model, averages of the biases $\text{Bias}_{\mathcal{A}}$ and $\text{Bias}_{\mathcal{A}^c}$ for different methods across 500 simulation rounds, with standard errors as subscripts.

4.4.3. Real Data Example. In this section, we apply Fiducial Selector to analyze a real dataset about riboflavin (vitamin B_2) production rate. This riboflavin production

$1-\alpha$	Oracle	Fiducial Selector	AutoGFI	LDPE	SSLasso	pBLPR	MOCE
90%	$0.916_{0.095}$	$0.907_{0.107}$	$0.923_{0.127}$	$0.954_{0.143}$	$0.865_{0.142}$	$0.929_{0.297}$	$0.751_{0.130}$
95%	$0.957_{0.113}$	$0.946_{0.127}$	$0.961_{0.152}$	$0.981_{0.171}$	$0.923_{0.164}$	$0.964_{0.350}$	$0.826_{0.157}$
99%	$0.994_{0.149}$	$0.985_{0.165}$	$0.990_{0.197}$	$0.999_{0.224}$	$0.979_{0.207}$	$0.994_{0.418}$	$0.939_{0.201}$

TABLE 4.14. In scenario of misspecified linear model, average empirical coverage probabilities $\mathrm{CP}_{\mathcal{A}}^{\alpha}$, with interval widths $\mathrm{Wid}_{\mathcal{A}}^{\alpha}$ as subscripts, for the signal parameters across 500 simulation rounds. The table includes results for target levels $\alpha = 0.1, 0.05$, and 0.01.

data set was first made public by Bühlmann et al. (2014), which contains $n = 71$ samples and $p = 4088$ covariates. The response variable is the logarithm of riboflavin production rate, while the 4088 covariates are the logarithm of the expression level of $p = 4088$ genes. Many studies have been done on this dataset to detect the genes that are related to the riboflavin production rate. Under the 5% significance level, Meinshausen et al. (2009) used the multisample-splitting method to find only 'YXLD-at' as a significant variable. At the same time, Javanmard and Montanari (2014) applied the SSLasso method and located 'YXLD-at' and 'YXLE-at' as significant variables. Bühlmann et al. (2014) identified 'LYSC-at', 'YOAB-at', and 'YXLD-at' as significant ones by the method stability selection proposed by Meinshausen and Bühlmann (2010). Chichignoud et al. (2016) calibrated the lasso in the supremum norm ℓ_∞-loss to find 'YXLD-at', 'YOAB-at', 'YEBC-at', 'ARGF-at' and 'XHLB-at' as significant genes. However, van de Geer et al. (2014) applied their method that is fundamentally the same as LDPE (RLDPE) and found no variable significant at a 5% significance level.

We applied Fiducial Selector directly to the original dataset without variable screening. Under 5% significance level, it detected 'YXLD-at', 'YOAB-at', 'YEBC-at', 'ARGF-at', 'YCKE-at', 'YDDK-at' and 'XTRA-at' as significant variables. In other words, Fiducial Selector was able to locate 'YXLD-at' and 'YOAB-at', which are considered significant in most previous studies. Moreover, it is computationally efficient and takes less than 1 minute to run on a personal laptop.

To allow for evaluation of the prediction accuracy of the proposed fiducial selector, the sample of size 71 was first randomly split into a training set of size 61 and a holdout test set of size 10. The entire process was then repeated 10 times to reliably compare the resulting prediction errors (SSPE, sum of squared prediction error) and model sizes. Comparison methods include AutoGFI, SSLasso, and pBLPR. We summarize the results in Table 4.15. One can see that Fiducial Selector achieves a comparable prediction error with pBLPR, which has the lowest prediction error but uses all the predictors due to its ridge penalty.

method:	Fiducial Selector	AutoGFI	SSLasso	pBLPR
SSPE	2.488	3.471	3.720	2.284
model size	12.8	11.7	64.1	4088

Table 4.15. SSPE and model size from different methods for the riboflavin dataset.

4.5. Conclusion

In this chapter, we have adapted the GFI framework to address the inference challenges in the high-dimensional regression problem through the development of Fiducial Selector. The Fiducial Selector is notable for its computational efficiency and incorporates an innovative de-biasing technique that greatly enhances its performance. Through rigorous theoretical analysis, we have demonstrated that Fiducial Selector consistently selects the correct model and offers unbiased estimates of the significant parameters under certain conditions. Empirical tests reinforce these theoretical assertions, confirming that Fiducial Selector delivers reliable uncertainty quantification for high-dimensional regression challenges. Comparative evaluations with other contemporary methods suggest that Fiducial Selector offers notable computational and statistical advantages, making it a promising choice for practitioners and researchers navigating the complexities of high-dimensional data analysis.

CHAPTER 5

Streamlined GFI for Binary Response Models

In this chapter, we extend the AutoGFI framework from continuous response models to binary response models. Specifically, we consider the model with a binary response vector $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^\top$, where each response Y_i is independently and identically distributed as Bernoulli with probability p_i, denoted by

$$Y_i \overset{i.i.d.}{\sim} \text{Bernoulli}(p_i). \tag{5.1}$$

The probabilities p_i are linked to a set of multivariate predictors $\boldsymbol{x}_i$ and a parameter vector $\boldsymbol{\theta}$ through the log-odds, utilizing a flexible function f. This relationship is formalized as:

$$\log\left(\frac{p_i}{1-p_i}\right) = f(\boldsymbol{x}_i, \boldsymbol{\theta}). \tag{5.2}$$

The function $f(\boldsymbol{x}_i, \boldsymbol{\theta})$ is assumed to capture complex interactions between predictors $\boldsymbol{x}_i$ and parameters $\boldsymbol{\theta}$, accommodating both linear and non-linear dependencies. This allows the model to flexibly adapt to varying data structures and relationships inherent in real-world scenarios.

This chapter introduces a binary version of AutoGFI to simplify the implementation of GFI for the aforementioned models. Specifically, we present the approach AutoGFI-B in Section 5.1 and its regularized and debiased counterpart, AutoGFI-BR, in Section 5.2. We then demonstrate the applicability and empirical performance of this approach by applying it to three particularly useful binary response models: the logistic regression model in Section 5.3, the covariate-assisted ranking estimation model (an extension of the Bradley-Terry-Luce model for pairwise comparison problems) in Section 5.4, and the Rasch model for item response theory applications in Section 5.5. Concluding remarks are provided in Section 5.6.

5.1. AutoGFI-B for Binary Response Models

The data generating equation for the model (5.1) has the following form:

$$Y_i = \mathbb{1}\{f(\boldsymbol{x}_i, \boldsymbol{\theta}) + L_i \geq 0\}, \quad i = 1, \dots, n, \tag{5.3}$$

where $\mathbb{1}$ is the indicator function such that $\mathbb{1}(x \geq 0) = 1$ if $x \geq 0$ and 0 otherwise, and $L_i = \log \frac{U_i}{1-U_i}$ with $U_i \overset{i.i.d.}{\sim}$ Uniform$(0, 1)$. In other words, $L_i \overset{i.i.d.}{\sim}$ Logistic$(0, 1)$. Let $\boldsymbol{L} = (L_1, \dots, L_n)^\top$, $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) = (f(\boldsymbol{x}_1, \boldsymbol{\theta}), \dots, f(\boldsymbol{x}_n, \boldsymbol{\theta}))^\top$, and $\boldsymbol{y} = (y_1, \dots, y_n)^\top$ be the observed data of $\boldsymbol{Y}$. According to the Definition (2.1.1) of GFD, in practice one could generate a piece of the approximate fiducial sample with a pre-defined semi-metric ρ and some $\epsilon > 0$ through the following steps:

(1) Generate $\boldsymbol{l}^*$ from $\boldsymbol{L}^*$, an independent copy of $\boldsymbol{L}$.

(2) Solve

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \rho(\mathbb{1}\{\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) + \boldsymbol{l}^* \geq \boldsymbol{0}\}, \boldsymbol{y}). \tag{5.4}$$

(3) Accept $\boldsymbol{\theta}^*$ if $\rho(\mathbb{1}\{\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}^*) + \boldsymbol{l}^* \geq \boldsymbol{0}\}, \boldsymbol{y}) \leq \epsilon$; otherwise reject and return to Step 1.

Here $\boldsymbol{\theta}^*$ represents a realization of the fiducial sample.

However, due to the discontinuity of the data generating equation (5.3), locating the optimizer of (5.4) presents significant challenges. To circumvent this issue, we propose using a sigmoid function with a relatively large scale parameter as an approximation for the indicator function. The sigmoid function is defined as follows:

$$\text{sigmoid}(x, a) = \frac{1}{1 + e^{-ax}}, \tag{5.5}$$

where a represents the scale parameter, controlling the steepness of the function. As illustrated in Figure 5.1, when a exceeds 5, the sigmoid function closely resembles the indicator function. With such approximation, the data generating equation (5.3) becomes

$$Y_i \approx \text{sigmoid}(f(\boldsymbol{x}_i, \boldsymbol{\theta}) + L_i, a), \quad i = 1, \ldots, n. \tag{5.6}$$

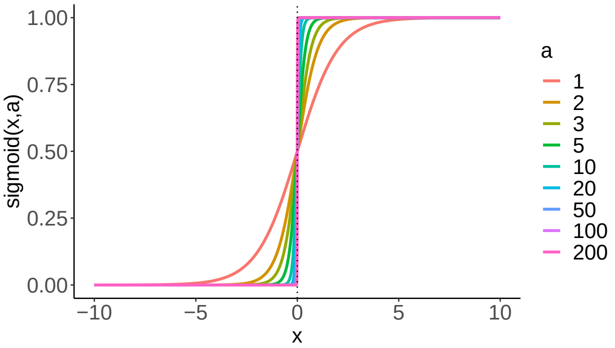


FIGURE 5.1. Sigmoid function with different values of a.

Plugging the approximate data generating equation (5.6) into the optimization problem (5.4) yields an objective function of the form $\rho(\text{sigmoid}(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) + \boldsymbol{l}^*, a), \boldsymbol{y})$. However, for commonly used semi-metrics ρ, such as the squared ℓ_2 norm, this objective function remains non-convex even for linear $\boldsymbol{f}$, still rendering the optimization problem challenging to solve. To address this issue, we propose further relaxing the semi-metric ρ by replacing it with the cross-entropy loss, denoted as $\tilde{\rho}$, which possesses more favorable optimization properties. Therefore, Step (2) becomes

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \tilde{\rho}(\boldsymbol{y}^*(\boldsymbol{\theta}), \boldsymbol{y}) = \arg\min_{\boldsymbol{\theta}} -\sum_{i=1}^{n} \left[y_i \log y_i^*(\boldsymbol{\theta}) + (1 - y_i) \log(1 - y_i^*(\boldsymbol{\theta})) \right], \tag{5.7}$$

where

$$\boldsymbol{y}^*(\boldsymbol{\theta}) = \text{sigmoid}(f(\boldsymbol{x}, \boldsymbol{\theta}) + \boldsymbol{l}^*, a) \tag{5.8}$$

with $y_i^*(\boldsymbol{\theta})$ being the i-th element of it.

Having relaxed the indicator function and introduced the cross-entropy loss, we can now generate the approximate fiducial sample for the model (5.1) using Algorithm (3), which we term AutoGFI-B, where “B” stands for binary data. The resulting fiducial sample forms a distribution estimate of $\boldsymbol{\theta}$, which can be used in a manner similar to a Bayesian posterior sample. We can use this fiducial sample to derive point estimates and confidence intervals for $\boldsymbol{\theta}$. For instance, the average of all copies in the fiducial sample can serve as the point estimator of $\boldsymbol{\theta}$, while the $\alpha/2$ percentile and $(1-\alpha/2)$ percentile can be used to construct the $(1-\alpha)$ confidence interval.

Algorithm 3 AutoGFI-B: Generating fiducial sample for model (5.1) without regularization.

```
Input: Data $\boldsymbol{x}$, $\boldsymbol{y}$; fiducial sample size $N$; sigmoid scale parameter $a$; tolerance $\epsilon$
Output: Approximate fiducial sample for $\boldsymbol{\theta}$ of size $N$
for $k = 1$ to $N$ do
    Generate $\boldsymbol{l}^*$ from $\boldsymbol{L}^*$.
    Solve $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \tilde{\rho}(\boldsymbol{y}^*(\boldsymbol{\theta}), \boldsymbol{y})$ where $\boldsymbol{y}^*(\boldsymbol{\theta})$ is defined in (5.8).
    if $\tilde{\rho}(\boldsymbol{y}^*(\boldsymbol{\theta}^*), \boldsymbol{y}) \leq \epsilon$ then
        Accept $\boldsymbol{\theta}^*$ as the $k-$th piece of the fiducial sample.
    else
        Reject $\boldsymbol{\theta}^*$ and repeat from line 4.
```

It is worth emphasizing that AutoGFI-B exhibits remarkable versatility, as it is compatible with a wide range of functions $\boldsymbol{f}$. For any given form of $\boldsymbol{f}$, AutoGFI-B can be effectively applied to conduct inference on the corresponding class of problems, provided that the optimization problem (5.7) can be solved. This inherent flexibility makes AutoGFI-B suitable for a diverse set of inference tasks, as demonstrated in Sections 5.3, 5.4, and 5.5.

5.2. AutoGFI-BR: The Regularized AutoGFI-B

Although the algorithm AutoGFI-B is flexible with the choice of $\boldsymbol{f}$, it is limited to models that do not require regularization. Taking inspiration from the approach AutoGFI introduced in Chapter 3, we propose the following generalizations to AutoGFI-B to accommodate regularized models: First, a penalty term $\Lambda(\boldsymbol{\theta})$ can be added to the optimization problem (5.7) to introduce regularization or shrinkage. Next, when penalty is applied, a de-biasing operation $\boldsymbol{d}(\boldsymbol{\theta})$ is subsequently performed to mitigate the bias introduced by the penalty.

The optimization problem with regularization has the following form:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \tilde{\rho}(\boldsymbol{y}^*(\boldsymbol{\theta}), \boldsymbol{y}) + \Lambda(\boldsymbol{\theta}), \tag{5.9}$$

where $\tilde{\rho}$ and $\boldsymbol{y}^*(\boldsymbol{\theta})$ are defined in (5.7) and (5.8), respectively. Due to the penalty term $\Lambda(\boldsymbol{\theta})$, $\boldsymbol{\theta}^*$ becomes a biased fiducial sample, necessitating a de-biasing step. The de-biasing idea is as follows: Under certain differentiability conditions, solving the problem (5.9) is often equivalent to solving the estimating equation

$$\nabla_{\boldsymbol{\theta}}\tilde{\rho}(\boldsymbol{y}^*(\boldsymbol{\theta}),\boldsymbol{y})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}+\boldsymbol{\xi}(\boldsymbol{\theta}^*)=0, \tag{5.10}$$

where $\boldsymbol{\xi}(\boldsymbol{\theta}^*)$ is a (sub-)gradient of the penalty term $\Lambda(\boldsymbol{\theta})$ evaluated at $\boldsymbol{\theta}^*$. To remove the bias associated with the penalty, we aim to modify $\boldsymbol{\theta}^*$ such that the first term of (5.10) is closer to zero. To this end, we employ a one-step modification and define the de-biased fiducial sample $\boldsymbol{\theta}^*_{\text{de}}$ by solving the equation:

$$\nabla_{\boldsymbol{\theta}}\tilde{\rho}(\boldsymbol{y}^*(\boldsymbol{\theta}),\boldsymbol{y})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}+\mathbf{H}(\boldsymbol{\theta}^*)(\boldsymbol{\theta}^*_{\text{de}}-\boldsymbol{\theta}^*)=\mathbf{0}, \tag{5.11}$$

where $\mathbf{H}(\boldsymbol{\theta}^*)$ is the Hessian matrix of second partial derivatives of $\tilde{\rho}(\boldsymbol{y}^*(\boldsymbol{\theta}),\boldsymbol{y})$ with respect to $\boldsymbol{\theta}$, evaluated at $\boldsymbol{\theta}^*$. More specifically, under our settings,

$$\nabla_{\boldsymbol{\theta}}\tilde{\rho}(\boldsymbol{y}^*(\boldsymbol{\theta}),\boldsymbol{y})=a\sum_{i=1}^{n}(y_i^*(\boldsymbol{\theta})-y_i)\nabla_{\boldsymbol{\theta}}f(\boldsymbol{x}_i,\boldsymbol{\theta}), \tag{5.12}$$

$$\mathbf{H}(\boldsymbol{\theta})=a^2\sum_{i=1}^{n}\left[y_i^*(\boldsymbol{\theta})(1-y_i^*(\boldsymbol{\theta}))\nabla_{\boldsymbol{\theta}}f(\boldsymbol{x}_i,\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}f(\boldsymbol{x}_i,\boldsymbol{\theta})^\top+(y_i^*(\boldsymbol{\theta})-y_i)\mathbf{H}_f(\boldsymbol{x}_i,\boldsymbol{\theta})\right], \tag{5.13}$$

where $\mathbf{H}_f(\boldsymbol{x}_i,\boldsymbol{\theta})$ is the Hessian matrix of second partial derivatives of $f(\boldsymbol{x}_i,\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

In high-dimensional settings, the matrix $\mathbf{H}(\boldsymbol{\theta}^*)$ is often rank-deficient and poorly conditioned, leading to high variability in the solution to (5.11). To manage this variability, we employ a two-step approach. First, when sparsity is introduced with a penalty, we treat formula (5.9) as a model selection step and only perform de-biasing for the "significant" parameters. These significant parameters are identified by computing the percentage of non-zero values in the fiducial sample for each component θ_j of the parameter vector $\boldsymbol{\theta}$. If this percentage exceeds 50%, we consider θ_j significant; otherwise, it is treated as a constant 0. Consequently, $\nabla_{\boldsymbol{\theta}}\tilde{\rho}(\boldsymbol{y}^*(\boldsymbol{\theta}), \boldsymbol{y})$ and $\mathbf{H}(\boldsymbol{\theta}^*)$ are only calculated along the coordinates corresponding to these significant components θ_j, reducing the computational burden and improving the stability of the de-biasing process.

Secondly, we mitigate the variability by employing a pseudo-inverse $\mathbf{H}^{\text{pinv}}$, which disregards small singular values during matrix inversion. Specifically, for a square matrix $\mathbf{H} \in \mathbb{R}^{n\times n}$ with Singular Value Decomposition (SVD) given by $\mathbf{H} = \mathbf{V\Sigma W}^\top$, where $\mathbf{\Sigma}$ is the diagonal matrix containing all singular values, we construct $\mathbf{S}$ as the diagonal matrix containing only singular values greater than a threshold c. The pseudo-inverse of $\mathbf{H}$ is then defined as

$$\mathbf{H}^{\text{pinv}} := \mathbf{W}\begin{pmatrix} \mathbf{S}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}\mathbf{V}^\top.$$

Essentially, we only use singular vectors corresponding to significant singular values of $\mathbf{H}$ for de-biasing. The threshold c can be determined in a data-dependent manner.

With these considerations, the de-biasing function is defined as:

$$\boldsymbol{\theta}^*_{s,\text{de}} := \boldsymbol{\theta}^*_s - \mathbf{H}(\boldsymbol{\theta}^*_s)^{\text{pinv}}\nabla_{\boldsymbol{\theta}_s}\tilde{\rho}(\boldsymbol{y}^*(\boldsymbol{\theta}_s), \boldsymbol{y})\big|_{\boldsymbol{\theta}_s=\boldsymbol{\theta}^*_s}, \tag{5.14}$$

where $\boldsymbol{\theta}_s$ is a vector of the significant components θ_j identified from the biased fiducial sample.

In summary, when regularization is added, the de-biased fiducial sample can be generated using Algorithm (4). We term this algorithm AutoGFI-BR, where “R” stands for regularization.

5.3. Logistic Regression

5.3.1. Introduction. When specializing the function $f(\boldsymbol{x}_i, \boldsymbol{\theta})$ as

$$f(\boldsymbol{x}_i, \boldsymbol{\theta}) = \boldsymbol{x}_i^\top\boldsymbol{\theta},$$

Algorithm 4 AutoGFI-BR: Generating fiducial sample for model (5.1) with regularization.

```
1: Input: Data $\boldsymbol{x}$, $\boldsymbol{y}$; fiducial sample size $N$; sigmoid scale parameter $a$; tolerance $\epsilon$; pseudo-inverse threshold $c$
2: Output: Approximate fiducial samples $\boldsymbol{\theta}^*$ for the parameter $\boldsymbol{\theta}$ of size $N$
3: for $k = 1$ to $N$ do
4:     Generate $\boldsymbol{l}^*$ from $\boldsymbol{L}^*$.
5:     Solve $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \tilde{\rho}(\boldsymbol{y}^*(\boldsymbol{\theta}), \boldsymbol{y}) + \Lambda(\boldsymbol{\theta})$ where $\boldsymbol{y}^*(\boldsymbol{\theta})$ is defined in (5.8).
6:     if $\tilde{\rho}(\boldsymbol{y}^*(\boldsymbol{\theta}^*), \boldsymbol{y}) \leq \epsilon$ then
7:         Accept $\boldsymbol{\theta}^*$ as the $k-$th piece of the fiducial sample.
8:     else
9:         Reject $\boldsymbol{\theta}^*$ and repeat from line 4.
10: Identify $\boldsymbol{\theta}_s$ and debias for $\boldsymbol{\theta}_s^*$ by equation (5.14).
```

model (5.2) becomes the logistic regression model, which is one of the most widely used models for classification problems. Its significance spans across diverse fields, from engineering and applied sciences to social sciences and beyond. An detailed introduction about the application of the logistic regression model can be found in Hosmer Jr et al. (2013).

Logistic regression in low-dimensional cases, where $n/p \gg 1$, has been thoroughly investigated. It is established that with fixed p and n approaching infinity, the maximum likelihood estimator (MLE) is efficient, with its covariance matrix converging to the inverse of the Fisher information matrix. Recent studies (Sur and Candès, 2018; Sur et al., 2019; Candès and Sur, 2020) have extended this analysis to situations where n/p approaches a fixed ratio, demonstrating that while MLE remains unbiased, its variability is significantly greater than previously estimated. For cases where $p/n \gg 1$, regularized logistic regression, particularly with ℓ_1 norm penalties assuming sparsity in parameters, has been proposed. Extensive research has addressed the properties of such penalized estimators, focusing on solving the ℓ_1-penalized optimization problem at scale (Lee et al., 2006; Koh et al., 2007), and analyzing estimation and variable selection characteristics (Bunea, 2008; van de Geer, 2008; Salehi et al., 2019), as well as statistical inference (van de Geer et al., 2014; Javanmard and Montanari, 2013; Dezeure et al., 2015; Ning and Liu, 2017; Ma et al., 2020; T. Tony Cai and Ma, 2023; Guo et al., 2021). Notably, van de Geer et al. (2014) achieve asymptotic normality for debiased estimators by inverting the Karush-Kuhn-Tucker (KKT) conditions of the ℓ_1 regularized loss function. Ning and Liu (2017) construct an approximately unbiased estimator through the decorrelated score function, offering a general framework for hypothesis testing and confidence region development in high-dimensional models. Ma et al. (2020) address the bias of the logistic Lasso estimator using a generalized low-dimensional projection method and derive the null distribution for conducting global tests. T. Tony Cai and Ma (2023) explore binary generalized linear models with various link functions, proposing a link-specific weighting method for bias correction in penalized estimators. Guo et al. (2021) propose a bias-corrected estimator for the case probability using linearization and variance enhancement techniques.

Given the widespread popularity and extensive application of logistic regression in practical scenarios, we selected it as our initial example to demonstrate the capability of our proposed method. By comparing our approach with both classical techniques in low-dimensional settings and advanced debiasing methods in high-dimensional contexts, we aim to illustrate the robustness of our method with respect to the hyperparameter a and the effectiveness of the debiasing step (5.14) in high-dimensional cases.

5.3.2. AutoGFI-B(R) for Logistic Regression.

In the absence of regularization, the approximate fiducial sample of $\boldsymbol{\theta}$ can be generated by applying AutoGFI-B with $f(\boldsymbol{x}_i, \boldsymbol{\theta}) = \boldsymbol{x}_i^\top \boldsymbol{\theta}$.

However, when the dimensionality p exceeds the sample size n or when $\boldsymbol{\theta}$ is assumed to be sparse, we introduce ℓ_1-regularization to the model. Under this regularization scheme, the approximate fiducial sample can be obtained using AutoGFI-BR, the Algorithm (4), with $f(\boldsymbol{x}_i, \boldsymbol{\theta}) = \boldsymbol{x}_i^\top \boldsymbol{\theta}$ and the penalty term $\Lambda(\boldsymbol{\theta}) = \lambda \|\boldsymbol{\theta}\|_1$. For the logistic regression model, the gradient and Hessian matrix of $\tilde{\rho}$ can be efficiently computed as follows:

$$\nabla_{\boldsymbol{\theta}} \tilde{\rho}(\boldsymbol{y}^*(\boldsymbol{\theta}), \boldsymbol{y}) = a\boldsymbol{X}^\top \left(\boldsymbol{y}^*(\boldsymbol{\theta}) - \boldsymbol{y}\right)$$

$$\mathbf{H}(\boldsymbol{\theta}) = a^2 \boldsymbol{X}^\top \boldsymbol{D}(\boldsymbol{\theta}) \boldsymbol{X},$$

where $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ and $\boldsymbol{D}(\boldsymbol{\theta}) = \mathrm{Diag}\big(y_1^*(\boldsymbol{\theta})(1 - y_1^*(\boldsymbol{\theta})), \ldots, y_n^*(\boldsymbol{\theta})(1 - y_n^*(\boldsymbol{\theta}))\big)$. Therefore, the de-biasing step in AutoGFI-BR is

$$\boldsymbol{\theta}^*_{\mathrm{s,de}} = \boldsymbol{\theta}^*_s - \big(\boldsymbol{X}_s^\top \boldsymbol{D}(\boldsymbol{\theta}^*) \boldsymbol{X}_s\big)^{\mathrm{pinv}} \boldsymbol{X}_{\boldsymbol{s}}^\top \big(\boldsymbol{y}^*(\boldsymbol{\theta}^*) - \boldsymbol{y}\big)/a, \tag{5.15}$$

where $\boldsymbol{X}_s$ is the columns of $\boldsymbol{X}$ corresponding to the significant coordinates.

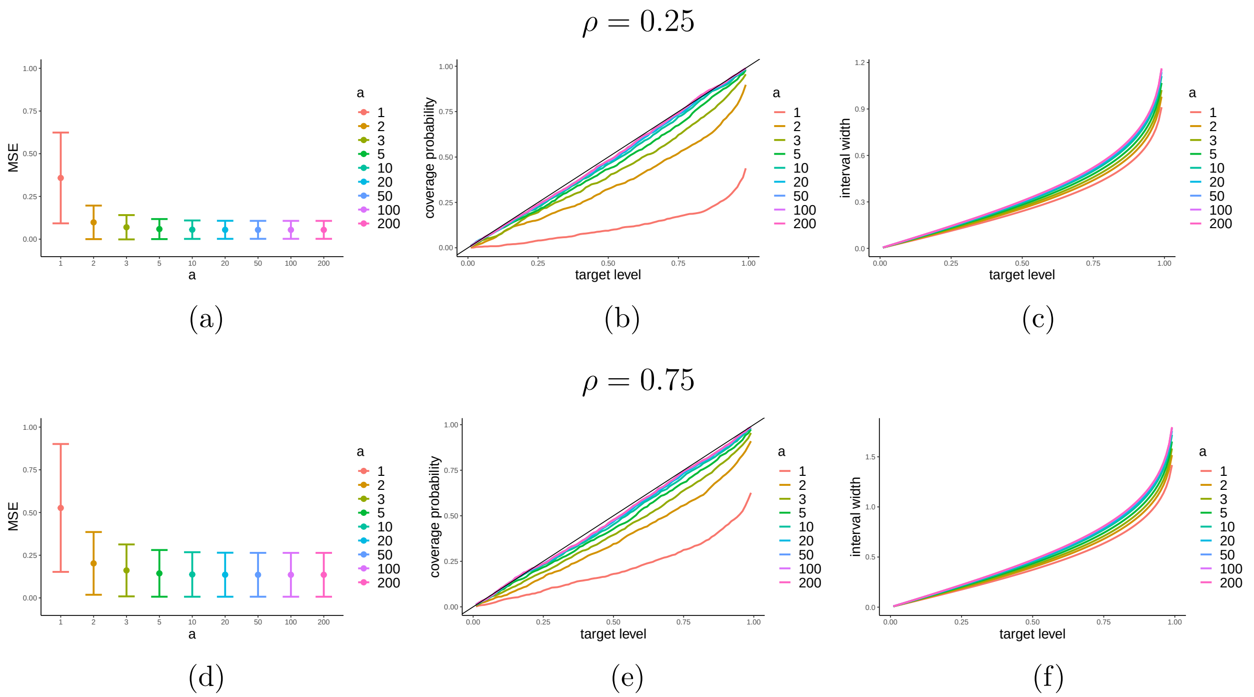
The point estimate and confidence intervals for $\boldsymbol{\theta}$ can then be constructed from the mean and the corresponding percentiles of the approximate fiducial sample, respectively.

5.3.3. Empirical Performance.

This subsection examines the practical performance of the proposed methods for logistic regression in two distinct scenarios. First, we assess the performance of AutoGFI-B in low-dimensional settings where regularization is not required. Second, we evaluate the performance of AutoGFI-BR in high-dimensional settings with sparse $\boldsymbol{\theta}$, necessitating the use of regularization techniques.

Case 1: Low dimensional settings. In this part, we conducted simulation studies to evaluate AutoGFI-B for logistic regression within low-dimensional contexts. We configured the simulations with $n = 200$, $p = 3$, and set $\theta_j = 1$ for $j = 1, \ldots, p$. The covariates were independently generated from a multivariate Gaussian distribution $N_p(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a Toeplitz matrix with $\Sigma_{ij} = \rho^{|i-j|}$. We considered two values of ρ: 0.25 and 0.75. A total of 500 datasets were generated according to the model outlined in (5.1) and (5.2), with $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$ specified as $\boldsymbol{X}\boldsymbol{\theta}$. For each dataset, we produced a fiducial sample of size 1000 using Algorithm (3) to estimate the model coefficients $\boldsymbol{\theta}$ and to construct confidence intervals.

Our analysis proceeded in two stages. Initially, we evaluated the performance of AutoGFI-B across various a values by examining the mean squared error (MSE) and the empirical coverage probabilities and widths of confidence intervals at different levels, ranging from 0.01 to 0.99. These results are illustrated in Figure 5.2. Subsequently, with a set to 100, we compared our method's point estimate, which is the mean of the fiducial sample, against the maximum likelihood estimate (MLE). We also evaluated our confidence intervals against two predominant alternatives: the profile likelihood-based intervals, denoted by "Profile", and the Wald confidence intervals, denoted by “Wald”. The results of this comparison are detailed in Table 5.1.



(a) (b) (c)

(d) (e) (f)

FIGURE 5.2. Performance of AutoGFI-B for logistic regression across varying values of a and correlation levels ρ. (a) and (d): The average MSE of the estimate of $\boldsymbol{\theta}$. (b) and (e): Target confidence levels versus average empirical coverage probabilities of confidence intervals of $\boldsymbol{\theta}$. (c) and (f): Target confidence levels versus the average interval widths of confidence intervals of $\boldsymbol{\theta}$. The top row corresponds to $\rho = 0.25$, and the bottom row corresponds to $\rho = 0.75$.

Figure 5.2 illustrates the robustness of our proposed method with respect to variations in the scale parameter a. We observe that for values of a greater than 5, the outputs of the method remain essentially unchanged. This observation aligns with the findings from Figure 5.1, which reveals that the sigmoid function effectively approximates the indicator function when a is larger than 5. In summary, our method is highly flexible with the choice of a. Consequently, we selected $a = 100$ for the subsequent comparison of our method with both the profile likelihood-based and Wald methods. As detailed in Table 5.1, our proposed method demonstrates competitive empirical coverage probability when compared to the other two methods, while also offering the advantage of narrower confidence interval widths. These results underscore the strength and reliability of our approach in the context of logistic regression in low-dimensional settings.

	Method	MSE (SE)	Coverage (Width)		
			0.9	0.95	0.99
$\rho = 0.25$	AutoGFI-B	$0.054_{(0.053)}$	$0.896_{(0.746)}$	$0.949_{(0.888)}$	$0.989_{(1.16)}$
	Profile	$0.054_{(0.053)}$	$0.899_{(0.754)}$	$0.953_{(0.899)}$	$0.990_{(1.19)}$
	Wald	$0.054_{(0.053)}$	$0.905_{(0.751)}$	$0.955_{(0.895)}$	$0.991_{(1.18)}$
$\rho = 0.75$	AutoGFI-B	$0.135_{(0.128)}$	$0.885_{(1.15)}$	$0.942_{(1.37)}$	$0.985_{(1.79)}$
	Profile	$0.136_{(0.129)}$	$0.883_{(1.16)}$	$0.943_{(1.39)}$	$0.989_{(1.83)}$
	Wald	$0.136_{(0.129)}$	$0.891_{(1.16)}$	$0.948_{(1.38)}$	$0.991_{(1.81)}$

Table 5.1. Average MSEs, empirical coverage probabilities, and confidence interval widths for the coefficients estimated using different methods, calculated over 500 simulated datasets in low-dimensional settings.

Case 2: High-dimensional settings. In this section, we conducted simulation studies to assess the performance of AutoGFI-BR in high-dimensional settings that require regularization in the model. Specifically, we set $n = 200$ while varying p across 100, 200, and 500. Assuming the true signal is sparse, we assigned $\theta_j = 1$ for $j = 1, 2, 3$, with the remaining coefficients set to 0. The covariates were independently generated from a multivariate Gaussian distribution $N_p(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ is a Toeplitz matrix with $\Sigma_{ij} = \rho^{|i-j|}$. In this case, we considered $\rho = 0.25$. We generated 200 datasets according to the model (5.1). For each dataset, a fiducial sample of size 1000 was produced via Algorithm (4) to estimate the model coefficients $\boldsymbol{\theta}$ and to construct confidence intervals. The scale parameter a was set to 5. This small value was chosen because a larger value of a steepens the sigmoid function, complicating the optimization process and making it harder to find the minimizer due to increased gradient sensitivity. When employing the ℓ_1 penalty, this sensitivity can easily cause divergent behavior. To avoid such issues, we set the scale parameter as small as possible. The penalty parameter associated with the ℓ_1 norm was chosen by cross-validation. The pseudo-inverse threshold c was set to 0.001.

We compared the proposed method with several popular existing methods, including the Lasso low-dimensional projection method ("Lasso-Proj") proposed by van de Geer et al. (2014) and implemented by the function `lasso.proj` in the R package `hdi`; the Ridge projection method ("Ridge-Proj") proposed by Bühlmann (2013) and implemented by the function `ridge.proj` in the R package `hdi`; and the link-specific weighting method ("LSW") proposed by T. Tony Cai and Ma (2023) and implemented by the function `LR` in the R package `SIHR`. The summary statistics of the numerical results are presented in Table 5.2.

	Method	MSE (SE)	Coverage (Width)		
			0.9	0.95	0.99
$p = 100$	AutoGFI-BR	$0.047_{(0.061)}$	$0.912_{(0.758)}$	$0.963_{(0.910)}$	$0.993_{(1.205)}$
	Lasso-proj	$0.052_{(0.089)}$	$0.887_{(0.683)}$	$0.927_{(0.814)}$	$0.983_{(1.069)}$
	ridge-proj	$0.070_{(0.107)}$	$0.867_{(0.794)}$	$0.923_{(0.946)}$	$0.980_{(1.243)}$
	LSW	$0.064_{(0.079)}$	$0.928_{(1.233)}$	$0.957_{(1.469)}$	$0.988_{(1.931)}$
$p = 200$	AutoGFI-BR	$0.049_{(0.072)}$	$0.922_{(0.823)}$	$0.968_{(0.989)}$	$0.995_{(1.317)}$
	Lasso-Proj	$0.048_{(0.070)}$	$0.872_{(0.661)}$	$0.930_{(0.788)}$	$0.977_{(1.036)}$
	Ridge-Proj	$0.094_{(0.125)}$	$0.872_{(0.937)}$	$0.922_{(1.116)}$	$0.985_{(1.466)}$
	LSW	$0.234_{(0.295)}$	$0.888_{(1.412)}$	$0.940_{(1.683)}$	$0.983_{(2.212)}$
$p = 500$	AutoGFI-BR	$0.079_{(0.118)}$	$0.877_{(0.851)}$	$0.952_{(1.024)}$	$0.987_{(1.367)}$
	Lasso-Proj	$0.058_{(0.073)}$	$0.803_{(0.662)}$	$0.888_{(0.788)}$	$0.980_{(1.036)}$
	Ridge-Proj	$0.062_{(0.082)}$	$0.918_{(0.910)}$	$0.965_{(1.067)}$	$0.993_{(1.374)}$
	LSW	$0.150_{(0.529)}$	$0.898_{(1.313)}$	$0.927_{(1.565)}$	$0.970_{(2.056)}$

TABLE 5.2. Average MSEs, empirical coverage probabilities, and confidence interval widths for the coefficients estimated using different methods, calculated over 500 simulated datasets in high-dimensional settings.

When comparing the debiased estimators, AutoGFI-BR and Lasso-Proj achieve the lowest average MSE for the parameter vector $\boldsymbol{\theta}$ at dimensionalities $p = 100$ and 200. However, the performance of AutoGFI-BR slightly diminishes as the dimensionality increases to $p = 500$. In terms of confidence interval analysis, Lasso-Proj tends to generate overly optimistic results, characterized by intervals that are too narrow. This leads to consistent under-coverage, particularly at higher dimensionalities. In contrast, the performance of Ridge-Proj improves with increasing p, although it also exhibits lower coverage probabilities at lower dimensionalities. Both LSW and AutoGFI-BR attain desired coverage probabilities across most tested scenarios. Notably, AutoGFI-BR maintains much narrower intervals compared to LSW. These findings highlight AutoGFI-BR's power in providing precise and reliable inference across different dimensional settings.

5.4. Covariate-Assisted Ranking Estimation Model for Pairwise Comparison

5.4.1. Introduction. Ranking is fundamentally important in a wide range of real-world scenarios, such as social science, psychology, recommendation systems, and many others. Numerous models and methods have been developed specifically to address the ranking problem across various fields, including statistics, machine learning, and operations research, etc. See (Hunter, 2004; Richardson et al., 2006; Jang et al., 2018; Chen et al., 2022; Liu et al., 2023) for more details. Pair comparison is one of the widely used methods for ranking which simplifies the ranking process by presenting respondents with pairs of items to compare, reducing cognitive burden and providing more information than other ranking methods.

Among the diverse models for pairwise comparison problems, the Bradley-Terry-Luce (BTL) model (Bradley and Terry, 1952; Luce, 2005) stands out as the most famous one. This model assumes that for n compared items, each item score, denoted as θ_i, is fixed and does not take into account any covariate or feature information. The preference between any two items, i and j, is modeled by:

$$P(\text{item } i \text{ is preferred over } j) = \frac{e^{\theta_i}}{e^{\theta_i} + e^{\theta_j}},$$

for all pairs $(i, j) \in \{1, \ldots, n\} \times \{1, \ldots, n\}$. Extensive studies have been done to the BTL model, including analyzing the non-asymptotic statistical consistency of various estimators for these underlying scores to achieve precise rank recoveries (Negahban et al. (2012); Chen and Suh (2015); Chen et al. (2019, 2022)) and exploring the asymptotic distributions and uncertainty quantification for ranking scores (Simons and Yao (1999); Han et al. (2020); Liu et al. (2023); Gao et al. (2023)).

However, in many real-world applications, the integration of covariate information is important. Addressing this need, Fan et al. (2024) introduced the Covariate-Assisted Ranking Estimation (CARE) model. This model extends the BTL framework by assuming the underlying score of the i-th item is composed of $\alpha_i + \boldsymbol{x}_i^\top \boldsymbol{\beta}$, where $\boldsymbol{x}_i^\top \boldsymbol{\beta}$ captures the effect of covariates and α_i represents the intrinsic score not explained by those covariates. The probability that item i is preferred over item j is modeled as:

$$P(\text{i is preferred over j}) = \frac{e^{\alpha_i + \boldsymbol{x}_i^\top \boldsymbol{\beta}}}{e^{\alpha_i + \boldsymbol{x}_i^\top \boldsymbol{\beta}} + e^{\alpha_j + \boldsymbol{x}_j^\top \boldsymbol{\beta}}}. \tag{5.16}$$

In their work, Fan et al. (2024) also analyzed the statistical rate for the maximum likelihood estimator (MLE) of parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and explored their asymptotic distributions. This work paves the way for uncertainty quantification within the CARE model, marking a significant advancement in the field. To the best of our knowledge, their study is the first one and remains the only one that rigorously examines the CARE model.

Our proposed method, AutoGFI-B, is designed to be versatile and applicable across a spectrum of models, including the advanced CARE model. In the following subsections, we delve into how AutoGFI-B integrates with the CARE model and provide empirical evidence through a detailed simulation study. This study aims to demonstrate that AutoGFI-B not only delivers competitive point estimates but also good at uncertainty quantification, offering a robust alternative to the method introduced by Fan et al. (2024).

5.4.2. AutoGFI-B for CARE. In this section, we introduce the CARE model and apply AutoGFI-B to it, noting that this model does not involve any regularization terms.

Letting $\boldsymbol{x}_i \in \mathbb{R}^d$ be the covariate observed for an item $i \in \{1, \ldots, n\}$, the CARE model assumes the outcomes of the pair comparisons between item i and item j as the Bernoulli trials with probabilities as shown in (5.16). Further, in this model, it is assumed that each pair of the items, say $(i, j) \in \{1, \ldots, n\} \times \{1, \ldots, n\}$ are compared at random with probability p_0, leading to the assumption that the underlying comparison graph resembles an Erdős-Rényi random graph characterized by an edge probability of p_0. Here we use

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to represent this comparison graph, where $\mathcal{V} := 1, \ldots, n$ and $\mathcal{E}$ are the collections of vertexes (n items) and edges, respectively. More specifically, $(i, j) \in \mathcal{E}$ if and only if item i and item j are compared. In addition, for any pair $(i, j) \in \mathcal{E}$, K independent and identically distributed realizations from the Bernoulli random variable are observed. In other words, if i and j are compared, then they are compared K times. Letting $Y_{i,j}^{(k)}$ represent the Bernoulli variable where 1 indicates that i is preferred over j in the k-th comparison, then

$$P(Y_{i,j}^{(k)} = 1) = \frac{e^{\alpha_i + \boldsymbol{x}_i^\top \boldsymbol{\beta}}}{e^{\alpha_i + \boldsymbol{x}_i^\top \boldsymbol{\beta}} + e^{\alpha_j + \boldsymbol{x}_j^\top \boldsymbol{\beta}}}, \tag{5.17}$$

where $\boldsymbol{\beta} \in \mathbb{R}^d$ is the feature coefficients and α_i is the intrinsic score for item i. A natural constraint is proposed for this model by Fan et al. (2024) to overcome the identifiability problem due to the overparametrizing. Let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top \in \mathbb{R}^{n \times d}$ and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top$. The CARE model assumes the parameter space is contrained in the set

$$\Theta = \{(\boldsymbol{\alpha}, \boldsymbol{\beta}) : \sum_{i}^{n} \alpha_i = 0 \text{ and } \boldsymbol{X}^\top \boldsymbol{\alpha} = \boldsymbol{0}\} \tag{5.18}$$

to ensure the identifiability. This constraint aligns with the definition of α_i, which represents the intrinsic score of item i that cannot be explained by the covariates.

To accommodate our method to the CARE model, we define f in (5.2) as

$$f((\boldsymbol{x}_i, \boldsymbol{x}_j), \boldsymbol{\theta}) = (\alpha_i + \boldsymbol{x}_i^\top \boldsymbol{\beta}) - (\alpha_j + \boldsymbol{x}_j^\top \boldsymbol{\beta}),$$

where $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$ is the unknown parameter in the model. In addition, we constraint $\boldsymbol{\theta} \in \Theta$ as defined in (5.18). The data generating equation for the CARE model is then

$$Y_{i,j}^{(k)} = \mathbb{1}\{(\alpha_i + \boldsymbol{x}_i^\top \boldsymbol{\beta}) - (\alpha_j + \boldsymbol{x}_j^\top \boldsymbol{\beta}) + L_{i,j}^{(k)} \geq 0\} \text{ with } (\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \Theta,$$

for $(i, j) \in \mathcal{E}$, $k \in \{1, \ldots, K\}$ and $L_{i,j}^{(k)} \overset{i.i.d.}{\sim} \text{Logistic}(0, 1)$. Let $y_{i,j}^{(k)}$ denote the observed value of the random variable $Y_{i,j}^{(k)}$. Plugging f into the sigmoid approximation (5.6) and the relaxed optimization problem (5.7), an approximate fiducial sample $(\boldsymbol{\alpha}, \boldsymbol{\beta})^*$ can be generated by

$$(5.19) \qquad (\boldsymbol{\alpha},\boldsymbol{\beta})^* = \underset{(\boldsymbol{\alpha},\boldsymbol{\beta})\in\Theta}{\arg\min} - \sum_{\substack{i>j,\\ i,j\in\mathcal{E}}} \sum_{k=1}^{K} \left[y_{i,j}^{(k)} \log(y_{i,j}^{(k)\,*}) + (1-y_{i,j}^{(k)}) \log(1-y_{i,j}^{(k)\,*}) \right]$$

where $y_{i,j}^{(k)\,*} = \text{sigmoid}((\alpha_i + \boldsymbol{x}_i^\top \beta) - (\alpha_j + \boldsymbol{x}_j^\top \beta) + l_{i,j}^{(k)\,*}, a)$ for some constant a and $l_{i,j}^{(k)\,*}$ is a realization from $L_{i,j}^{(k)*}$, an independent copy of $L_{i,j}^{(k)}$. The optimization problem (5.19) can be easily solved by the projected gradient descent algorithm introduced in Fan et al. (2024). By substituting the optimization problem in Algorithm (3) with (5.19), an approxiamte fiducial sample of $(\boldsymbol{\alpha},\boldsymbol{\beta})$ can be generated. This sample can then serve as a distribution estimate for the true parameters. Accordingly, point estimates and confidence intervals can be calculated.

5.4.3. Simulation Results. In this section, we evaluate the performance of AutoGFI-B by comparing it to the method given by Fan et al. (2024).

We configured the simulations following the numerical experiments done in Fan et al. (2024). In particular, we set $n = 200$ and $d = 5$. Each feature of the covariates $\boldsymbol{x}_i$ was independently generated from a uniform distribution, with $x_{ij} \sim U[-0.5, 0.5]$ for $i \in 1,\dots,n$ and $j \in 1,\dots,d$. The resulting matrix $\boldsymbol{X} = [\boldsymbol{x}_1,\dots,\boldsymbol{x}_n]^\top \in \mathbb{R}^{n\times d}$ was normalized to ensure a mean of 0 and a standard deviation of 1 for each column. Subsequently, each $\boldsymbol{x}_i$ was scaled by $\boldsymbol{x}_i/K$ to achieve $\max_{1\le i\le n} |\boldsymbol{x}_i|_2/K = \sqrt{(d+1)/n}$. The parameters α_i were independently drawn from $U[0.5, \log(5) - 0.5]$, and $\boldsymbol{\beta} \in \mathbb{R}^d$ was uniformly generated from the hypersphere defined by $\{\boldsymbol{\beta} : |\boldsymbol{\beta}|^2 = 0.5\sqrt{n/(d+1)}\}$. Finally, the true parameter pair $(\boldsymbol{\alpha}_0, \boldsymbol{\beta}_0)$, used for data generation, was obtained by projecting $(\boldsymbol{\alpha},\boldsymbol{\beta})$ onto the linear space Θ specified in (5.18). The comparison graph $\mathcal{E}$ and the observed binary outcomes $y_{i,j}^{(k)}$, for $k = 1,\dots,K$ and $(i,j) \in \mathcal{E}$, were generated for each pair of (p_0, K) listed in Table 5.3. This setup of simulations ensures different statistcal rates of the MLE estimator proposed in the paper Fan et al. (2024).

For each pair of (p_0, K), the graph $\mathcal{E}$ and data $\{y_{i,j}^{(k)}, k = 1,\dots,K, (i,j) \in \mathcal{E}\}$ were simulated 200 times. A fiducial sample of size 500 was produced by Algorithm (3) for each

p_0	1	0.5	0.222	0.625	0.4	0.278
K	50	25	25	5	5	5

TABLE 5.3. Pairs of (p_0, K) considered in the simulation studies.

dataset, with a specified as 100. The fiducial sample mean was used as a point estimate for the parameters $(\boldsymbol{\alpha}_0, \boldsymbol{\beta}_0)$, and percentiles of the sample were used to construct confidence intervals. The summary statistics of the numerical results from AutoGFI-B and the method proposed by Fan et al. (2024), referred to as "MLE", are reported in Tables 5.4 and 5.5. Specifically, Table 5.4 presents the average point estimate error, defined as $\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}_0\|_\infty$, along with the confidence intervals' empirical coverage probabilities and interval widths for $\boldsymbol{\alpha}_0$, across the 200 datasets. Table 5.5 illustrates the average point estimate error, $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_2/\|\boldsymbol{\beta}_0\|_2$, and the coverage probabilities with interval widths for $\boldsymbol{\beta}_0$.

The performance comparison between the two methods indicates very close results. They produce similar errors for both $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$, and their confidence intervals consistently achieve the desired coverage probabilities, with AutoGFI-B often having slightly narrower widths. This simulation result empirically demonstrates the potential of AutoGFI-B as a robust alternative to the method introduced by Fan et al. (2024).

5.5. One-Parameter Logistic Item Response Model: The Rasch Model

5.5.1. Introduction. Item response theory (IRT) refers to a collection of latent variable models and statistical methods used for assessing the likelihood of a specific response to a test or questionnaire, based on the interaction of the examinee's abilities and the item's characteristics. It serves as a powerful tool in many educational and psychological assessments, enabling precise measurement of individual capabilities and facilitating the development of effective testing approaches.

In scenarios where responses are binary, the model is called a dichotomous item response model, in which the response distribution is typically modeled as a function of the examinee's ability and one or more item-descriptive parameters. The Rasch model, proposed by

p_0	K	Method	Error (SE)	Coverage (Width)		
				0.9	0.95	0.99
0.222	25	AutoGFI-B	$0.185_{0.024}$	$0.895_{0.199}$	$0.946_{0.236}$	$0.987_{0.308}$
		MLE	$0.184_{0.024}$	$0.897_{0.200}$	$0.949_{0.238}$	$0.990_{0.313}$
0.278	5	AutoGFI-B	$0.367_{0.052}$	$0.899_{0.396}$	$0.948_{0.472}$	$0.987_{0.615}$
		MLE	$0.368_{0.047}$	$0.899_{0.399}$	$0.950_{0.476}$	$0.991_{0.625}$
0.4	5	AutoGFI-B	$0.302_{0.037}$	$0.894_{0.328}$	$0.944_{0.391}$	$0.986_{0.509}$
		MLE	$0.300_{0.041}$	$0.902_{0.331}$	$0.950_{0.394}$	$0.990_{0.518}$
0.5	25	AutoGFI-B	$0.119_{0.016}$	$0.894_{0.131}$	$0.945_{0.155}$	$0.988_{0.202}$
		MLE	$0.119_{0.016}$	$0.901_{0.132}$	$0.951_{0.157}$	$0.991_{0.206}$
0.625	5	AutoGFI-B	$0.236_{0.031}$	$0.897_{0.261}$	$0.948_{0.311}$	$0.988_{0.405}$
		MLE	$0.241_{0.034}$	$0.900_{0.263}$	$0.951_{0.314}$	$0.990_{0.412}$
1	50	AutoGFI-B	$0.059_{0.027}$	$0.896_{0.065}$	$0.946_{0.077}$	$0.986_{0.101}$
		MLE	$0.058_{0.007}$	$0.902_{0.066}$	$0.951_{0.078}$	$0.991_{0.103}$

TABLE 5.4. The average point estimate error,$\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}_0\|_\infty$, and the coverage probabilities with interval widths for $\boldsymbol{\alpha}_0$, across the 200 datasets.

Rasch (1993), is one of the well-known dichotomous item response models. It uses a single parameter, “difficulty,” to describe the item and assumes the probability that a person with ability θ will give a “correct” response is given by:

$$P(\theta) = \frac{1}{1 + e^{-(\theta - b_j)}}$$

where $j \in 1, \ldots, n$ is the item index and b_j denotes the difficulty parameter of the item. As a result, the Rasch model is also called the one-parameter logistic item response model. Item response models, with the Rasch model as a particular case, have been extensively researched for their effectiveness in measurement, covering various topics such as test calibration, marginal and joint maximum likelihood estimation of ability and item parameters, model adequacy testing, and asymptotic behavior of statistical inference techniques. Interested readers can refer to review papers and books, including works by Hambleton et al. (1991); Fischer and Molenaar (1995); Baker and Kim (2004); Christensen et al. (2013); Bond and Fox (2013), for further details.

p_0	K	Method	Error (SE)	Coverage (Width)		
				0.9	0.95	0.99
0.222	25	AutoGFI-B	$0.0679_{0.0204}$	$0.891_{0.297}$	$0.955_{0.353}$	$0.988_{0.460}$
		MLE	$0.0687_{0.0222}$	$0.895_{0.300}$	$0.946_{0.357}$	$0.990_{0.469}$
0.278	5	AutoGFI-B	$0.136_{0.0411}$	$0.893_{0.594}$	$0.944_{0.706}$	$0.989_{0.921}$
		MLE	$0.132_{0.0399}$	$0.916_{0.598}$	$0.957_{0.713}$	$0.992_{0.937}$
0.4	5	AutoGFI-B	$0.110_{0.0375}$	$0.906_{0.492}$	$0.948_{0.585}$	$0.985_{0.761}$
		MLE	$0.109_{0.0378}$	$0.897_{0.495}$	$0.950_{0.590}$	$0.987_{0.775}$
0.5	25	AutoGFI-B	$0.0439_{0.0134}$	$0.901_{0.196}$	$0.950_{0.233}$	$0.993_{0.303}$
		MLE	$0.0455_{0.0142}$	$0.899_{0.197}$	$0.942_{0.235}$	$0.988_{0.309}$
0.625	5	AutoGFI-B	$0.0898_{0.0284}$	$0.897_{0.391}$	$0.946_{0.465}$	$0.982_{0.604}$
		MLE	$0.0872_{0.0300}$	$0.907_{0.394}$	$0.947_{0.469}$	$0.989_{0.617}$
1	50	AutoGFI-B	$0.0212_{0.0112}$	$0.899_{0.0973}$	$0.956_{0.116}$	$0.992_{0.151}$
		MLE	$0.0213_{0.00746}$	$0.912_{0.0981}$	$0.955_{0.117}$	$0.988_{0.154}$

TABLE 5.5. The average point estimate error, $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_2/\|\boldsymbol{\beta}_0\|_2$, and the coverage probabilities with interval widths for $\boldsymbol{\beta}_0$, across the 200 datasets.

Several works have be done to quantify the uncertainty of the parameters within the Rasch Model. In particular, Klauer (1991) and Doebler et al. (2013) introduced exact confidence intervals for the ability parameter in the Rasch model, where "exact" means the coverage probabilities are guaranteed to be at least the specified nominal level. Mair and Strasser (2018) studies the asymptotic properties of conditional maximum-likelihood estimators for item parameters in the Rasch model as the number of items increases. Liu and Hannig (2016) applies generalized fiducial inference to the two-parameter logistic item response models, enabling the construction of confidence intervals from the generalized fiducial distribution. Veronese and Melilli (2021) proposes the confidence distribution (CD) method for the ability parameter in the Rasch model, providing an approach to quantify the uncertainty associated with the ability estimates.

Given the extensive application and significance of Item Response Theory, we use the Rasch model as our last example to demonstrate the versatility and effectiveness of our proposed AutoGFI-B. In the subsequent subsection, we integrate AutoGFI-B with the Rasch model and conduct a comprehensive simulation study to evaluate its performance.

5.5.2. AutoGFI-B for Rasch Model. In this section, we show the detailed framework of the Rasch Model, which notably does not include a regularization term. Then we discuss the deployment of AutoGFI-B, Algorithm (3) within this context.

The Rasch model analyzes m examinees responding to n items. Each item is described by a difficulty parameter $b_j \in \mathbb{R}$, for $j = 1, \ldots, n$, and each examinee by an ability parameter $\theta_i \in \mathbb{R}$, for $i = 1, \ldots, m$, which determines their position on a latent trait. The response Y_{ij} of the i-th examinee to the j-th item is a bernoulli variable, with the assumption that responses are independent across examinees for each item and across items for each examinee. The probability of a correct response, $Y_{ij} = 1$, is modeled by

$$P(Y_{ij} = 1) = \frac{1}{1 + e^{-(\theta_i - b_j)}}. \tag{5.20}$$

Additionally, following the work of Veronese and Melilli (2021), we consider scenarios where item difficulty parameters, b_j, are treated as fixed. This implies that the values of b_j are either derived from previous tests or initially estimated from the current data, with these estimates subsequently used for deducing θ_i. Adopting fixed item difficulty parameters is a standard practice in the field when focusing on the ability parameter, as supported by references such as Veronese and Melilli (2021); Klauer (1991); Doebler et al. (2013), and Ogasawara (2012). However, this assumption entails a loss of variability in the estimates of b_j. We revisit this consideration is the simulation study, where we demonstrate the robustness of our proposed method against variations in the standard errors of the difficulty parameters. Additionally, we empirically show that our method is capable of simultaneously estimating b_j when it is not readily available.

To adapt AutoGFI-B for the Rasch model, with observed data y_{ij} of the variable Y_{ij} for $(i, j) \in \{1, \dots, m\} \times \{1, \dots, n\}$, we need to define f in (5.2) as

$$f(b_j, \theta_i) = \theta_i - b_j.$$

The data generating equation for the Rasch model is then

$$Y_{i,j} = \mathbb{1}\{\theta_i - b_j + L_{ij} \geq 0\},$$

where $L_{ij} \sim$ Logistic(0,1) for $i = 1, \dots, m$ and $j = 1, \dots, n$. Therefore, with fixed difficulty parameters, an approximate fiducial sample of θ_i for the i-th person can be generated by solving

$$\theta_i^* = \arg\min_{\theta_i} -\sum_{j=1}^{n} \left[y_{ij} \log(y_{ij}^*(\theta_i)) + (1 - y_{ij}) \log(1 - y_{ij}^*(\theta_i)) \right], \tag{5.21}$$

with $y_{ij}^*(\theta_i) = \text{sigmoid}(\theta_i - b_j + l_{ij}^*, a)$ and l_{ij}^* being a realization of L_{ij}^* which is an independent copy of L_{ij}, for $i = 1, \dots, n$. The optimization problem (5.21) is convex and can be easily solved. By substituting the optimization problem in Algorithm (3) with (5.21), we can generate a fiducial sample of $\boldsymbol{\theta} = (\theta_1, \dots, \theta_m)^\top$. This sample can then be used for computing the point estimate and interval estimate.

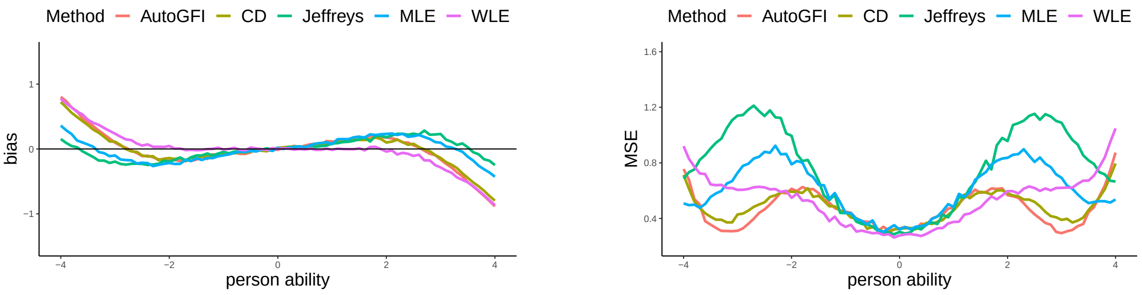
5.5.3. Simulation Results. In this section, we evaluate the performance of AutoGFI-B in point and interval estimation of the ability parameter θ under three different scenarios. First, we assume that the difficulty parameters b_j are known, and we examine the inference performance of AutoGFI-B for various values of θ. Second, we investigate the robustness of AutoGFI-B when the b_j values are corrupted by noise. Finally, we assess AutoGFI-B's ability to simultaneously estimate b_j and infer θ. For the first two scenarios, we compare the performance of AutoGFI-B with that of other standard procedures to benchmark its effectiveness.

In particular, we compared our proposed method with four competing approaches for point estimation: the suitably adjusted maximum likelihood estimator (MLE), the weighted likelihood estimator (WLE) introduced by Warm (1989), the median of the Bayesian posterior distribution using the well-known Jeffreys prior, and the mean of the confidence distribution (CD) proposed by Veronese and Melilli (2021). It is important to note that when $\sum_{j=1}^{n} y_{ij} = 0$ or n, the MLE of θ_i does not exist, and special procedures are necessary, as reviewed in Wright (1998). We adopt the strategy of adding or subtracting a constant c to 0 or n to adjust the extreme scores of $\sum_{j=1}^{n} y_{ij}$. The R package `TAM` (Robitzsch et al., 2024) computes the MLE for item response models using this adjustment with $c = 0.3$, which we followed in our subsequent simulation studies when solving for the MLE and addressing the optimization problem (5.21). The WLE addresses the non-existence of the MLE for extreme scores by multiplying the likelihood function by a weight function $\omega(\theta)$. In the Rasch Model with fixed item parameters, $\omega(\theta) = I_n(\theta)^{1/2}$, where $I_n(\theta)$ denotes the Fisher information. The Jeffreys prior in our settings is $\pi_J(\theta) \propto I_n(\theta)^{1/2}$, coinciding with $\omega(\theta)$ when $\theta \in \mathbb{R}$. Thus, under these conditions, the WLE is essentially the mode of the Jeffreys posterior distribution. According to our experiments, the median of the Jeffreys posterior is more stable than the mean; therefore, we considered using the median as the point estimator for this method. The CD method addresses the boundary values of $\sum_{j=1}^{n} y_{ij}$ in the definition of the confidence distribution with a hyperparameter β. We adopted $\beta = 0.8$ for our simulation studies, following the suggestion in Veronese and Melilli (2021). When employing AutoGFI-B, we chose the scaling parameter a as 100, as we did in the CARE model. For interval estimation, we compared our proposed method with the Wald interval based on the MLE, the Wald interval based on WLE, the equal-tailed interval derived from the Jeffreys posterior, and the equal-tailed interval based on CD.

Scenario 1: Inferring θ with known b_j. In this scenario, we set up the simulation studies following the configurations in Veronese and Melilli (2021). We considered a test with $n = 15$ items, where the difficulty parameters are given by

$$\boldsymbol{b} = (-0.5, -0.5, -0.5, -0.5, -0.5, 0, 0, 0, 0, 0, 0.34, 0.34, 0.34, 0.34, 0.34). \tag{5.22}$$

For the examinees, we considered ability parameters ranging from -4 to 4 with an increment of 0.1, resulting in a total of $m = 81$ examinees with abilities $\theta_i = -4 + 0.1(i-1)$, for $i = 1, \ldots, m$. We then generated the observed response data $\boldsymbol{y} = \{y_{ij}\}$ for $(i, j) \in \{1, \ldots, m\} \times \{1, \ldots, n\}$ 2000 times from the Rasch model (5.20). For each generated dataset, we estimated θ_i assuming the item difficulty parameters b_j were known. To compare the point estimators, we calculated the bias, $\hat{\theta}_i - \theta_i$, and the MSE, $(\hat{\theta}_i - \theta_i)^2$, over the 2000 datasets for each θ_i. These results are presented in Figures 5.3a and 5.3b, respectively. For interval estimators, Figure 5.4 displays the empirical coverage probabilities of the 95% confidence intervals obtained from different methods for each θ_i, while Figure 5.5c shows the widths of these intervals. To summarize the results, Figures 5.5a and 5.5b present boxplots of the average MSEs and empirical coverage probabilities, respectively, across all 81 values of θ.

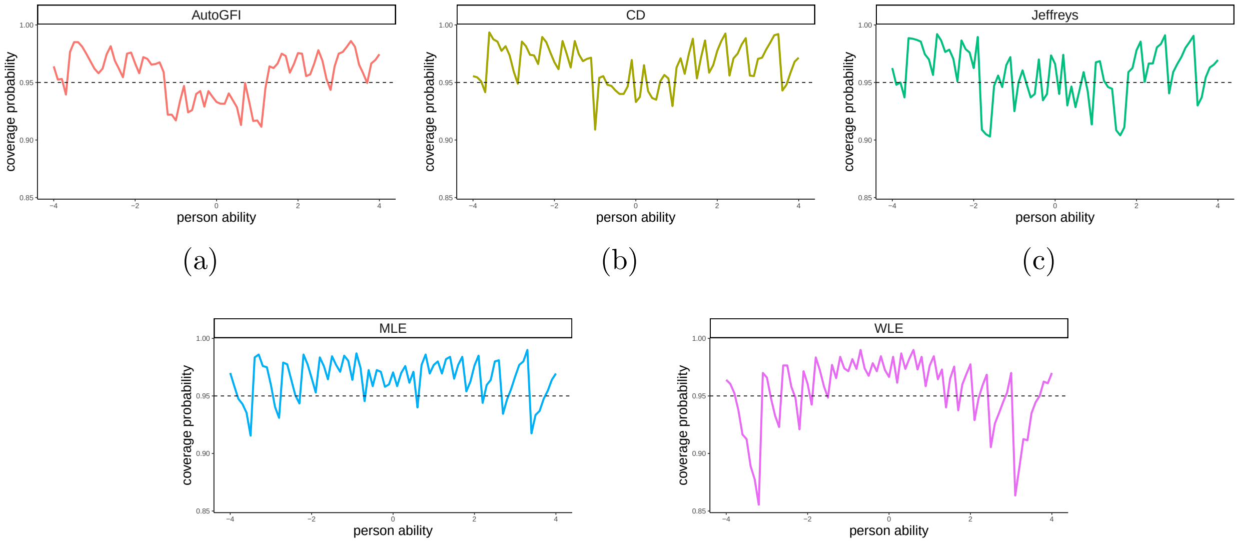


(a)

(b)

FIGURE 5.3. Biases (a) and MSEs (b) of each θ_j estimates for $j = 1, \ldots, 81$, across different methods.

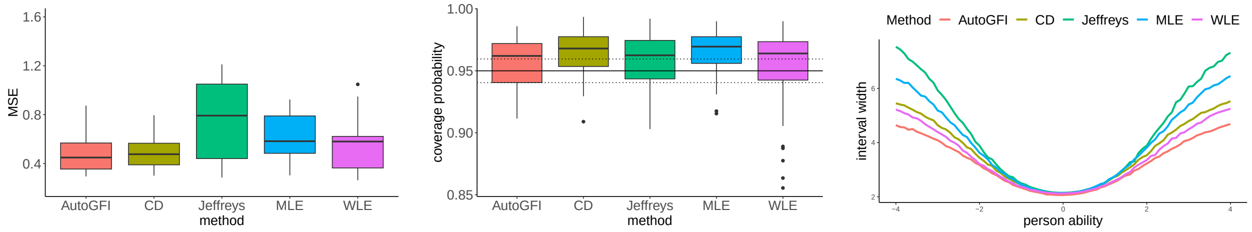
Figure 5.3 demonstrates that AutoGFI-B performs exceptionally well when the largest and smallest values of θ are excluded. AutoGFI-B's point estimator achieves the smallest MSE for non-central values of θ and exhibits an MSE comparable to the smallest one observed with WLE for central values. While WLE is optimal for central values of θ, its performance diminishes as θ deviates from 0. MLE and Jeffreys posterior exhibit similar behaviors, showing greater deviations from the true values as θ moves away from the central region. CD's performance is very close to that of AutoGFI-B but with a slightly larger MSE for non-central values of θ. Interestingly, the biases of AutoGFI-B and CD's estimators behave



(a) (b) (c)

(d) (e)

FIGURE 5.4. Empirical coverage probabilities of 95% confidence intervals for each θ_j, $j = 1, \ldots, 81$, across different methods.



(a) (b) (c)

FIGURE 5.5. (a) Boxplots of MSE for each θ_j estimate, $j = 1, \ldots, 81$. (b) Boxplots of empirical coverage probability for each θ_j, $j = 1, \ldots, 81$, targeting 95% confidence level. (c) Interval width for each θ_j, $j = 1, \ldots, 81$.

in opposite ways, with AutoGFI-B's being slightly larger than CD's due to the biases, $\hat{\theta}_i - \theta_i$, being calculated without uniformly adjusting the sign. The plotted values, averaged over 2000 datasets, may reflect the offsetting effects of positive and negative biases. Figure 5.5a provides a summary plot of the MSEs, clearly demonstrating that AutoGFI-B's point estimator generally offers a smaller MSE for most values of θ compared to other methods.

Regarding empirical coverage probabilities, Figure 5.4 reveals that AutoGFI-B, CD, and Jeffreys posterior generally over-cover for non-central values of θ. In contrast, WLE tends to under-cover these values while over-covering for central values. MLE consistently over-covers across all values of θ. These trends are corroborated by Figure 5.5b, which indicates that the methods tend to over-cover for more than half of the θ values. However, AutoGFI-B remains the method closest to the target level. Furthermore, according to Figure 5.5c, AutoGFI-B consistently provides narrower confidence intervals compared to other methods.

Scenario 2: Inferring θ with noisy b_j. In this scenario, we investigated the robustness of the methods when the item difficulty parameters b_j are subject to variability. Following the studies described by Veronese and Melilli (2021) and references therein (Ogasawara, 2012; Lord, 1975), we modeled b_j using a truncated normal distribution $N(0.5, 1)$, bounded between -1.5 and 2.2, to mimic real-world data. We independently generated b_j values from this distribution for the n items and simulated the observed response data $\boldsymbol{y} = \{y_{ij}\}$ for $(i, j) \in \{1, \ldots, 8\} \times \{1, \ldots, n\}$ using the model specified in (5.20), assuming the true ability parameters $\boldsymbol{\theta} = (-3.5, -2.5, -1.5, 0, 1.5, 2.5, 3.5, 4.5)$. The selection of $\boldsymbol{\theta}$, not symmetrical around zero, accommodates the asymmetry in b_j. This procedure was replicated 1000 times for two small sample cases, $n = 10$ and $n = 20$.

Tables 5.6 and 5.7 present the summarized results for $n = 10$ and $n = 20$, respectively, including the average MSE, empiriccal coverage probability, and interval width with a 95% confidence level for each θ_i across the datasets. The results reveal that AutoGFI-B achieves the smallest average MSE for estimating θ when the extreme values are excluded. Moreover, the empirical coverage probabilities exhibit similar trends to those observed in Scenario 1, with AutoGFI-B, CD, and Jeffreys posterior over-covering for non-central values of θ, while MLE and WLE tend to under-cover. Remarkably, AutoGFI-B's confidence intervals average closest to the target level of 95% and consistently provide the narrowest interval widths compared to other methods. These findings highlight the robustness of AutoGFI-B in accommodating potential variations in the estimates of b_j. Despite the introduction of noise in the item difficulty parameters, AutoGFI-B maintains its performance in both point and interval estimation.

When the item difficulty parameters b_j are unknown in practice, well-established methods such as the Conditional Maximum Likelihood (CML) estimator and the Nonparametric Marginal Maximum Likelihood (NP-MML) estimator can be employed to estimate them. Cohen et al. (2008) reported that these two estimators yield nearly identical standard errors for b_j. For a set of 15 items based on 200 observations, the average standard error is approximately 0.2. As the number of observations increases to 1000, the standard error decreases to 0.09, as demonstrated in Table 2 of Cohen et al. (2008). The simulated case in Scenario 2 showcases the robustness of AutoGFI-B to the randomness in b_j. Even when a noise of magnitude 0.2 was added to the b_j values in (5.22), the results remained largely unchanged and are therefore omitted here for brevity.

θ	-3.5	-2.5	-1.5	0	1.5	2.5	3.5	4.5	mean
MSE									
AutoGFI-B	2.131	0.534	0.434	0.647	0.657	0.626	0.389	0.907	0.791
CD	1.897	0.541	0.511	0.607	0.626	0.633	0.471	0.840	0.765
Jeffreys	0.874	0.849	1.234	0.624	0.653	0.977	1.125	0.721	0.882
MLE	1.076	0.554	0.869	0.655	0.678	0.851	0.756	0.581	0.752
WLE	1.922	0.722	0.668	0.495	0.523	0.635	0.692	0.981	0.830
Coverage									
AutoGFI-B	0.989	0.934	0.971	0.926	0.921	0.963	0.963	0.967	0.954
CD	0.985	0.952	0.975	0.962	0.952	0.971	0.966	0.968	0.966
Jeffreys	0.988	0.962	0.982	0.953	0.945	0.960	0.965	0.967	0.965
MLE	0.960	0.934	0.970	0.973	0.974	0.967	0.963	0.967	0.964
WLE	0.844	0.928	0.959	0.981	0.979	0.961	0.946	0.937	0.942
Interval Width									
AutoGFI-B	4.999	4.771	4.155	2.872	2.782	3.425	4.271	4.825	4.012
CD	5.802	5.430	4.516	2.984	2.952	3.725	4.833	5.612	4.482
Jeffreys	8.195	7.414	5.569	3.074	2.967	4.006	5.900	7.439	5.570
MLE	6.801	6.265	4.971	3.008	2.958	3.915	5.401	6.516	4.979
WLE	5.678	5.279	4.324	2.902	2.867	3.562	4.649	5.482	4.343

TABLE 5.6. MSEs, empirical coverage probabilities, and interval widths for different values of θ with random b_j, using a 95% confidence level, in the case $n = 10$.

θ	-3.5	-2.5	-1.5	0	1.5	2.5	3.5	4.5	mean
					MSE				
AutoGFI-B	0.702	0.352	0.584	0.296	0.295	0.480	0.487	0.326	0.440
CD	0.664	0.440	0.570	0.289	0.289	0.454	0.517	0.394	0.452
Jeffreys	0.674	1.172	0.849	0.272	0.277	0.498	0.989	0.952	0.710
MLE	0.491	0.772	0.748	0.291	0.291	0.509	0.792	0.604	0.562
WLE	0.895	0.624	0.529	0.249	0.255	0.375	0.552	0.627	0.513
					Coverage				
AutoGFI-B	0.967	0.950	0.963	0.939	0.922	0.950	0.978	0.975	0.955
CD	0.965	0.956	0.965	0.948	0.938	0.973	0.983	0.979	0.963
Jeffreys	0.965	0.957	0.959	0.948	0.934	0.948	0.979	0.977	0.958
MLE	0.965	0.940	0.960	0.961	0.950	0.968	0.973	0.971	0.961
WLE	0.938	0.933	0.949	0.966	0.958	0.962	0.954	0.899	0.945
					Interval Width				
AutoGFI-B	4.825	4.336	3.250	1.961	1.929	2.492	3.570	4.410	3.347
CD	5.412	4.645	3.813	7.144	2.028	2.609	3.832	4.977	4.308
Jeffreys	7.512	5.964	3.614	2.050	1.986	2.616	4.285	6.319	4.293
MLE	6.339	5.235	3.442	2.020	1.989	2.642	4.134	5.708	3.939
WLE	5.152	4.354	3.067	1.996	1.970	2.490	3.569	4.706	3.413

TABLE 5.7. MSEs, empirical coverage probabilities, and interval widths for different values of θ with random b_j, using a 95% confidence level, in the case $n = 20$.

Scenario 3: Inferring θ with unknown b_j. A notable feature of AutoGFI-B is its ability to simultaneously provide inference for θ_i and estimate b_j when a sufficient number of examinees are available, distinguishing it from the other methods previously considered. This can be achieved by generating fiducial samples for both $\boldsymbol{\theta} = (\theta_1, \dots, \theta_m)^\top$ and $\boldsymbol{b} = (b_1, \dots, b_n)^\top$ through minimizing the objective function in (5.23) for both $\boldsymbol{\theta}$ and $\boldsymbol{b}$ simultaneously:

$$-\sum_{i=1}^{m}\sum_{j=1}^{n}\left[y_{ij}\log(y_{ij}^*) + (1 - y_{ij})\log(1 - y_{ij}^*)\right], \tag{5.23}$$

where $y_{ij}^* = \text{sigmoid}(\theta_i - b_j + L_{ij}^*, a)$, and L_{ij}^* is an independent copy of $L_{ij} \sim \text{Logistic}(0, 1)$. This optimization problem can be effectively addressed using an alternating optimization strategy, where item and person parameters are updated sequentially—optimizing one while holding the other fixed—to iteratively converge on stable estimates. Inference for θ_i is then conducted by analyzing the fiducial distribution formed by the samples, and b_j is estimated by its fiducial sample mean. We conducted simulation studies to evaluate the performance of AutoGFI-B in simultaneously estimating b_j and inferring θ_i. We considered $n = 15$ items with b_j defined in (5.22) and m examinees with ability parameters θ_i independently generated from a standard normal distribution $N(0, 1)$. The observed data y_{ij} were generated from the model (5.20) for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. We considered three cases for m: $m = 100$, 500, and 1000, with 200 replicated datasets generated for each case. A fiducial sample of size 500 for $(\boldsymbol{\theta}, \boldsymbol{b})$ was produced for each dataset, with the scale parameter for AutoGFI-B chosen as 100.

The sample mean was used to estimate $\boldsymbol{b}$, and the average MSE of each value of b_j over the 200 datasets is reported in Table 5.8. It is evident that AutoGFI-B provides highly accurate estimates of item difficulty parameters with very small MSE.

	b		
m	-0.5	0	0.34
100	0.0612	0.0570	0.0643
500	0.0113	0.0134	0.0174
1000	0.0060	0.0097	0.0125

Table 5.8. Average MSE for b_j with different number of examinees m.

Similarly, a point estimate and a 95% confidence interval were constructed for each θ_i. The average MSE of the point estimate and the average empirical coverage probabilities with interval widths for the confidence intervals are summarized in Table 5.9. Figure 5.6 specifically presents the boxplot of the empirical coverage probabilities across θ_i, for $i = 1, \ldots, m$. Both Table 5.9 and Figure 5.6 demonstrate that AutoGFI-B consistently achieves the desired confidence level in its confidence intervals for the ability parameter.

These findings demonstrate the versatility and robustness of AutoGFI-B in handling situations where both item and person parameters are unknown. By leveraging an alternating optimization strategy and generating fiducial samples for both $\boldsymbol{\theta}$ and $\boldsymbol{b}$, AutoGFI-B is able

		95% CI	
m	MSE	Coverage	Width
100	0.542	0.940	2.629
500	0.490	0.947	2.551
1000	0.483	0.948	2.537

TABLE 5.9. Average MSE, empirical coverage probability, and interval width for θ when b_js are unknown.

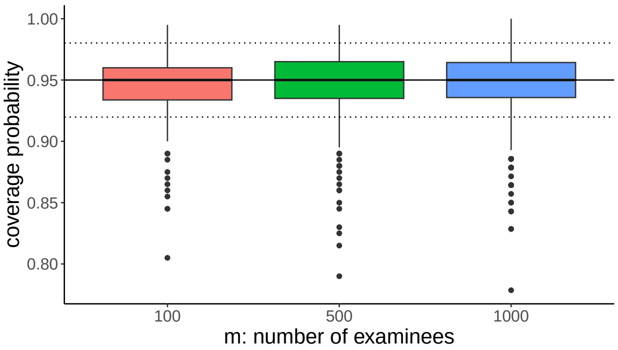


FIGURE 5.6. Boxplot of empirical coverage probabilities for θ_i, $i = 1, \ldots, m$, across three sample sizes: $m = 100$, 500, and 1000, all targeting a 95% confidence level.

to offer a comprehensive solution for simultaneously estimating item difficulty and inferring individual abilities in the Rasch model.

5.6. Conclusion

In this chapter, we introduced AutoGFI-B and its regularized version, AutoGFI-BR. These two algorithms serve as an extension of the AutoGFI algorithm, adapting its beneficial properties from additive noise models to binary response models. This extension greatly simplifies and broadens the application of GFI in real-world scenarios involving binary data. We conducted numerical studies on three challenging binary problems: (high-dimensional) logistic regression, the covariate-assisted ranking estimation (CARE) model, and the Rasch model. The results demonstrated the competitive performance of both AutoGFI-B and AutoGFI-BR compared to established methods in terms of point estimation accuracy, confidence interval coverage, and interval width. These findings highlight the potential of

AutoGFI-B and AutoGFI-BR as powerful and flexible tools for inference in such binary models. Moreover, the results further showcase GFI as a promising alternative to traditional methods for addressing important and practical inference problems. With AutoGFI, AutoGFI-B, and AutoGFI-BR, we provide accessible implementations of GFI, facilitating its application in complex analytical scenarios.

CHAPTER 6

Summary and Future Prospects

In this dissertation, we have presented three novel approaches designed to streamline and simplify the implementation of GFI in modern statistical problems. First, we introduced AutoGFI for additive noise models, followed by Fiducial Selector for high-dimensional linear regression problems. We then presented AutoGFI-B for binary response models and its regularized version, AutoGFI-BR. The key innovation of these methods lies in transforming the traditional sampling process of GFI, which typically involves complex mathematical derivations and MCMC techniques, into a series of optimization problems. This transformation significantly improves GFI's accessibility for researchers and practitioners working with complex models. The theoretical and empirical evaluations presented in this dissertation have validated the effectiveness of these new approaches, showcasing the substantial potential of GFI in addressing modern inference challenges. In summary, by streamlining the adoption of GFI, this work expands the toolkit available for robust statistical inference in many modern problems.

However, we believe that this work is just the beginning of simplifying the adoption of GFI. To make GFI accessible to more researchers, we need to adapt it to more application scenarios and further improve its performance. Potential future work includes extending AutoGFI to handle more types of data, such as count data, accommodating different model structures, and refining the de-biasing procedure within the algorithm. With continued efforts to refine and extend these methods, we hope that GFI will be more widely adopted and utilized by researchers and practitioners in tackling modern statistical problems.

APPENDIX A

Appendix for Chapter 3

This appendix outlines the proofs for Theorems 3.3.2 to 3.3.6 in Chapter 3.

Proof for Theorem 3.3.2. Select $\epsilon > 0$ and set $C = \min(h(\epsilon)/2, \epsilon)$. By Assumption 3.3.1(c),

$$P\left(n^{-1}\sup_{\boldsymbol{\theta}\in K}\|\sum_{i=1}^{n} -\nabla_{\boldsymbol{\theta}} G(\boldsymbol{X}_i, \boldsymbol{\theta})U_i + \boldsymbol{\xi}_n(\boldsymbol{\theta})\| \geq C\right) \to 0.$$

Since (3.5) is equivalent to

$$n^{-1}\sum_{i=1}^{n}\nabla_{\boldsymbol{\theta}} G(\boldsymbol{X}_i, \hat{\boldsymbol{\theta}})(G(\boldsymbol{X}_i, \boldsymbol{\theta}_0) - G(\boldsymbol{X}_i, \hat{\boldsymbol{\theta}})) = n^{-1}\boldsymbol{\xi}_n(\hat{\boldsymbol{\theta}}) - n^{-1}\sum_{i=1}^{n}\nabla_{\boldsymbol{\theta}} G(\boldsymbol{X}_i, \hat{\boldsymbol{\theta}})U_i,$$

there is a solution $\hat{\boldsymbol{\theta}} \in K$ with probability going to 1 by Assumption 3.3.1(a). At the same time, if $\hat{\boldsymbol{\theta}} \in K$ and $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\| \geq \epsilon$, then

$$n^{-1}\|\sum_{i=1}^{n}\nabla_{\boldsymbol{\theta}} G(\boldsymbol{X}_i, \hat{\boldsymbol{\theta}})(G(\boldsymbol{X}_i, \boldsymbol{\theta}_0) - G(\boldsymbol{X}_i, \hat{\boldsymbol{\theta}}))\| \geq h(\epsilon)$$

with probability going to 1. By the union bound $P(\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\| \geq \epsilon) \to 0$.

The same argument also shows that $P(\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_0\| \geq \epsilon) \to 0$. □

Proof for Theorem 3.3.4. The same argument as in the proof of Theorem 3.3.2 shows that $\hat{\boldsymbol{\theta}}_0 \xrightarrow{P} \boldsymbol{\theta}_0$.

Rewrite (3.5) using (3.7) to obtain

$$\boldsymbol{T}_n(\hat{\boldsymbol{\theta}}_0)n^{1/2}(\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}_0) + n^{1/2}\boldsymbol{R}_n(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}_0) = n^{-1/2}\sum_{i=1}^{n}\nabla_{\boldsymbol{\theta}} G(\boldsymbol{X}_i, \hat{\boldsymbol{\theta}})U_i.$$

By Theorem 3.3.2 $\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}_0 \xrightarrow{P} 0$ and therefore Assumption 3.3.3(b) implies that $n^{1/2}\boldsymbol{R}_n(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}_0) = o_p(\|n^{1/2}(\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}_0)\|)$. Finally, Assumptions 3.3.3(a) and 3.3.1(c) and Slutsky's lemma imply that $n^{1/2}(\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}_0) \xrightarrow{\mathcal{D}} N(\mathbf{0}, \boldsymbol{T}_\infty^{-1}\boldsymbol{S}(\boldsymbol{\theta}_0)\boldsymbol{T}_\infty^{-1\top})$. The second part of the theorem is proved similarly. □

Proof for Theorem 3.3.6. Notice that

$$\begin{aligned} n^{1/2}(\boldsymbol{\theta}^*_{\mathrm{de}} - \hat{\boldsymbol{\theta}}_{\mathrm{de}}) &= n^{1/2}(\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}) \\ &\quad + n^{1/2}\left(\mathbf{H}(\boldsymbol{\theta}^*)^{\mathrm{pinv}} - \mathbf{H}_0(\boldsymbol{\theta}^*)^{\mathrm{pinv}}\right)\boldsymbol{\xi}_n(\boldsymbol{\theta}^*) \\ &\quad + n^{1/2}\left(\mathbf{H}_0(\boldsymbol{\theta}^*)^{\mathrm{pinv}}\boldsymbol{\xi}_n(\boldsymbol{\theta}^*) - \mathbf{H}_0(\hat{\boldsymbol{\theta}})^{\mathrm{pinv}}\boldsymbol{\xi}_n(\hat{\boldsymbol{\theta}})\right) \\ &\quad + n^{1/2}\left(\mathbf{H}_0(\hat{\boldsymbol{\theta}})^{\mathrm{pinv}} - \hat{\mathbf{H}}(\hat{\boldsymbol{\theta}})^{\mathrm{pinv}}\right)\boldsymbol{\xi}_n(\hat{\boldsymbol{\theta}}). \end{aligned}$$

The claimed asymptotic normality now follows using Slutsky's lemma, Theorem 3.3.4 and Assumption 3.3.5. The argument for $n^{1/2}(\hat{\boldsymbol{\theta}}_{\mathrm{de}} - \hat{\boldsymbol{\theta}}_{0,\mathrm{de}})$ is analogous.

Next, Taylor's theorem and (3.6) imply

$$-\boldsymbol{\xi}_n(\hat{\boldsymbol{\theta}}_0) = \left(\int_0^1 \mathbf{H}_0(v\hat{\boldsymbol{\theta}}_0 + (1-v)\boldsymbol{\theta}_0)\, dv\right)(\hat{\boldsymbol{\theta}}_0 - \boldsymbol{\theta}_0).$$

Consequently,

$$\|\hat{\boldsymbol{\theta}}_{0,\mathrm{de}} - \boldsymbol{\theta}_0\| \le R_n\,\|\hat{\boldsymbol{\theta}}_0 - \boldsymbol{\theta}_0\|.$$

This concludes the proof. □

APPENDIX B

Appendix for Chapter 4

This appendix outlines the proofs for Theorems 4.3.2 and 4.3.3 in Chapter 4. Additionally, it includes supplementary simulation results for Fiducial Selector, further illustrating its performance and applicability.

B.1. Theoretical Proofs

We first cite the following two theorem from Zhao and Yu (2006).

THEOREM B.1.1 (Small p and s). *For fixed s, p and $\beta^n = \beta$, under the strong irrepresentable condition (Definition 4.3.1) and regularity conditions (4.13) and (4.14), lasso is strongly sign consistent, i.e. for $\forall \lambda_n$ s.t. $\frac{\lambda_n}{n} \to 0$ and $\frac{\lambda_n}{n^{\frac{1+c}{2}}} \to \infty$ with $0 \le c < 1$, we have*

$$P(\hat{\boldsymbol{\beta}}^n(\lambda_n) =_s \boldsymbol{\beta}^n) = 1 - o(e^{-n^c}),$$

where $=_s$ stands for sign equivalence, i.e. $\boldsymbol{a} =_s \boldsymbol{b} \leftrightarrow sign(\boldsymbol{a}) = sign(\boldsymbol{b})$.

THEOREM B.1.2 (large p and s). *Assume the error term has finite $2k$-th moment for an integer $k > 0$. Under the strong irrepresentable condition (Definition 4.3.1) and regularity conditions (4.15), (4.16), (4.17) and (4.18), lasso is strongly sign consistent for $p_n = o(n^{(c_2 - c_1)k})$. To be more specific, for $\forall \lambda_n$ s.t. $\frac{\lambda_n}{\sqrt{n}} = o(n^{\frac{c_2 - c_1}{2}})$ and $\frac{1}{p_n}(\frac{\lambda_n}{\sqrt{n}})^{2k} \to \infty$, we have*

$$P(\hat{\boldsymbol{\beta}}^n(\lambda_n) =_s \boldsymbol{\beta}^n) \ge 1 - O(\frac{p_n n^k}{\lambda_n^{2k}}) \to 1 \text{ as } n \to \infty.$$

PROOF OF THEOREM 4.3.2. In the fiducial sample generation procedure described in Section 4.2.2, each $\boldsymbol{\beta}^*$ in the biased fiducial sample is derived using the lasso estimator applied to $(\boldsymbol{y}', \boldsymbol{X})$, where $\boldsymbol{y}'$ comes from the equation

$$\boldsymbol{y}' = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

with $\boldsymbol{\epsilon} = \boldsymbol{e} - \boldsymbol{e}^*$. Here, $\boldsymbol{e}$ represents the i.i.d. random normal errors from the original observations, and $\boldsymbol{e}^* = \sigma \boldsymbol{u}^*$, which follows the normal distribution $N(0, \sigma \mathbf{I}_n)$, is the random term introduced during the fiducial sample generation. Note that $\boldsymbol{e}$ remains the same throughout the fiducial sample generation process, while $\boldsymbol{e}^*$ keeps changing.

By applying Theorem B.1.1 and Theorem B.1.2, given the corresponding conditions listed in the theorems are satisfied, we have the following equation hold for both 'small p and s' and 'large p and s' cases:

$$P(\boldsymbol{\beta}^* =_s \boldsymbol{\beta}) \to 1 \text{ as } n \to \infty,$$

where $\boldsymbol{\beta}^*$ represents an arbitrary copy in the un-debiased fiducial sample of the fiducial selector.

Since $P(\boldsymbol{\beta}^* =_s \boldsymbol{\beta}) \leq P(\{j : \beta_j^* \neq 0\} = \mathcal{A})$, where the subscript j represents the j-th element of the vector. Therefore, we have,

$$P(\{j : \beta_j^* \neq 0\} = \mathcal{A}) \to 1 \quad \text{as} \quad n \to \infty. \tag{B.1}$$

Denote $P(\{j : \beta_j^* \neq 0\} \neq \mathcal{A}) = \alpha$. We have $\alpha \to 0$ as $n \to \infty$. Recall that we use 50% as the threshold to identify the set of significant parameters. Thus

$$\begin{aligned} P(\hat{\mathcal{A}} = \mathcal{A}) &\geq P(\Sigma_{k=1}^N \mathbb{1}(\{j : \beta_{(k),\,j}^* \neq 0\} \neq \mathcal{A}) \leq \lfloor \frac{N}{2} \rfloor) \\ &= 1 - P(\Sigma_{k=1}^N \mathbb{1}(\{j : \beta_{(k),\,j}^* \neq 0\} \neq \mathcal{A}) \geq \lceil \frac{N}{2} \rceil), \end{aligned} \tag{B.2}$$

where $\hat{\mathcal{A}}$ is the identified significant set defined in (4.10), $\beta_{(k),\,j}^*$ stands for the j-th element of the k-th copy in the biased fiducial sample of fiducial selector, $\mathbb{1}(\cdot)$ is the indicator function. Let $\Psi = (\psi_1, \ldots, \psi_{\binom{N}{\lceil \frac{N}{2} \rceil}})$, where each ψ_i is a combination of indexes from $\{1, \ldots, N\}$, such that $|\psi_i| = \lceil \frac{N}{2} \rceil$, $i = 1, \ldots, \binom{N}{\lceil \frac{N}{2} \rceil}$. Then we have

$$\begin{aligned} &P\left(\Sigma_{k=1}^N \mathbb{I}\left(\{j : \beta_{(k),\,j}^* \neq 0\} \neq \mathcal{A}\right) \geq \lceil \frac{N}{2} \rceil\right) \\ &\leq \sum_{j}^{\binom{N}{\lceil \frac{N}{2} \rceil}} P(\bigcap_{k \in \psi_j} (\{j : \beta_{(k),\,j}^* \neq 0\} \neq \mathcal{A})) \leq \sum_{j}^{\binom{N}{\lceil \frac{N}{2} \rceil}} \alpha = \alpha \binom{N}{\lceil \frac{N}{2} \rceil}. \end{aligned} \tag{B.3}$$

From (B.2) and (B.3), we have

$$P(\hat{\mathcal{A}} = \mathcal{A}) \geq 1 - \alpha \binom{N}{\lceil \frac{N}{2} \rceil} \to 1 \text{ as } n \to \infty, \tag{B.4}$$

which completes our proof. □

Proof of Theorem 4.3.3. :

From (4.11) we have

$$\begin{aligned} \hat{\boldsymbol{\beta}}_{\hat{\boldsymbol{\mathcal{A}}}} &= (\boldsymbol{X}_{\hat{\boldsymbol{\mathcal{A}}}}^{\boldsymbol{T}} \boldsymbol{X}_{\hat{\boldsymbol{\mathcal{A}}}})^{-1} \boldsymbol{X}_{\hat{\boldsymbol{\mathcal{A}}}}^{\boldsymbol{T}} (\boldsymbol{X}_{\hat{\mathcal{A}}} \boldsymbol{\beta}_{\hat{\mathcal{A}}} + \boldsymbol{X}_{\hat{\mathcal{A}}^c} \boldsymbol{\beta}_{\hat{\mathcal{A}}^c} + \boldsymbol{e}) + \hat{\sigma} (\boldsymbol{X}_{\hat{\mathcal{A}}}^{T} \boldsymbol{X}_{\hat{\mathcal{A}}})^{-1/2} \hat{\boldsymbol{S}}_{\boldsymbol{\beta}_{\hat{\boldsymbol{\mathcal{A}}}}^{*}}^{-1/2} (\boldsymbol{\beta}_{\hat{\boldsymbol{\mathcal{A}}}}^{*} - \overline{\boldsymbol{\beta}}_{\hat{\boldsymbol{\mathcal{A}}}}^{*}) \\ &= \boldsymbol{\beta}_{\hat{\mathcal{A}}} + (\boldsymbol{X}_{\hat{\boldsymbol{\mathcal{A}}}}^{\boldsymbol{T}} \boldsymbol{X}_{\hat{\boldsymbol{\mathcal{A}}}})^{-1} \boldsymbol{X}_{\hat{\boldsymbol{\mathcal{A}}}}^{\boldsymbol{T}} \boldsymbol{e} + \hat{\sigma} (\boldsymbol{X}_{\hat{\mathcal{A}}}^{T} \boldsymbol{X}_{\hat{\mathcal{A}}})^{-1/2} S_{\boldsymbol{\beta}_{\hat{\boldsymbol{\mathcal{A}}}}^{*}}^{-1/2} (\boldsymbol{\beta}_{\hat{\boldsymbol{\mathcal{A}}}}^{*} - \overline{\boldsymbol{\beta}}_{\hat{\boldsymbol{\mathcal{A}}}}^{*}), \end{aligned} \tag{B.5}$$

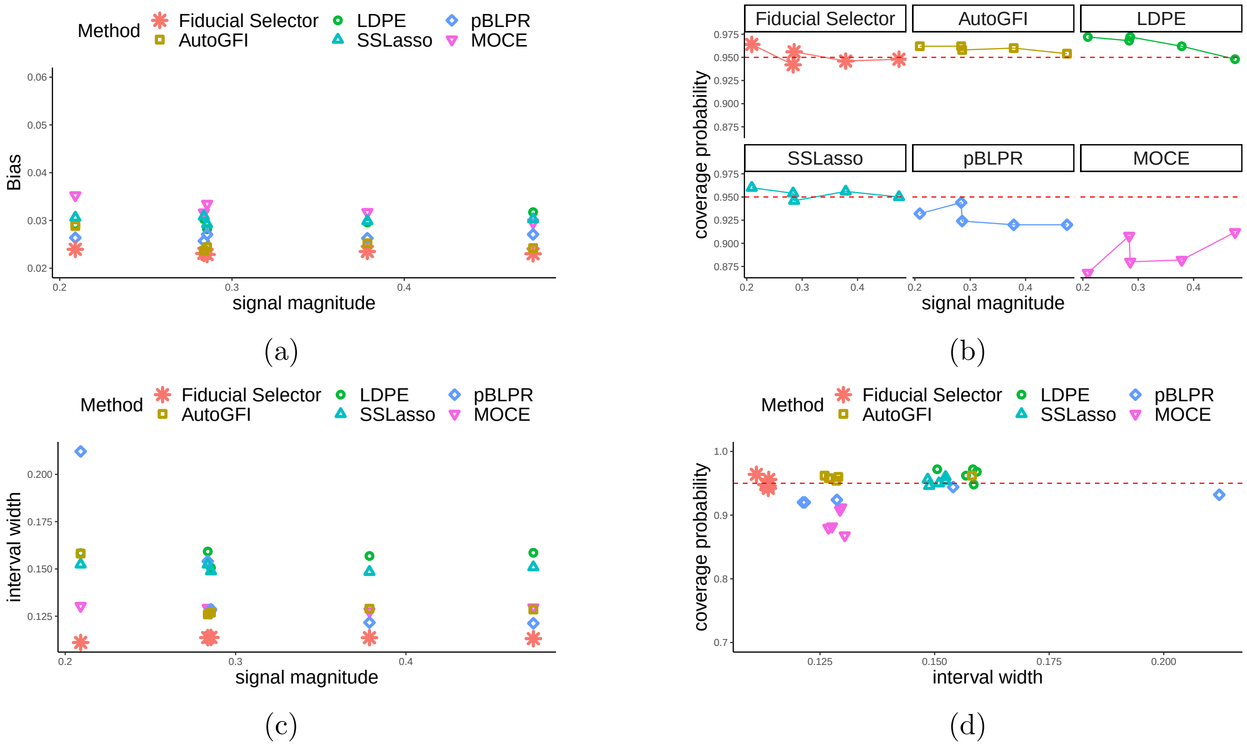
which implies that,

$$E[\text{mean}(\hat{\boldsymbol{\beta}}_{\hat{\boldsymbol{\mathcal{A}}}})] = \boldsymbol{\beta}_{\hat{\mathcal{A}}}. \tag{B.6}$$

□

B.2. Supplementary Simulation Results

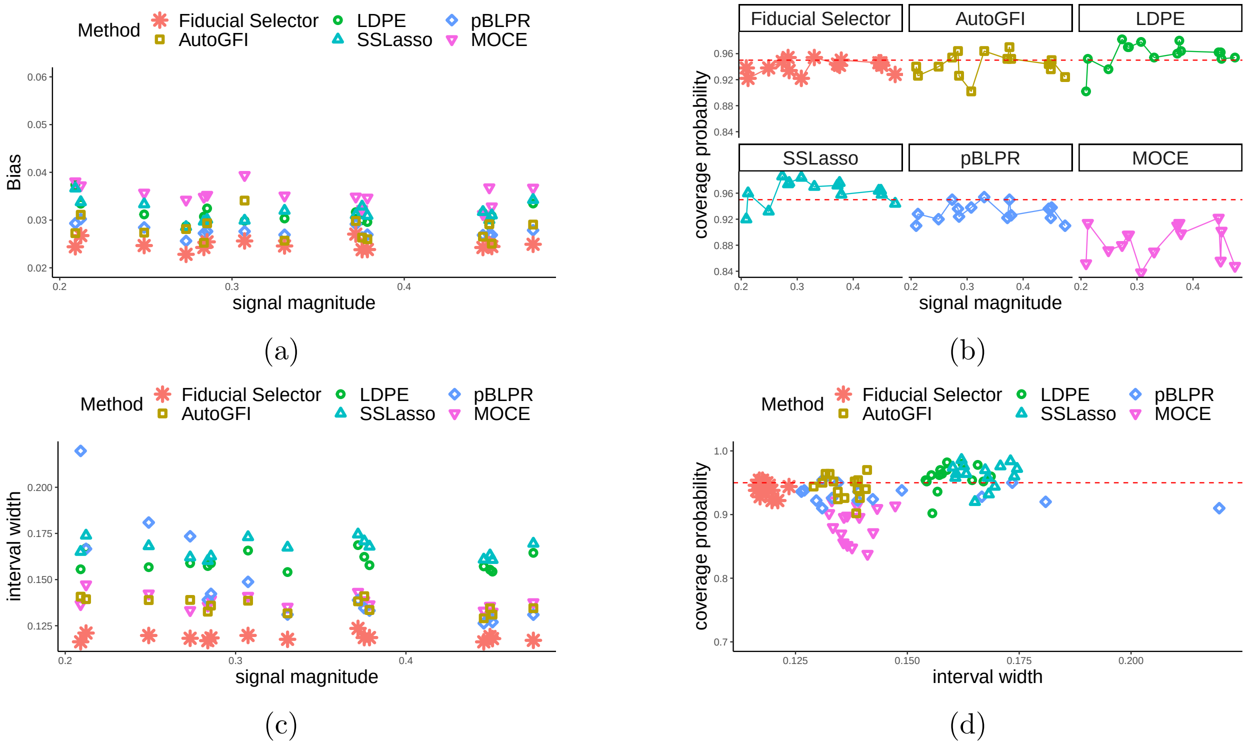
Building on the analysis initiated in Section 4.4, this appendix further explores the relationships previously examined between signal magnitude and bias, signal magnitude and empirical coverage probabilities, signal magnitude and interval widths, as well as empirical coverage probabilities and interval widths. While the main text focused on the most challenging scenarios involving $n = 300$, $p = 1000$, $s = 5$ and $s = 15$ with $\rho = 0.5$, here we present additional cases with $n = 300$, $p = 400, 600$, $s = 5, 15$, and $\rho = 0.5$ to provide a comprehensive view of the observed trends across different settings.



(a) (b)

(c) (d)

FIGURE B.1. Scenario: $n = 300$, $p = 400$, $\rho = 0.5$ and $s = 5$.



(a) (b)

(c) (d)

FIGURE B.2. Scenario: $n = 300$, $p = 400$, $\rho = 0.5$ and $s = 15$.

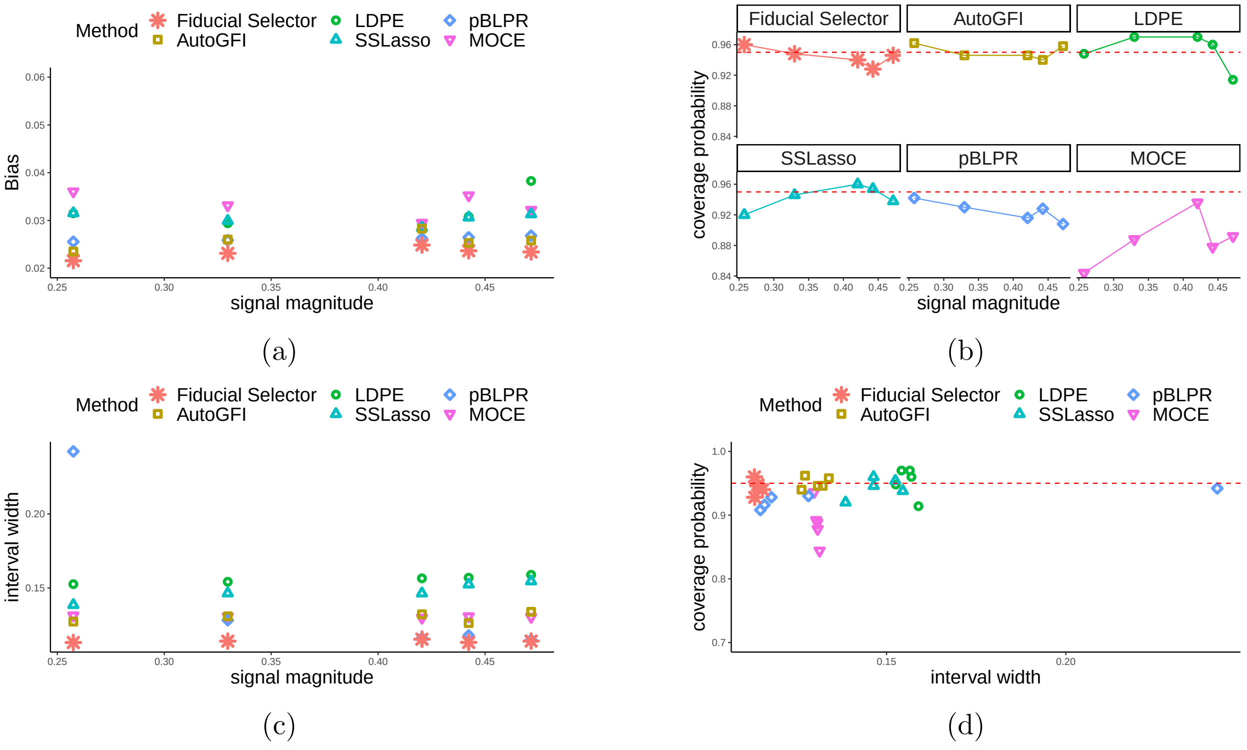


FIGURE B.3. Scenario: $n = 300$, $p = 600$, $\rho = 0.5$ and $s = 5$.

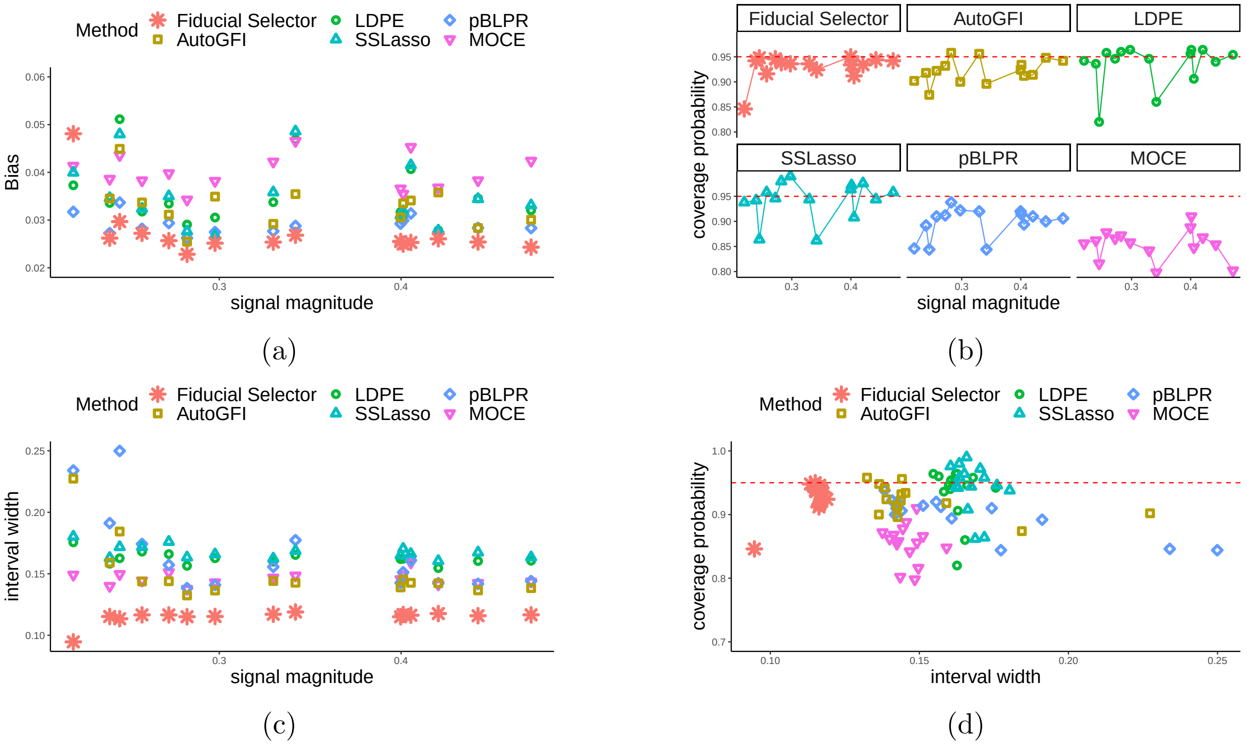


FIGURE B.4. Scenario: $n = 300$, $p = 600$, $\rho = 0.5$ and $s = 15$.

Bibliography

Baker, F. B. and S.-H. Kim (2004). *Item response theory: Parameter estimation techniques.* CRC press.

Barbieri, M. M. and J. O. Berger (2004). Optimal predictive model selection. *The Annals of Statistics 32*, 870–897.

Battey, H. S. and N. Reid (2023). On inference in high-dimensional regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology 85*(1), 149–175.

Beaumont, M. A., W. Zhang, and D. J. Balding (2002). Approximate Bayesian computation in population genetics. *Genetics 162*(4), 2025–2035.

Bennett, J. and S. Lanning (2007). The Netflix prize. In *Proceedings of KDD Cup and Workshop*, Volume 2007, pp. 35. New York, NY, USA.

Binkiewicz, N., J. T. Vogelstein, and K. Rohe (2017). Covariate-assisted spectral clustering. *Biometrika 104*(2), 361–377.

Bond, T. G. and C. M. Fox (2013). *Applying the Rasch model: Fundamental measurement in the human sciences.* Psychology Press.

Bradley, R. A. and M. E. Terry (1952). Rank analysis of incomplete block designs the method of paired comparisons. *Biometrika 39*, 324–345.

Bühlmann, P. (2013). Statistical significance in high-dimensional linear models. *Bernoulli 19*, 1212–1242.

Bühlmann, P., M. Kalisch, and L. Meier (2014). High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Application 1*, 255–278.

Bunea, F. (2008). Honest variable selection in linear and logistic regression models via ℓ_1 and $\ell_1 + \ell_2$ penalization. *Electronic Journal of Statistics 2*, 1153 – 1194.

Candes, E. and T. Tao (2007). The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics 35*(6), 2313 – 2351.

Candès, E. J., X. Li, Y. Ma, and J. Wright (2011). Robust principal component analysis? *Journal of the ACM (JACM) 58*(3), 11.

Candès, E. J. and Y. Plan (2010). Matrix completion with noise. *Proceedings of the IEEE 98*(6), 925–936.

Candès, E. J. and B. Recht (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics 9*(6), 717.

Candès, E. J. and P. Sur (2020). The phase transition for the existence of the maximum likelihood estimate in high-dimensional logistic regression. *The Annals of Statistics 48*(1), 27 – 42.

Casella, G. and R. L. Berger (2024). *Statistical Inference* (2 ed.). Chapman and Hall/CRC.

Chatterjee, A. and S. N. Lahiri (2011). Bootstrapping Lasso estimators. *Journal of the American Statistical Association 106*(494), 608–625.

Chen, P., C. Gao, and A. Y. Zhang (2022). Partial recovery for top-k ranking: Optimality of MLE and SubOptimality of the spectral method. *The Annals of Statistics 50*(3), 1618 – 1652.

Chen, P. and D. Suter (2004). Recovering the missing components in a large noisy low-rank matrix: Application to sfm. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*(8), 1051–1063.

Chen, Y., Y. Chi, J. Fan, C. Ma, and Y. Yan (2020). Noisy matrix completion: Understanding statistical guarantees for convex relaxation via nonconvex optimization. *SIAM Journal on Optimization 30*, 3098–3121.

Chen, Y., J. Fan, C. Ma, and K. Wang (2019). Spectral method and regularized MLE are both optimal for top-K ranking. *The Annals of Statistics 47*(4), 2204 – 2235.

Chen, Y., J. Fan, C. Ma, and Y. Yan (2019). Inference and uncertainty quantification for noisy matrix completion. *Proceedings of the National Academy of Sciences 116*(46), 22931–22937.

Chen, Y. and C. Suh (2015, 07–09 Jul). Spectral MLE: Top-k rank aggregation from pairwise comparisons. In F. Bach and D. Blei (Eds.), *Proceedings of the 32nd International Conference on Machine Learning*, Volume 37 of *Proceedings of Machine Learning Research*, Lille, France, pp. 371–380. PMLR.

Chen, Y. and M. J. Wainwright (2015). Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees.

Chichignoud, M., J. Lederer, and M. J. Wainwright (2016). A practical scheme and fast algorithm to tune the Lasso with optimality guarantees. *Journal of Machine Learning Research 17*(229), 1–20.

Christensen, K. B., S. Kreiner, and M. Mesbah (2013). *Rasch models in health.* John Wiley & Sons.

Cohen, J., T. Chan, T. Jiang, and M. Seburn (2008). Consistent estimation of Rasch item parameters and their standard errors under complex sample designs. *Applied Psychological Measurement 32*(4), 289–310.

Dezeure, R., P. Bühlmann, L. Meier, and N. Meinshausen (2015). High-Dimensional Inference: Confidence Intervals, p-Values and R-Software hdi. *Statistical Science 30*(4), 533 – 558.

Doebler, A., P. Doebler, and H. Holling (2013). Optimal and most exact confidence intervals for person parameters in item response theory models. *Psychometrika 78*(1), 98–115.

Du, W., J. Hannig, T. C. M. Lee, Y. Su, and C. Zhang (2024). AutoGFI: Streamlined generalized fiducial inference for modern inference problems.

Edlefsen, P. T., C. Liu, and A. P. Dempster (2009). Estimating limits from Poisson counting data using Dempster–Shafer analysis. *The Annals of Applied Statistics 3*(2), 764 – 790.

Efron, B. and R. J. Tibshirani (1994). *An introduction to the bootstrap.* CRC press.

Fan, J., S. Guo, and N. Hao (2012). Variance estimation using refitted cross-validation in ultrahigh dimensional regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 74*, 37–65.

Fan, J., J. Hou, and M. Yu (2024). Uncertainty quantification of MLE for entity ranking with covariates.

Fan, J. and R. Li (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association 96*, 1348–1360.

Fei, Z. and Y. Li (2021). Estimation and inference for high dimensional generalized linear models: A splitting and smoothing approach. *The Journal of Machine Learning Research 22*(1), 2681–2712.

Fischer, G. H. and I. W. Molenaar (1995). *Rasch models: foundations, recent developments and applications*. Springer New York.

Fisher, R. A. (1930). Inverse probability. In *Mathematical Proceedings of the Cambridge Philosophical Society*, Volume 26, pp. 528–535. Cambridge University Press.

Fisher, R. A. (1933). The concepts of inverse probability and fiducial probability referring to unknown parameters. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character 139*(838), 343–348.

Fisher, R. A. (1935). The fiducial argument in statistical inference. *Annals of Eugenics 6*(4), 391–398.

Frostig, R., M. J. Johnson, and C. Leary (2018). Compiling machine learning programs via high-level tracing. *Systems for Machine Learning 4*(9).

Gao, C., Y. Shen, and A. Y. Zhang (2023, 01). Uncertainty quantification in the Bradley–Terry–Luce model. *Information and Inference: A Journal of the IMA 12*(2), 1073–1140.

Gross, D. (2011). Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory 57*(3), 1548–1566.

Guhaniyogi, R., S. Qamar, and D. B. Dunson (2017). Bayesian tensor regression. *The Journal of Machine Learning Research 18*(1), 2733–2763.

Guo, W., I. Kotsia, and I. Patras (2011). Tensor learning for regression. *IEEE Transactions on Image Processing 21*(2), 816–827.

Guo, Z., P. Rakshit, D. S. Herman, and J. Chen (2021). Inference for the case probability in high-dimensional logistic regression. *Journal of Machine Learning Research 22*(254), 1–54.

Haldar, J. P. and Z.-P. Liang (2010). Spatiotemporal imaging with partially separable functions: A matrix recovery approach. In *Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on*, pp. 716–719. IEEE.

Hambleton, R. K., H. Swaminathan, and H. J. Rogers (1991). *Fundamentals of item response theory*, Volume 2. Sage.

Han, R., R. Ye, C. Tan, and K. Chen (2020). Asymptotic theory of sparse Bradley–Terry model. *The Annals of Applied Probability 30*(5), 2491 – 2515.

Han, Y. and T. C. M. Lee (2022). Uncertainty quantification for sparse estimation of spectral lines. *IEEE Transactions on Signal Processing 70*, 6243–6256.

Hannig, J. (2009). On generalized fiducial inference. *Statistica Sinica 19*(2), 491–544.

Hannig, J. (2013). Generalized fiducial inference via discretization. *Statistica Sinica 23*, 489–514.

Hannig, J., H. Iyer, R. C. Lai, and T. C. M. Lee (2016). Generalized fiducial inference: A review and new results. *Journal of the American Statistical Association 111*, 1346–1361.

Hannig, J., H. Iyer, and P. Patterson (2006). Fiducial generalized confidence intervals. *Journal of the American Statistical Association 101*, 254–269.

Hannig, J. and T. C. M. Lee (2009). Generalized fiducial inference for wavelet regression. *Biometrika 96*, 847–860.

Hosmer Jr, D. W., S. Lemeshow, and R. X. Sturdivant (2013). *Applied logistic regression.* John Wiley & Sons.

Hunter, D. R. (2004). MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics 32*(1), 384 – 406.

James Leiner, Boyan Duan, L. W. and A. Ramdas (2023). Data fission: splitting a single data point. *Journal of the American Statistical Association 0*(ja), 1–22.

Jang, M., S. Kim, and C. Suh (2018). Top-k rank aggregation from m-wise comparisons. *IEEE Journal of Selected Topics in Signal Processing 12*(5), 989–1004.

Janková, J. and S. van de Geer (2018). Inference in high-dimensional graphical models. In *Handbook of Graphical Models*, pp. 325–350. CRC Press.

Javanmard, A. and A. Montanari (2013). Confidence intervals and hypothesis testing for high-dimensional statistical models. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger (Eds.), *Advances in Neural Information Processing Systems*, Volume 26. Curran Associates, Inc.

Javanmard, A. and A. Montanari (2014). Confidence intervals and hypothesis testing for high-dimensional regression. *Journal of Machine Learning Research 15*, 2869–2909.

Jessica Minnier, L. T. and T. Cai (2011). A perturbation method for inference on regularized regression estimates. *Journal of the American Statistical Association 106*(496), 1371–1382. PMID: 22844171.

Keshavan, R. H., A. Montanari, and S. Oh (2010). Matrix completion from a few entries. *IEEE Transactions on Information Theory 56*(6), 2980–2998.

Klauer, K. C. (1991). Exact and best confidence intervals for the ability parameter of the Rasch model. *Psychometrika 56*, 535–547.

Koh, K., S.-J. Kim, and S. P. Boyd (2007). An interior-point method for large-scale l1-regularized logistic regression. *J. Mach. Learn. Res. 8*, 1519–1555.

Kolda, T. G. and B. W. Bader (2009). Tensor decompositions and applications. *SIAM Review 51*(3), 455–500.

Koltchinskii, V., K. Lounici, and A. B. Tsybakov (2011). Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics 39*(5), 2302–2329.

Lai, R. C., J. Hannig, and T. C. M. Lee (2015). Generalized fiducial inference for ultrahigh-dimensional regression. *Journal of the American Statistical Association 110*, 760–772.

Lee, J. D., D. L. Sun, Y. Sun, and J. E. Taylor (2016). Exact post-selection inference, with application to the Lasso. *The Annals of Statistics 44*, 907–927.

Lee, S.-I., H. Lee, P. Abbeel, and A. Ng (2006). Efficient l1 regularized logistic regression. In *AAAI Conference on Artificial Intelligence*.

Li, T., E. Levina, J. Zhu, et al. (2019). Prediction models for network-linked data. *The Annals of Applied Statistics 13*(1), 132–164.

Li, X., D. Xu, H. Zhou, and L. Li (2018). Tucker tensor regression and neuroimaging analysis. *Statistics in Biosciences 10*(3), 520–545.

Lidong E, J. H. and H. Iyer (2008). Fiducial intervals for variance components in an unbalanced two-component normal mixed linear model. *Journal of the American Statistical Association 103*(482), 854–865.

Liu, H., X. Xu, and J. J. Li (2020). A bootstrap Lasso + partial Ridge method to construct confidence intervals for parameters in high-dimensional sparse linear models. *Statistica Sinica 30*(3), 1333–1355.

Liu, H. and B. Yu (2013). Asymptotic properties of Lasso+mLS and Lasso+Ridge in sparse high-dimensional linear regression. *Electronic Journal of Statistics 7*(none), 3124 – 3169.

Liu, Y., E. X. Fang, and J. Lu (2023). Lagrangian inference for ranking problems. *Operations research 71*(1), 202–223.

Liu, Y. and J. Hannig (2016). Generalized fiducial inference for binary logistic item response models. *Psychometrika 81*, 290–324.

Lockhart, R., J. Taylor, R. J. Tibshirani, and R. Tibshirani (2014). A significance test for the Lasso. *Annals of Statistics 42*(2), 413–468.

Lord, F. M. (1975). Evaluation with artificial data of a procedure for estimating ability and item characteristic curve parameters1. *ETS Research Bulletin Series 1975*(2), i–52.

Luce, R. D. (2005). *Individual choice behavior: A theoretical analysis.* Courier Corporation.

Ma, H., D. Zhou, C. Liu, M. R. Lyu, and I. King (2011). Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 287–296.

Ma, R., T. Tony Cai, and H. Li (2020). Global and simultaneous hypothesis testing for high-dimensional logistic regression models. *Journal of the American Statistical Association 116*(534), 984–998.

Mair, P. and H. Strasser (2018). Large-scale estimation in Rasch models: asymptotic results, approximations of the variance–covariance matrix of item parameters, and divide-and-conquer estimation. *Behaviormetrika 45*(1), 189–209.

Majumder, A. P. and J. Hannig (2016). Higher order asymptotics of generalized fiducial distribution.

Martin, R. and C. Liu (2013). Inferential models: A framework for prior-free posterior probabilistic inference. *Journal of the American Statistical Association 108*(501), 301–313.

Martin, R., J. Zhang, C. Liu, et al. (2010). Dempster–Shafer theory and statistical inference with weak beliefs. *Statistical Science 25*(1), 72–87.

McNally, R. J., H. Iyer, and T. Mathew (2003). Tests for individual and population bioequivalence based on generalized p-values. *Statistics in Medicine 22*(1), 31–53.

Meinshausen, N. and P. Bühlmann (2010). Stability selection. *Journal of the Royal Statistical Society Series B: Statistical Methodology 72*(4), 417–473.

Meinshausen, N., L. Meier, and P. Bühlmann (2009). P-values for high-dimensional regression. *Journal of the American Statistical Association 104*(488), 1671–1681.

Minge Xie, K. S. and W. E. Strawderman (2011). Confidence distributions and a unifying framework for meta-analysis. *Journal of the American Statistical Association 106*(493), 320–333.

Murph, A. C., J. Hannig, and J. P. Williams (2023). Introduction to generalized fiducial inference.

Negahban, S., S. Oh, and D. Shah (2012). Iterative ranking from pair-wise comparisons. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger (Eds.), *Advances in Neural Information Processing Systems*, Volume 25. Curran Associates, Inc.

Negahban, S. and M. J. Wainwright (2012). Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research 13*(1), 1665–1697.

Ning, Y. and H. Liu (2017). A general theory of hypothesis tests and confidence regions for sparse high dimensional models. *The Annals of Statistics 45*(1), 158 – 195.

Ogasawara, H. (2012). Asymptotic expansions for the ability estimator in item response theory. *Computational Statistics 27*(4), 661–683.

Ou-Yang, L., X.-F. Zhang, and H. Yan (2020). Sparse regularized low-rank tensor regression with applications in genomic data analysis. *Pattern Recognition 107*, 107516.

Papadogeorgou, G., Z. Zhang, and D. B. Dunson (2021). Soft tensor regression. *Journal of Machine Learning Research 22*(219), 1–53.

Rao, N., H.-F. Yu, P. K. Ravikumar, and I. S. Dhillon (2015). Collaborative filtering with graph information: Consistency and scalable methods. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems 28*, pp. 2107–2115. Curran Associates, Inc.

Rasch, G. (1993). *Probabilistic models for some intelligence and attainment tests.* ERIC.

Rasines, D. G. and G. A. Young (2022, 12). Splitting strategies for post-selection inference. *Biometrika 110*(3), 597–614.

Rennie, J. D. and N. Srebro (2005). Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 713–719. ACM.

Richardson, M., A. Prakash, and E. Brill (2006). Beyond pagerank: machine learning for static ranking. In *Proceedings of the 15th International Conference on World Wide Web*, New York, NY, USA, pp. 707–715. Association for Computing Machinery.

Robitzsch, A., T. Kiefer, and M. Wu (2024). *TAM: Test Analysis Modules*. R package version 4.2-21.

Salehi, F., E. Abbasi, and B. Hassibi (2019). The impact of regularization on high-dimensional logistic regression. In *Advances in Neural Information Processing Systems*, Volume 32. Curran Associates, Inc.

Shang, F., Y. Liu, and J. Cheng (2014). Generalized higher-order tensor decomposition via parallel ADMM. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Simons, G. and Y.-C. Yao (1999). Asymptotics when the number of parameters tends to infinity in the Bradley-Terry model for paired comparisons. *The Annals of Statistics 27*(3), 1041 – 1060.

Singh, K., M. Xie, and W. E. Strawderman (2007). Confidence distribution (CD): Distribution estimator of a parameter. *Lecture Notes-Monograph Series 54*, 132–150.

Sonderegger, D. L. and J. Hannig (2014). Fiducial theory for free-knot splines. In *Contemporary Developments in Statistical Theory*, pp. 155–189. Springer.

Srebro, N. and T. Jaakkola (2003). Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 720–727.

Su, Y., R. K. W. Wong, and T. C. M. Lee (2020). Network estimation via graphon with node features. *IEEE Transactions on Network Science and Engineering 7*(3), 2078–2089.

Sun, T. and C.-H. Zhang (2012). Scaled sparse linear regression. *Biometrika 99*, 879–898.

Sur, P. and E. J. Candès (2018). A modern maximum-likelihood theory for high-dimensional logistic regression. *Proceedings of the National Academy of Sciences of the United States of America 116*, 14516 – 14525.

Sur, P., Y. Chen, and E. J. Candès (2019). The likelihood ratio test in high-dimensional logistic regression is asymptotically a rescaled chi-square. *Probability theory and related* *fields 175*, 487–558.

T. Tony Cai, Z. G. and R. Ma (2023). Statistical inference for high-dimensional generalized linear models with binary outcomes. *Journal of the American Statistical Association 118*(542), 1319–1332. PMID: 37366472.

Tao, C., T. E. Nichols, X. Hua, C. R. Ching, E. T. Rolls, P. M. Thompson, J. Feng, A. D. N. Initiative, et al. (2017). Generalized reduced rank latent factor regression for high dimensional tensor fields, and neuroimaging-genetic applications. *NeuroImage 144*, 35–57.

Taylor, J. and R. J. Tibshirani (2015). Statistical learning and selective inference. *Proceedings of the National Academy of Sciences 112*(25), 7629–7634.

Tian, X. and J. Taylor (2018). Selective inference with a randomized response. *The Annals of Statistics 46*(2), 679 – 710.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological) 58*, 267–288.

Tibshirani, R. J., A. Rinaldo, R. Tibshirani, and L. Wasserman (2018). Uniform asymptotic inference and the bootstrap after model selection. *The Annals of Statistics 46*(3), 1255 – 1287.

van de Geer, S., P. Bühlmann, Y. Ritov, and R. Dezeure (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics 42*(3), 1166 – 1202.

van de Geer, S. A. (2008). High-dimensional generalized linear models and the Lasso. *Annals of Statistics 36*, 614–645.

Veronese, P. and E. Melilli (2021). Confidence distribution for the ability parameter of the Rasch model. *Psychometrika 86*(1), 131–166.

Wang, F., L. Zhou, L. Tang, and P. X. Song (2021). Method of contraction-expansion (MOCE) for simultaneous inference in linear models. *Journal of Machine Learning Research 22*(192), 1–32.

Warm, T. A. (1989). Weighted likelihood estimation of ability in item response theory. *Psychometrika 54*(3), 427–450.

Wasserman, L. and K. Roeder (2009). High-dimensional variable selection. *The Annals of Statistics 37*(5A), 2178 – 2201.

Wright, B. D. (1998). Estimating measures for extreme scores. *Rasch Measurement Transactions 12*, 632–633.

Xie, M.-g. and K. Singh (2013). Confidence distribution, the frequentist distribution estimator of a parameter: A review. *International Statistical Review 81*(1), 3–39.

Xie, M.-g. and P. Wang (2022). Repro samples method for finite- and large-sample inferences.

Yuchi, H. S., S. Mak, and Y. Xie (2022). Bayesian uncertainty quantification for low-rank matrix completion. *Bayesian Analysis 1*(1), 1–28.

Zabell, S. L. et al. (1992). RA Fisher and fiducial argument. *Statistical Science 7*(3), 369–387.

Zhang, C.-H. and S. S. Zhang (2014). Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76*, 217–242.

Zhang, D., A. Khalili, and M. Asgharian (2022). Post-model-selection inference in linear regression models: An integrated review. *Statistics Surveys 16*(none), 86 – 136.

Zhang, J. and C. Liu (2011). Dempster-Shafer inference with weak beliefs. *Statistica Sinica 21*, 475–494.

Zhang, Y., E. Levina, J. Zhu, et al. (2016). Community detection in networks with node features. *Electronic Journal of Statistics 10*(2), 3153–3178.

Zhao, P. and B. Yu (2006). On model selection consistency of Lasso. *Journal of Machine Learning Research 7*, 2541–2563.

Zhou, H., L. Li, and H. Zhu (2013). Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association 108*(502), 540–552.

Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association 101*(476), 1418–1429.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*(2), 301–320.