Summer 2020

High-Dimensional Inference Based on the Leave-One-Covariate-Out Regularization Path

Xiangyang Cao

High-Dimensional Inference based on the Leave-One-Covariate-Out regularization path

by

Xiangyang Cao

Bachelor of Science
Central University of Finance and Economics, 2016

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Statistics

College of Arts and Sciences

University of South Carolina

2020

Accepted by:

Karl Gregory, Major Professor

Dewei Wang, Major Professor

John Grego, Committee Member

Feifei Xiao, Committee Member

Cheryl L. Addy, Vice Provost and Dean of the Graduate School

Acknowledgments

I am grateful to the Department of Statistics for giving me a chance to be a student here. I want to say thank you for all of my professors and friends at USC. Particularly I want to thank Dr. John Grego and Dr. Feifei Xiao for being my committee members.

Most importantly, I want to express my deepest gratitude to my advisors, Dr. Karl Gregory and Dr. Dewei Wang. I would not be able to complete this dissertation without the guidance and help. I may not be the best student you advised, but I really appreciate your help and understanding. Last but not least, I want to say thank you to my family and my wife, Qianqian, for their endless and unconditional support.

Abstract

The increasingly rapid emergence of high dimensional data, where the number of variables p may be larger than the sample size n, has necessitated the development of new statistical methodologies. LASSO and variants of LASSO are proposed and have been the most popular estimators for the high dimensional regression models. However, not much work has focused on analyzing and summarizing the information contained in the entire solution path of the LASSO. This dissertation consists of three research projects that propose and extend the Leave-One-Covariate-Out(LOCO) solution path statistic to regression and graphical models.

In the first chapter, we propose a new measure of variable importance in high-dimensional regression based on the change in the LASSO solution path when one covariate is left out. For low-dimensional linear models, our method can achieve higher power than the T-test. In the high-dimensional setting, our proposed solution path based test achieves greater power than some other recently developed high-dimensional inference methods.

In the second and third chapter, we extends the LOCO path statistic developed for linear regression with a continuous response to generalized linear models and graphical models. Our procedure allows for the construction of P-values for testing hypothesis about single regression coefficients as well as hypotheses involving multiple regression coefficients and variable screening for graphical models. In the high-dimensional setting, our proposed solution path based test achieves greater power than some other recently developed high-dimensional inference and screening methods.

Table of Contents

List of Tables

List of Figures

Chapter 1

High-Dimensional Inference based on the Leave-One-Covariate-Out LASSO path

1.1 Introduction

We consider the linear regression model

$$Y = \mathbf{X}\beta + \epsilon, \tag{1.1}$$

where $\mathbf{X} = [X_1^T, X_2^T, \ldots, X_n^T]^T$ with $X_i \in \mathbb{R}^p$, $Y \in \mathbb{R}^n$, $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2 \mathbf{I}_n)$, where $\mathbf{I}_n$ is the $n \times n$ identity matrix, and $\beta \in \mathbb{R}^p$ is a vector of unknown regression coefficients. We consider both the cases $p > n$ and $p \leq n$.

We propose a measure of variable importance based on the change in the LASSO solution path due to removing a covariate from the model. Regarding the LASSO solution path

$$\hat{\beta} := \hat{\beta}(\lambda) = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}}(||Y - \mathbf{X}\beta||_2^2 + \lambda||\beta||_1), \quad \lambda > 0 \tag{1.2}$$

as a function of λ taking values in $(0, \infty)$ and returning values $\hat{\beta}(\lambda)$ in $\mathbb{R}^p$, we propose to measure the importance of covariate X_j, for any $j \in \{1, \ldots, p\}$, by comparing the path $\hat{\beta}$ to the path

$$\hat{\beta}^{(-j)} := \hat{\beta}^{(-j)}(\lambda) = \underset{\beta \in \mathbb{R}^p, \beta_j = 0}{\operatorname{argmin}}(||Y - \mathbf{X}\beta||_2^2 + \lambda||\beta||_1), \quad \lambda > 0, \tag{1.3}$$

which is the LASSO solution path when the covariate X_j is removed from the model. Herein, for a vector $\mathbf{v} = (v_1, \ldots, v_K)^T$, $||\mathbf{v}||_2^2 = \sum_{k=1}^K v_k^2$ and $||\mathbf{v}||_1 = \sum_{k=1}^K |v_k|$. We will refer to $\hat{\beta}^{(-j)}$ as the leave-one-covariate-out solution path, or the LOCO path of

the LASSO. It is important to note that for a given j, $\hat{\beta}_j^{(-j)}(\lambda) = 0$ for all λ. We reason that if covariate X_j is important, its importance will be reflected in a large difference between the paths $\hat{\beta}$ and $\hat{\beta}^{(-j)}$, whereas if it is not important, the difference between the paths $\hat{\beta}$ and $\hat{\beta}^{(-j)}$ will be small.

The measure of variable importance we propose, which we shall call the LOCO path statistic, can be used for variable selection and variable screening; moreover, we suggest that it can be used as a test statistic for testing the hypotheses H_0: $\beta_j = 0$ versus H_1: $\beta_j \neq 0$. We also use the LOCO solution path idea to construct a test statistic for testing more complicated hypotheses involving several coefficients, specifically hypotheses of the form

$$H_0\text{: } \beta_j = \beta_{j,0}, \text{ for all } j \in \mathcal{A} \text{ versus } H_1\text{: } \beta_j \neq \beta_{j,0} \text{ for some } j \in \mathcal{A},$$

for some $\{\beta_{j,0}, j \in \mathcal{A}\}$, where $\mathcal{A} \subset \{1, \ldots, p\}$. We propose a bootstrap procedure to calibrate the rejection regions of hypothesis tests based on the LOCO solution path.

We now place our ideas in the literature: the LASSO proposed by Tibshirani, 1996 has been one of the most popular estimators for the linear regression model of (1.1), particularly in the $p > n$ case. It belongs to a class of penalized estimators designed to promote sparsity among the estimated regression coefficients in order to achieve simultaneous variable selection and estimation. Implementing the LASSO requires choosing a value, usually via cross validation, of the tuning parameter λ, which governs the sparsity and shrinkage towards zero of the estimated regression coefficients. Although the LASSO is a powerful tool, the LASSO estimator has a very complicated sampling distribution, so that statistical inference based on LASSO estimators is problematic.

Other estimators for model (1.1) with $p > n$ have been proposed which have, under some conditions, limiting normal distributions, such as the desparsified LASSO estimator, which was proposed by Geer et al., 2014 and Zhang and Zhang, 2014 as well as the estimator introduced by Javanmard and Montanari, 2014; these methods

enable inference, but a downside is that they require the choice of an additional tuning parameter and inferences may be very sensitive to the choice of tuning parameter. The adaptive LASSO estimator proposed by Zou, 2006, under some conditions and with tuning parameters appropriately chosen, has a limiting normal distribution (for non-zero coefficients), though convergence seems to be slow; a bootstrap procedure has been shown to be consistent for the adaptive LASSO in Das et al., 2019. A bootstrap method for the LASSO is proposed in Chatterjee and Lahiri, 2011and Chatterjee, Lahiri, et al., 2013, which is consistent for a modified LASSO and adaptive LASSO estimator. A sequential significance testing procedure for variables entering the model along the LASSO solution path was proposed in Lockhart et al., 2014. Inferential methods for the high-dimensional linear model based on sample splitting, for example in Wasserman and Roeder, 2009 and Meinshausen et al., 2009, have also been proposed and implemented with success.

As variable selection methods, sure independence screening (SIS) and iterative sure independence screening (ISIS) are proposed by Fan and Lv, 2008 for ultra-high dimensional linear regression. Ultra-high dimensional regression focuses on the settings with $\log(p) = O(n^{\varsigma})$. It has been extended to GLM Fan, Song, et al., 2010, GAM Fan et al., 2011 and multivariate regression models Ke et al., 2014. Although these methods enjoy the sure screening property Fan and Lv, 2008, SIS only considers the marginal contribution of each variable to the response.

To our knowledge, however, not much work has focused on analyzing and summarizing the information contained in the entire solution path of the LASSO with respect to the importance of each variable. We propose to consider the LASSO solution path in its entirety, and then measure how it changes when we leave one covariate out.

The idea of leave-one-covariate-out (LOCO) inference is not new. The following LOCO-based procedure for measuring variable importance is described in Lei et al.,

2018: Let $\hat{\mu}$ be an estimate of $E(Y|\mathbf{X})$ based on some training data $(\mathbf{X}, Y)$, and let $\hat{\mu}_{(-j)}$ be the same estimator based on the training data $(\mathbf{X}_{(-j)}, Y)$, where $\mathbf{X}_{(-j)}$ is the matrix $\mathbf{X}$ with column j removed. Then we measure the excess prediction error on new data (X_{new}, Y_{new}) as

$$|Y_{new} - \hat{\mu}_{(-j)}(X_{new})| - |Y_{new} - \hat{\mu}(X_{new})|,$$

where the "new" data can come from crossvalidation testing sets or from a separate testing data set. The larger the above quantity, the greater importance we assign to covariate X_j, as it measures how much worse our predictions become due to removing covariate X_j.

Permutation feature importance, introduced by Breiman, 2001 and generalized by Fisher et al., 2018, is a similar to the LOCO approach to measuring variable importance; instead of removing covariate X_j from the model, the observed values of covariate X_j are randomly permuted. By this permutation, the association between covariate X_j and the response is broken and the resulting model is different from the one fit to the original data.

What we propose falls into the framework of LOCO variable importance and inference; however, rather than measuring the change in the prediction error due to removing a covariate, we consider the change in the LASSO solution path.

This paper is organized as follows: Section 1.2 defines our measure of variable importance based on the change in the LASSO solution path due to the removal of a covariate and discusses its use as a variable selection and variable screening tool. Section 1.3 explains how we propose to use the LOCO solution path idea to construct test statistics for testing hypotheses about the regression coefficients. We also describe a bootstrap procedure for estimating the null distribution of our LOCO path-based test statistics. Section 1.4 presents simulation results and Section 1.5 illustrates the method on a real data set. Section 1.6 provides additional discussion.

1.2 The leave-one-covariate-out path statistic

To formulate our metric for the difference between the LASSO solution path $\hat{\beta}$ defined in (2.2) and the LOCO solution path of the LASSO defined in (2.3), we define a quantity for functions taking values in $(0, \infty)$ and returning values in $\mathbb{R}^p$. Firstly, for any function g taking values in $(0, \infty)$ and returning values in $\mathbb{R}$, let

$$\|g\|_s = \begin{cases} (\int_0^\infty |g(\lambda)|^s d\lambda)^{1/s}, & 0 < s < \infty \\ \sup_{\lambda > 0} |g(\lambda)|, & s = \infty. \end{cases}$$

Secondly, for a vector $x \in \mathbb{R}^p$, let

$$||x||_t = \begin{cases} (\sum_{j=1}^p |x_j|^t)^{1/t}, & 0 < t < \infty \\ \max_{1 \le j \le p} |x_j|, & t = \infty. \end{cases}$$

Now, for a function f taking values in $(0, \infty)$ and returning values in $\mathbb{R}^p$ such that $f(\lambda) = (f_1(\lambda), \ldots, f_p(\lambda))^T$, define the quantity $\|f\|_{s,t}$ as

$$\|f\|_{s,t} = \|(\|f_1\|_s, \ldots, \|f_p\|_s)^T\|_t.$$

Having defined a quantity for functions taking values in $(0, \infty)$ and returning values in $\mathbb{R}^p$, we define the LOCO path statistic for covariate X_j as

$$T_j(s, t) = ||\hat{\beta} - \hat{\beta}^{(-j)}||_{s,t},$$

which measures the change in the LASSO solution path due to removing covariate X_j from the model.

In practice, it is convenient to use $s = t$; if $s = t = q$, we have

$$T_j(q, q) = \begin{cases} \left(\sum_{k=1}^p \int_0^\infty |\hat{\beta}_k(\lambda) - \hat{\beta}_k^{(-j)}(\lambda)|^q d\lambda \right)^{\frac{1}{q}} & q < \infty \\ \max_{1 \le k \le p} \sup_{\lambda > 0} |\hat{\beta}_k(\lambda) - \hat{\beta}_k^{(-j)}(\lambda)| & q = \infty. \end{cases}$$

We recommend using $q = 1$ or $q = 2$ in practice. We have found that under $q = \infty$ our hypothesis test tend to have lower power, so we do not recommend this setting. We illustrate this in the simulation section.

We posit that the quantity $T_j(s,t)$ will be large if $\beta_j \neq 0$ and small if $\beta_j = 0$, for $j = 1, \ldots, p$, so that $T_j(s,t)$ may serve as a measure of variable importance for covariate X_j. Since the LASSO solution path is piecewise linear, we can calculate $T_j(s,t)$ exactly. More details about the calculation can be found in the section S.1 of the Supplementary Material.

The LOCO path statistic as a measure of variable importance

For the sake of illustration, let us consider one special case of $T_j(s,t)$, with $s = t = 1$. We have

$$T_j(1,1) = ||\hat{\beta} - \hat{\beta}^{(-j)}||_{1,1} = \sum_{k=1}^{p} \int_0^\infty |\hat{\beta}_k(\lambda) - \hat{\beta}_k^{(-j)}(\lambda)| d\lambda,$$

which is equal to the sum of all the areas under the curves $|\hat{\beta}_k(\cdot) - \hat{\beta}_k^{(-j)}(\cdot)|$, $k = 1, \ldots, p$. We depict this for the following simple example: We generate one dataset from the linear regression model (1.1) with $n = 100$, $p = 4$ and $\beta = (1, 1, 0, 0)^T$, and compute the test statistics $T_1(1,1)$ and $T_3(1,1)$. The left and right panels of Figure 1.1 show the original LASSO solution path as well as the solution path after removing the first and third covariates, respectively, from the model. In each panel, the sum of the areas of the shaded regions is the value of the test statistic.

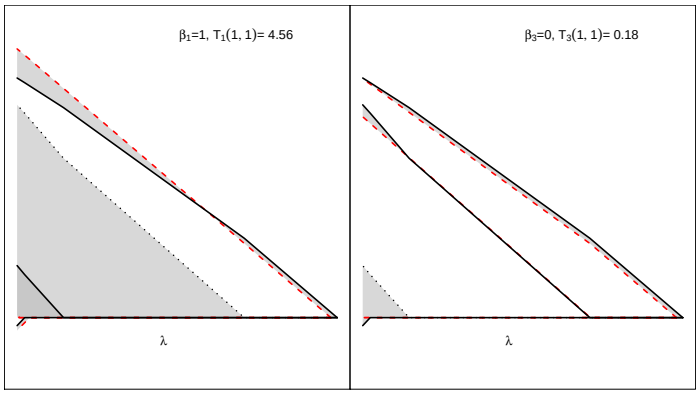

Figure 1.1: Shaded areas show how $T_j(1,1)$ measures the change in LASSO path. Black solid line depicts the solution path before removal. Black dotted line depicts the solution path of the covariates being removed. Red dashed line depicts the solution path after removal. Left: $T_1(1,1)$. Right: $T_3(1,1)$.

We propose to summarize the importance of the variables measured by the LOCO path statistic in the following way. After standardizing the values of $T_j(s,t)$, $j = 1,\ldots,p$, so that they sum to one, for example by defining

$$\overline{T}_j(s,t) = T_j(s,t)\left(\sum_{k=1}^{p} T_k(s,t)\right)^{-1}, \quad j = 1,\ldots,p,$$

we can make a plot such as the one in Figure 1.2, which shows the values of $\overline{T}_1(1,1),\ldots,\overline{T}_{12}(1,1)$, expressed as percentages. This is based on a single dataset simulated from (1.1) with $n = 100$, $p = 12$, $\beta = (1,1,1,0,\ldots,0)^T$, for the sake of illustration. The first three covariates are seen to have the highest importance according to the LOCO path statistic.

Furthermore, we consider attaching to the variable importance a measure of uncertainty. The LOCO path $\hat{\beta}_k^{(-j)}(\lambda)$ could be fitted by permuting variable j in $\mathbf{X}$. By permuting variable j in $\mathbf{X}$, we break the association between X_j and Y, which has an effect similar to removing variable j. By permuting the observed values of covariate j multiple times we can obtain an interval for the variable importance. Figure 1.2 also shows the permutation interval calculated for the importance measure of each variable.

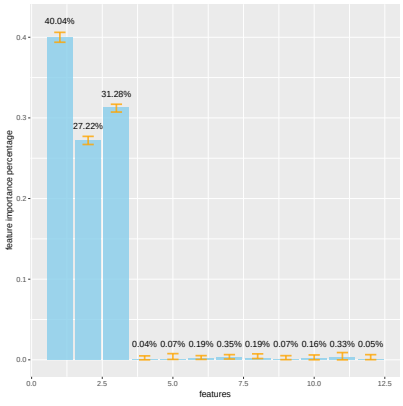

Figure 1.2: Variable importance for all variables based on the LOCO path statistic. The error bar is our permutation interval. The variable importance is also shown as percentages on top of the error bar.

The so-called ultra-high dimensional setting was discussed in Fan and Lv, 2008, where the dimensionality p grows exponentially ($\log(p) = O(n^{\zeta})$) as n grows. For ultra-high dimensional problems, preliminary variable screening is often done to reduce the dimension of the data.

Our method naturally adapts to ultra-high dimensional settings. By calculating how the removal of each variable will alter the LASSO solution path, we have a simple way to screen out variables which are likely to be irrelevant. Our method uses the information contained in the LASSO solution path, which utilizes both joint and marginal information. One interesting result of LASSO in the high-dimensional setting is that some variables never enter the model. If we take a closer look at the solution path of such variables, they are equal to 0 for all values of λ. If we were to use cross validation to select the LASSO tuning parameter and obtain the final selection results, these variables would never be selected. This means we can safely screen out these variables at the beginning.

Based on this intuition, we suggest the following screening procedure: Compute the solution path with all variables in the model. Then remove one variable at a time and compute the LOCO solution path; compute the values $T_1(s,t), \ldots, T_p(s,t)$, which compare the solution path based on the full set of covariates to the LOCO solution paths. Then screen out variables for which $T_j(s,t) \leq \epsilon$, where ϵ is a user-specified threshold. Choosing $\epsilon = 0$ discards only those variables which never enter the solution path. We can also rank $T_j(s,t)$ and only select the top K variables, where we might choose K to be $n-1$ and n is the sample size.

1.3 Hypothesis testing using the LOCO path idea

We now consider using the LOCO path idea to test hypotheses of the form

$$H_0\colon\ \beta_j = \beta_{j,0} \text{ for all } j \in \mathcal{A} \text{ versus } H_1\colon\ \beta_j \neq \beta_{j,0} \text{ for some } j \in \mathcal{A}, \tag{1.4}$$

for some $\{\beta_{j,0}, j \in \mathcal{A}\}$, where $\mathcal{A} \subset \{1, \ldots, p\}$. We first calculate the LASSO solution path with all variables included. Next, we compute the solution path subject to the constraint specified by the null hypothesis, which is given by

$$\hat{\beta}_0 := \hat{\beta}_0(\lambda) = \underset{\beta \in \mathbb{R}^p, \beta_j = 0 \in \mathcal{A}}{\operatorname{argmin}} (||(Y - \mathbf{X}_{\mathcal{A}}\beta_{0,\mathcal{A}}) - \mathbf{X}\beta||_2^2 + \lambda||\beta||_1), \tag{1.5}$$

where $\beta_{0,\mathcal{A}} = (\beta_{j,0}, j \in \mathcal{A})^T$ and $\mathbf{X}_{\mathcal{A}}$ is the matrix constructed out of the columns of $\mathbf{X}$ with indices in $\mathcal{A}$.

We then suggest as a test statistic for testing H_0 versus H_1 the quantity

$$T_0(s,t) = \|\hat{\beta} - \hat{\beta}_0\|_{s,t}, \tag{1.6}$$

which compares the solution paths $\hat{\beta}_0$ and $\hat{\beta}$. For testing the hypotheses

$$H_0\colon\ \beta_j = 0 \text{ versus } H_1\colon\ \beta_j \neq 0,$$

for some $j \in \{1, \ldots, p\}$, we have $\hat{\beta}_0 = \hat{\beta}^{(-j)}$, so that the test statistic $T_0(s,t)$ is equal to the LOCO path variable importance statistic $T_j(s,t) = \|\hat{\beta} - \hat{\beta}^{(-j)}\|_{s,t}$.

A bootstrap estimator of the null distribution

In order to test the hypotheses in (2.5) using the test statistic $T_0(s,t)$ in (2.7), we need to know the distribution of $T_0(s,t)$ under H_0. We propose estimating this null distribution using a residual bootstrap procedure.

In order to obtain residuals from which to resample, we propose obtaining an initial estimator $\tilde{\beta}$, which we will discuss at the end of this section, of the vector β from which we can obtain residuals

$$\tilde{\epsilon} = Y - \mathbf{X}\tilde{\beta}.$$

Let $\tilde{Y}^*$ be the $n \times 1$ random vector with entries given by $\tilde{Y}_i^* = X_i^T \tilde{\beta} + \tilde{\epsilon}_i^*$, for $i = 1, \ldots, n$, where $\epsilon_1^*, \ldots, \epsilon_n^*$ are sampled with replacement from the entries of the residual vector $\tilde{\epsilon} = (\tilde{\epsilon}_1, \ldots, \tilde{\epsilon}_n)^T$.

For testing the hypotheses in (2.5), the bootstrap versions $\hat{\beta}^*$ and $\hat{\beta}_0^*$ of $\hat{\beta}$ and $\hat{\beta}_0$ are constructed as

$$\hat{\beta}^* := \hat{\beta}^*(\lambda) = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}}(||(\tilde{Y}^* - \mathbf{X}_{\mathcal{A}}(\tilde{\beta}_{\mathcal{A}} + \beta_{0,\mathcal{A}})) - \mathbf{X}\beta||_2^2 + \lambda||\beta||_1) \tag{1.7}$$

and

$$\hat{\beta}_0^* := \hat{\beta}_0^*(\lambda) = \underset{\beta \in \mathbb{R}^p, \beta_j = 0, j \in \mathcal{A}}{\operatorname{argmin}}(||(\tilde{Y}^* - \mathbf{X}_{\mathcal{A}}(\tilde{\beta}_{\mathcal{A}} + \beta_{0,\mathcal{A}})) - \mathbf{X}\beta||_2^2 + \lambda||\beta||_1), \tag{1.8}$$

respectively. Then the bootstrap version of $T_0(s,t) = \|\hat{\beta} - \hat{\beta}_0\|_{s,t}$ is given by

$$T_0^*(s,t) = \|\hat{\beta}^* - \hat{\beta}_0^*\|_{s,t}.$$

Given a large number B of Monte-Carlo replicates of $T_0^*(s,t)$, denoted by, say, $T_0^{*,(1)}(s,t) < \cdots < T_0^{*,(B)}(s,t)$, when ordered, our bootstrap-based test of H_0 at significance level α has decision rule

$$\text{Reject } H_0 \text{ if and only if } T_0(s,t) > T_0^{*,(\lfloor B(1-\alpha) \rfloor)},$$

where $T_0^{*,(\lfloor B(1-\alpha) \rfloor)}$ is the Monte-Carlo approximation to the bootstrap estimator of the upper α-quantile of the null distribution of $T_0(s,t)$, and $\lfloor \cdot \rfloor$ is the floor function. We could also obtain a bootstrapped P-value by

$$B^{-1} \sum_{i=1}^{B} I\{T_0^{*,(i)}(s,t) > T_0(s,t)\},$$

where $I(\cdot)$ is the indicator function.

For the simpler hypotheses H_0: $\beta_j = 0$ versus H_1: $\beta_j \neq 0$ for any $j = 1, \ldots, p$, we need to construct a bootstrap version of the LOCO path statistic $T_j(s,t) = \|\hat{\beta} - \hat{\beta}^{(-j)}\|_{s,t}$.

The bootstrap versions of $\hat{\beta}$ and $\hat{\beta}^{(-j)}$, following (2.9) and (2.10), are

$$\hat{\beta}^* := \hat{\beta}^*(\lambda) = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}}(||(\tilde{Y}^* - \mathbf{X}_j\tilde{\beta}_j) - \mathbf{X}\beta||_2^2 + \lambda||\beta||_1)$$

and

$$\hat{\beta}^{*(-j)} := \hat{\beta}^{*(-j)}(\lambda) = \underset{\beta \in \mathbb{R}^p, \beta_j = 0}{\operatorname{argmin}}(||(\tilde{Y}^* - \mathbf{X}_j\tilde{\beta}_j) - \mathbf{X}\beta||_2^2 + \lambda||\beta||_1),$$

respectively, where $\mathbf{X}_j$ is column j of the matrix $\mathbf{X}$. Then the bootstrap version of $T_j(s,t)$ is given by

$$T_j^*(s,t) = \|\hat{\beta}^* - \hat{\beta}^{*(-j)}\|_{s,t}.$$

Regarding the choice of the initial estimator $\tilde{\beta}$ of β, which is used only to obtain residuals suitable for resampling, we suggest, when $p \geq n$, the adaptive LASSO estimator

$$\hat{\beta}^{Ada} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}}(||Y - \mathbf{X}\beta||_2^2 + \gamma \sum_{j=1}^{p} \hat{w}_j|\beta_j|),$$

where the tuning parameter γ is selected via 10-fold cross validation and the weights $\hat{w}_1, \ldots, \hat{w}_p$ are given by

$$\hat{w}_j = 1/|\hat{\beta}_j^L|, \quad j = 1, \ldots, p,$$

where $\hat{\beta}_1^L, \ldots, \hat{\beta}_p^L$ are the LASSO estimates of $\beta_1, \ldots, \beta_p$ from (2.2) under the 10-fold cross validation choice of λ. This is the initial estimator we have used in our simulation studies, and it appears to work well. For the $p < n$ case the least-squares estimator could be used, though even in the low-dimensional case, we still recommend using the adaptive LASSO estimator when p is close to n.

Justification of the bootstrap for a simple case

Finding the sampling distribution of $T_0(s,t)$ in general is a very hard problem which we do not attempt to solve. However, we do provide in this section an argument for why the bootstrap method described in the previous section will work in a simple case: the low-dimensional case, with $p < n$, with a design matrix having orthonormal

columns. We focus on the null distribution of the test statistic $T_j(1,1) = \|\hat{\beta} - \hat{\beta}^{(-j)}\|_{1,1}$ for testing H_0: $\beta_j = 0$ versus H_1: $\beta_j \neq 0$ for some $j \in \{1, \ldots, p\}$.

In low-dimension, if the design matrix $\mathbf{X}$ satisfies $\mathbf{X}^T\mathbf{X} = \mathbf{I}_p$, where $\mathbf{I}_n$ is the $n \times n$ identity matrix, the LASSO solution path $\hat{\beta}$ has entries given by

$$\hat{\beta}_k(\lambda) = S_\lambda(\hat{\beta}_k^{\mathrm{LS}}), \quad k = 1, \ldots, p,$$

where $\hat{\beta}^{\mathrm{LS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y = \mathbf{X}^T Y$ is the least-squares estimator of β and $S_\lambda(\cdot)$ is the soft-thresholding operator defined by

$$S_\lambda(x) = \begin{cases} x - \lambda, & x > \lambda \\ 0, & -\lambda < x < \lambda \\ x + \lambda, & x < -\lambda \end{cases}$$

for $\lambda \geq 0$. The solution path $\hat{\beta}^{(-j)}$ has entries given by

$$\hat{\beta}_k^{(-j)}(\lambda) = \begin{cases} 0 & k = j \\ S_\lambda(\hat{\beta}_k^{\mathrm{LS}}) & k \neq j \end{cases}$$

for $k = 1, \ldots, p$.

In this case, the LOCO path statistic $T_j(1,1)$ is given by

$$\begin{aligned} T_j(1,1) = \|\hat{\beta} - \hat{\beta}^{(-j)}\|_{1,1} &= \sum_{k=1}^{p} \int_0^\infty |\hat{\beta}_k(\lambda) - \hat{\beta}_k^{(-j)}(\lambda)| d\lambda \\ &= \int_0^{|\hat{\beta}_j^{\mathrm{LS}}|} (|\hat{\beta}_j^{\mathrm{LS}}| - \lambda) d\lambda = \frac{1}{2}|\hat{\beta}_j^{\mathrm{LS}}|^2. \end{aligned}$$

So, our test statistic is merely a 1-to-1 mapping of the least-squares estimator. Hence, under H_0: $\beta_j = 0$,

$$nT_j(1,1) = \frac{n}{2}|\hat{\beta}_j^{\mathrm{LS}}|^2 \sim W\frac{\sigma^2}{2},$$

where $W \sim \chi_1^2$.

Now consider the bootstrap version $T_j^*(1,1)$ of $T_j(1,1)$ in the $p < n$ and orthonormal design case; we assume that the least-squares estimator is used as the initial

estimator from which the residuals are obtained. Let $\hat{\beta}^{*,\mathrm{LS}} = \mathbf{X}^T\tilde{Y}^*$ be the bootstrap version of $\hat{\beta}^{\mathrm{LS}}$. Now, we can write the entries of

$$\hat{\beta}^*(\lambda) = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}}(||(\tilde{Y}^* - \mathbf{X}_j\hat{\beta}_j^{\mathrm{LS}}) - \mathbf{X}\beta||_2^2 + \lambda||\beta||_1)$$

as

$$\hat{\beta}_k^*(\lambda) = \begin{cases} S_\lambda(\hat{\beta}_k^{*,\mathrm{LS}} - \hat{\beta}_k^{\mathrm{LS}}), & k = j \\ S_\lambda(\hat{\beta}_k^{*,\mathrm{LS}}), & k \neq j \end{cases} \quad \text{for } k = 1, \ldots, p,$$

using the fact that

$$\mathbf{X}_k^T(\tilde{Y}^* - \mathbf{X}_j\hat{\beta}_j^{\mathrm{LS}}) = \begin{cases} \hat{\beta}_k^{*,\mathrm{LS}} - \hat{\beta}_k^{\mathrm{LS}}, & k = j \\ \hat{\beta}_k^{*,\mathrm{LS}} & k \neq j, \end{cases} \quad \text{for } k = 1, \ldots, p.$$

In addition, we can write the entries of

$$\hat{\beta}^{*(-j)}(\lambda) = \underset{\beta\in\mathbb{R}^p,\beta_j=0}{\operatorname{argmin}}\ (||(\tilde{Y}^* - \mathbf{X}_j\hat{\beta}_j^{\mathrm{LS}}) - \mathbf{X}\beta||_2^2 + \lambda||\beta||_1)$$

as

$$\hat{\beta}_k^{*(-j)}(\lambda) = \begin{cases} 0, & k = j \\ S_\lambda(\hat{\beta}_k^{*,\mathrm{LS}}), & k \neq j \end{cases} \quad \text{for } k = 1, \ldots, p.$$

So we have

$$\begin{aligned} T_j^*(1,1) = \|\hat{\beta}^* - \hat{\beta}^{*(-j)}\|_{1,1} &= \sum_{k=1}^{p}\int_0^\infty |\hat{\beta}_k^{*(-j)}(\lambda) - \hat{\beta}_k^*(\lambda)|d\lambda \\ &= \int_0^{|\hat{\beta}_k^{*,\mathrm{LS}} - \hat{\beta}_k^{\mathrm{LS}}|}(|\hat{\beta}_k^{*,\mathrm{LS}} - \hat{\beta}_k^{\mathrm{LS}}| - \lambda)d\lambda = \frac{1}{2}|\hat{\beta}_j^{*,\mathrm{LS}} - \hat{\beta}_j^{\mathrm{LS}}|^2. \end{aligned}$$

It can be established that

$$\sup_{x\in\mathbb{R}}\left|P_*\left(\frac{n}{2}|\hat{\beta}_j^{*,\mathrm{LS}} - \hat{\beta}_j^{\mathrm{LS}}|^2 < x\right) - P\left(\frac{n}{2}|\hat{\beta}_j^{\mathrm{LS}} - \beta_j|^2 < x\right)\right| \xrightarrow{p} 0,$$

as $n \to \infty$, where P_* denotes probability conditional on the observed data Mammen, 2012. This means our bootstrap works in the low-dimensional orthonormal design case. In the high-dimensional case, or even in the low-dimensional case without the assumption of an orthogonal design, (2.2) does not admit a simple solution, and

in this setting the derivation of the distribution of the test statistic would be very difficult. Our simulation studies, however, suggest that our bootstrap procedure can consistently estimate the null distributions of the test statistics in the non-orthogonal design and high-dimensional cases.

1.4 Simulation studies

We now study via simulation the effectiveness of the LOCO path statistic as a variable screening tool as well as the properties of our proposed LOCO-path-based tests of hypotheses which use the residual bootstrap to estimate the null distributions of the test statistics. An R package `LOCOpath` that implements all of our proposed methods is publicly available at `http://github.com/devcao/LOCOpath`. We first present the variable screening results.

Variable screening

To assess the performance of the LOCO-path-based variable screening procedure described in Section 1.2, we follow the simulation examples in Fan and Lv, 2008, generating data from the model

$$Y = \beta X_1 + \beta X_2 + \beta X_3 + \epsilon,$$

where $\epsilon \sim \mathcal{N}(0, 1)$, with a total of p predictors $X_1, \ldots, X_p$ in the model. The rows of the design matrix are generated as independent multivariate normal random vectors with covariance matrix $\Sigma = (\rho^{|i-j|})_{1 \leq i,j \leq p}$, where $\rho = 0, 0.1, 0.5$ and 0.9. Models with $\beta = 1, 2, 3$, $p = 100$, $n = 20$, and $p = 1000$, $n = 50$ are considered. We simulated 200 data sets for each model. To compare with SIS and ISIS, we utilized the R package `SIS` Saldana and Feng, 2018. We simulated 200 data sets and for each model we calculate $T_j(1, 1)$ and $T_j(2, 2)$ for $j = 1, 2, \ldots, p$ and select the top $n - 1$ covariates, selecting the same number of covariates with SIS and ISIS in order to make a fair

comparison. For our method, we utilized the R package `lars` Hastie and Efron, 2013 with LASSO modification to calculate our test statistic.

In Table 2.1 we show the proportion of times that the true model is contained in the set of selected covariates for our method and for the SIS and ISIS variable screening methods. In most cases, the model selected by our LOCO-path-based method contains the true model with greater frequency than that of the SIS and ISIS methods. We note that our method achieves this without any need for selecting tuning parameters, whereas the ISIS methods involves iterated LASSO fits for which the strength of the sparsity penalty must be chosen.

Table 1.1: Proportion of times SIS, ISIS and our method selected a set of covariates containing $\{X_1, X_2, X_3\}$.

Setting	β	$T_j(1,1)$	$T_j(2,2)$	SIS	ISIS
$p = 1000, n = 50, \Sigma = \mathbf{I}_p$	1	0.995	0.995	0.900	0.945
	2	1.000	1.000	0.945	1.000
	3	1.000	1.000	0.990	1.000
$p = 1000, n = 50, \Sigma = (0.1^{\lvert i-j \rvert})_{1 \le i,j \le p}$	1	0.990	0.990	0.960	0.960
	2	1.000	1.000	0.995	1.000
	3	1.000	1.000	0.990	1.000
$p = 1000, n = 50, \Sigma = (0.5^{\lvert i-j \rvert})_{1 \le i,j \le p}$	1	1.000	1.000	1.000	0.890
	2	1.000	1.000	1.000	1.000
	3	1.000	1.000	1.000	1.000
$p = 1000, n = 50, \Sigma = (0.9^{\lvert i-j \rvert})_{1 \le i,j \le p}$	1	0.980	0.975	1.000	0.535
	2	1.000	1.000	1.000	0.825
	3	1.000	1.000	1.000	0.965
$p = 100, n = 20, \Sigma = \mathbf{I}_p$	1	0.630	0.630	0.560	0.440
	2	0.915	0.920	0.700	0.860
	3	0.955	0.955	0.710	0.905
$p = 100, n = 20, \Sigma = (0.1^{\lvert i-j \rvert})_{1 \le i,j \le p}$	1	0.705	0.700	0.685	0.495
	2	0.960	0.965	0.810	0.890
	3	0.970	0.970	0.845	0.970
$p = 100, n = 20, \Sigma = (0.5^{\lvert i-j \rvert})_{1 \le i,j \le p}$	1	0.940	0.940	0.945	0.505
	2	1.000	1.000	0.990	0.940
	3	1.000	1.000	0.995	0.975
$p = 100, n = 20, \Sigma = (0.9^{\lvert i-j \rvert})_{1 \le i,j \le p}$	1	0.745	0.740	1.000	0.465
	2	0.995	0.995	1.000	0.635
	3	1.000	1.000	1.000	0.805

Study of power and size of LOCO path tests of hypotheses

Test involving a single coefficient

We first study the size and power of the LOCO path test for testing the hypotheses H_0: $\beta_j = 0$ versus H_1: $\beta_j \neq 0$ for some $j \in \{1, \ldots, p\}$, where the rejection region of the test is calibrated using the residual bootstrap procedure described in Section 1.3. We consider the test statistics $T_j(1,1)$, $T_j(2,2)$, and $T_j(\infty, \infty)$.

In high-dimensional ($p \geq n$) settings, we compare the empirical size and power of our test based on these statistics with the test based on the desparsified LASSO estimator of Geer et al., 2014. We use the R package `hdi` Dezeure et al., 2015a to obtain the P-value based on the desparsified LASSO estimator using default settings Dezeure et al., 2015b. And we utilize the R package `lars` Hastie and Efron, 2013 with lasso modification to implement our method. In low-dimensional ($p < n$) settings, we compare the performance of our tests to that of the classical t-test.

We generate data according to the model

$$Y = \mathbf{X}\beta + \epsilon,$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I}_n)$ and consider three cases with $n = 100$, $p = 80$ and $p = 1000$. For $p = 1000$, we set $\beta = (\beta_1, \ldots, \beta_p)^T$ such that $\beta_2 = \cdots = \beta_{10} = 1$, $\beta_{11} = \cdots = \beta_{1000} = 0$. For $p = 80$, we set $\beta_2 = \beta_3 = 1$, $\beta_4 = \cdots = \beta_{80} = 0$.

To simulate the power curve, we take different values of $\beta_1 \in \{0/10, 1/10, \ldots, 1\}$. Each row of $\mathbf{X}$ is generated independently from the multivariate normal distribution $\mathcal{N}(0, \Sigma)$, where we consider different choices of the $p \times p$ covariance matrix Σ.

For each choice of Σ and for each value of $\beta_1 \in \{0/10, 1/10, \ldots, 1\}$, we generate $N = 500$ data sets and with each data set we test H_0: $\beta_1 = 0$ versus H_1: $\beta_1 \neq 0$. For each data set, we draw $B = 500$ bootstrap samples to estimate the null distribution. We record the proportion of rejections of H_0 at the $\alpha = 0.05$ significance level.

The empirical size of the simulation for H_0: $\beta_1 = 0$ under $p = 1000$, is given in Table 1.2 under different choices of Σ. We also recorded the empirical size of the test based on the desparsified LASSO estimator.

It is clear that our method nicely controlled the size under different choices of Σ and different quantities $T_1(1,1)$, $T_1(2,2)$ and $T_1(\infty,\infty)$. The desparsified LASSO does not control the size in many cases.

The empirical power curves of our test based on the LOCO path statistics $T_1(1,1)$ and $T_1(\infty,\infty)$ as well as of the test based on the desparsified LASSO under settings $n = 1000$ and $p = 80$ over the values $\beta_1 \in \{0/10, 1/10, \ldots, 1\}$ are depicted in Figures 1.3 and 1.4.

For most cases, $T_1(1,1)$ have the highest power, while $T_1(\infty,\infty)$ loses a lot of power under the correlated design. Under different designs, our method outperformed desparsified LASSO using quantity $T_1(1,1)$.

It is interesting to see that the desparsified LASSO appears to outperform our method under the design $\Sigma = (0.9^{|i-j|})_{1\leq i,j\leq p}$.

However, since its size is inflated in that case, we dismiss its power curve. Overall, our methods achieves comparable or higher power, with size well-controlled, compared to the desparsified LASSO method.

For the $p = 80$ case, we will compare our method to the classical t-test. From the power curve in Figure 1.4, it is clear that our method achieved considerably greater power than the t-test using both $T_1(1,1)$ and $T_1(\infty,\infty)$, while controlling the size at the same time.

Table 1.2: Empirical size of the test under different Σ with $n = 100$, $p = 1000$.

Design	Method	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
$\Sigma = \mathbf{I}_p$	$T_1(1,1)$	0.194	0.106	0.048	0.008
	$T_1(2,2)$	0.186	0.110	0.056	0.016
	$T_1(\infty,\infty)$	0.230	0.140	0.078	0.012
	Desparsified	0.138	0.058	0.030	0.010
$\Sigma = (0.5^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.226	0.110	0.054	0.018
	$T_1(2,2)$	0.192	0.084	0.040	0.004
	$T_1(\infty,\infty)$	0.196	0.090	0.042	0.008
	Desparsified	0.222	0.138	0.084	0.020
$\Sigma = (0.9^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.214	0.116	0.086	0.024
	$T_1(2,2)$	0.238	0.124	0.076	0.030
	$T_1(\infty,\infty)$	0.264	0.160	0.086	0.018
	Desparsified	0.274	0.162	0.102	0.054
$\Sigma = (0.5^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.194	0.126	0.064	0.018
	$T_1(2,2)$	0.212	0.098	0.050	0.008
	$T_1(\infty,\infty)$	0.180	0.102	0.050	0.014
	Desparsified	0.126	0.048	0.028	0.004
$\Sigma = (0.8^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.242	0.116	0.056	0.010
	$T_1(2,2)$	0.182	0.086	0.040	0.010
	$T_1(\infty,\infty)$	0.198	0.084	0.050	0.008
	Desparsified	0.070	0.022	0.010	0.002

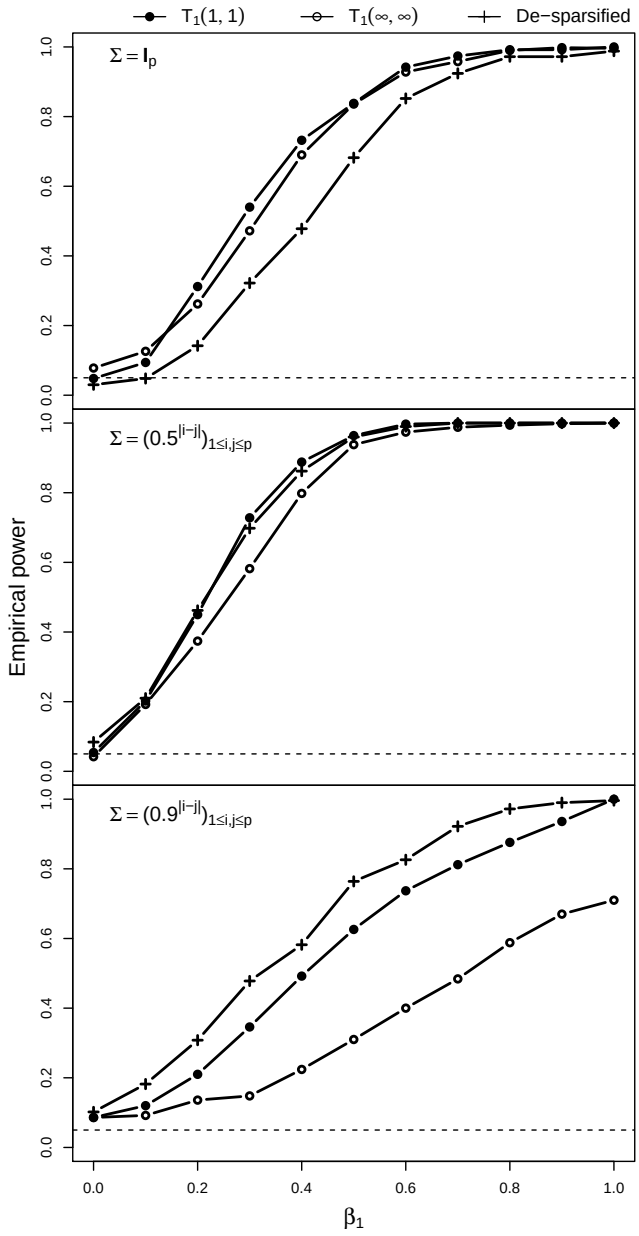

Figure 1.3: Empirical power for testing H_0: $\beta_1 = 0$ vs H_1: $\beta_1 \neq 0$ under different correlation design with $n = 100$, $p = 1000$ (from top to bottom: $\Sigma = \mathbf{I}_p$, $\Sigma = (0.5^{|i-j|})_{1\leq i,j\leq p}$, and $\Sigma = (0.9^{|i-j|})_{1\leq i,j\leq p}$).

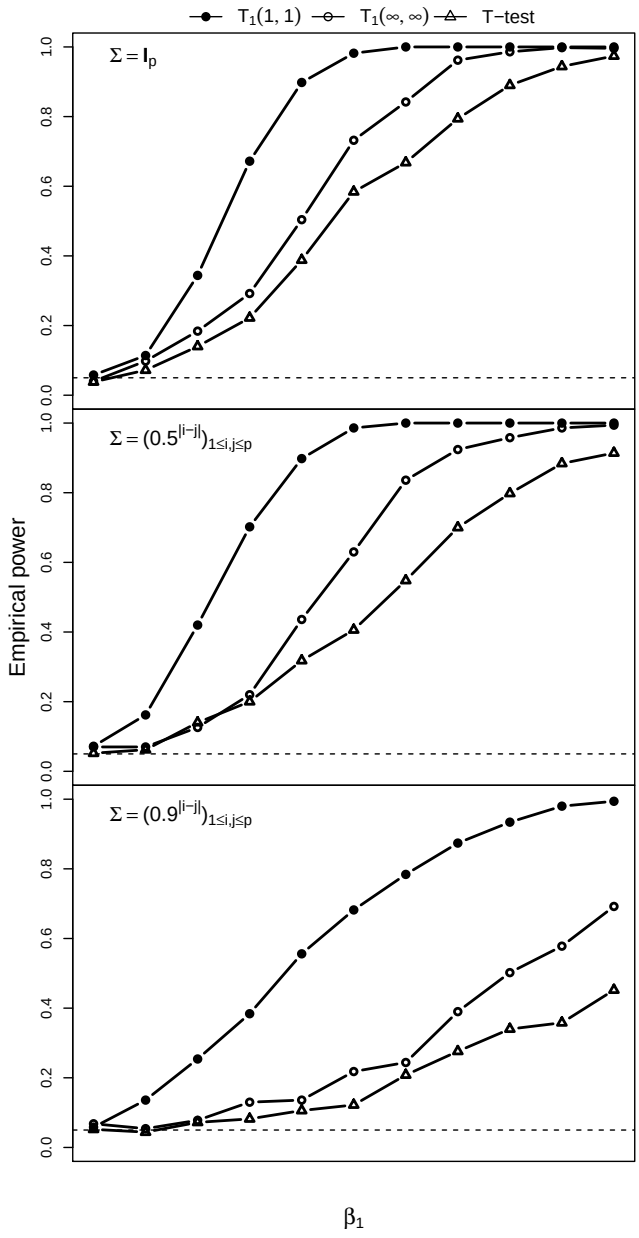

Figure 1.4: Empirical power for testing H_0: $\beta_1 = 0$ vs H_1: $\beta_1 \neq 0$ under different correlation design with $n = 100$, $p = 80$ (from top to bottom: $\Sigma = \mathbf{I}_p$, $\Sigma = (0.5^{|i-j|})_{1\leq i,j\leq p}$, and $\Sigma = (0.9^{|i-j|})_{1\leq i,j\leq p}$).

Test involving multiple coefficients

For the simultaneous test, we consider similar settings. We generate data according to the model

$$Y = \mathbf{X}\beta + \epsilon,$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I}_n)$ with $n = 100$ and $\beta = (\beta_1, \ldots, \beta_p)^T$. For $p = 1000$, we set $\beta_2 = \cdots = \beta_{10} = 1$, $\beta_{11} = \cdots = \beta_{1000} = 0$, and $\beta_1 \in \{1, 11/10, \ldots, 2\}$. For $p = 80$, we set $\beta_2 = \beta_3 = 1$. Other settings remain the same as those under which we tested H_0: $\beta_1 = 0$ versus H_1: $\beta_1 \neq 0$.

For $p = 1000$, We will test

$$H_0\text{: } \beta_1 = 1,\ \beta_{11} = 0,\ \beta_{12} = 0 \text{ vs } H_1\text{: } \beta_1 \neq 1 \text{ or } \beta_{11} \neq 0 \text{ or } \beta_{12} \neq 0.$$

and for $p = 80$, we will test

$$H_0\text{: } \beta_1 = 1,\ \beta_4 = 0,\ \beta_5 = 0 \text{ vs } H_1\text{: } \beta_1 \neq 1 \text{ or } \beta_4 \neq 0 \text{ or } \beta_5 \neq 0.$$

For the $p = 1000$ case, Figure 1.5 shows the power curves of the tests under different choices of Σ. The size is well controlled when H_0 is true, and $T_1(1, 1)$ achieved higher power than $T_1(\infty, \infty)$ as the correlation increases.

We only showed the power curve for our method, since the desparsified LASSO estimators cannot do simultaneous test.

For the $p = 80$ case, we will compare our method to the classical F-test. From the power curve in Figure 1.6, it is clear our method achieved considerably greater power than the F-test both for $T_1(1, 1)$ and $T_1(\infty, \infty)$, while controlling the size at the same time.

Overall, out method outperformed the F-test and works well for simultanoues test in high-dimesional settings.

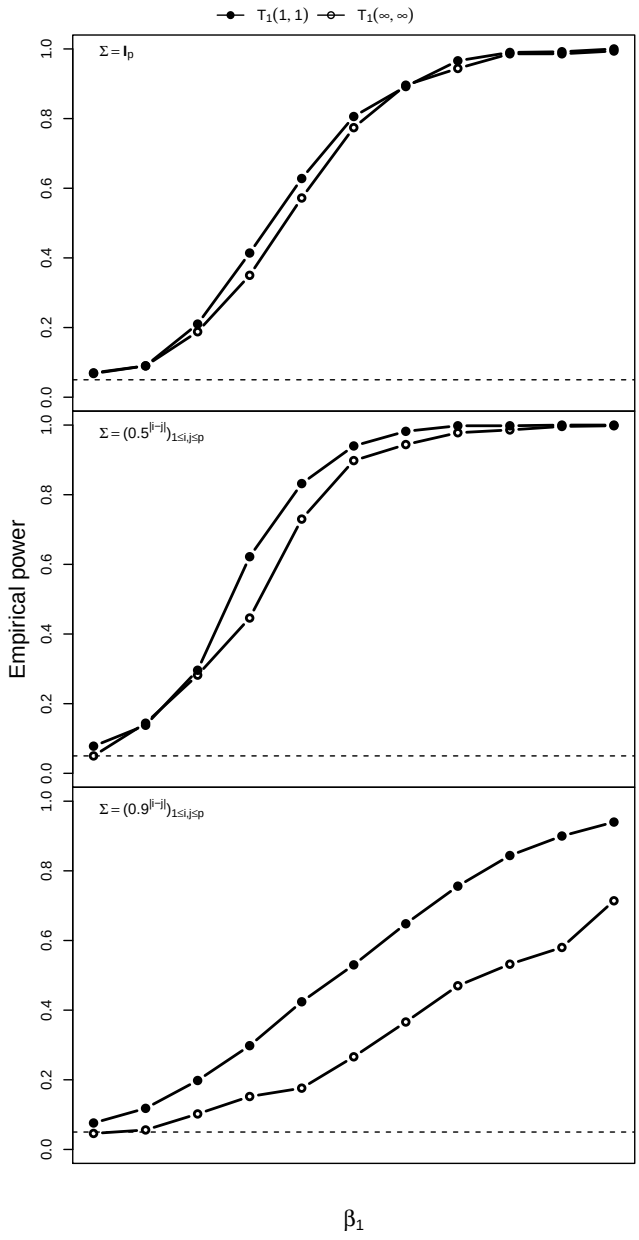

Figure 1.5: Multiple testing empirical power under different correlation design with $n = 100$, $p = 1000$ (from top to bottom: $\Sigma = \mathbf{I}_p$, $\Sigma = (0.5^{|i-j|})_{1\leq i,j\leq p}$, and $\Sigma = (0.9^{|i-j|})_{1\leq i,j\leq p}$).

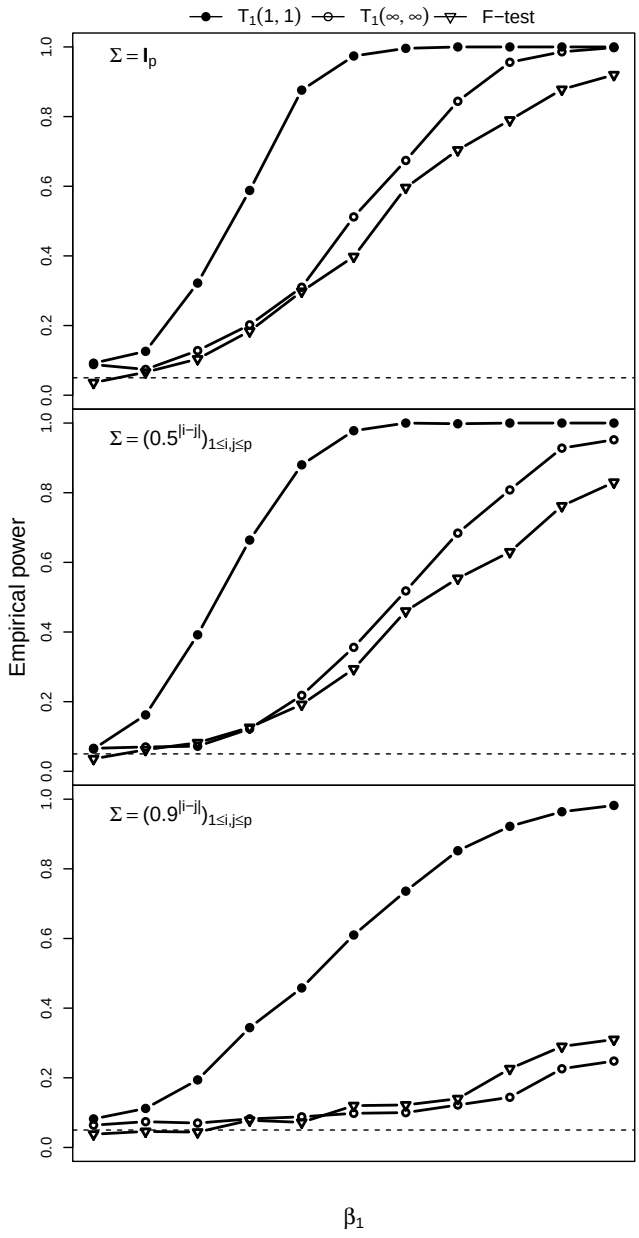

Figure 1.6: Multiple testing empirical power under different correlation design with $n = 100$, $p = 80$ (from top to bottom: $\Sigma = \mathbf{I}_p$, $\Sigma = (0.5^{|i-j|})_{1\leq i,j\leq p}$, and $\Sigma = (0.9^{|i-j|})_{1\leq i,j\leq p}$).

1.5 Real data analysis

To provide a concrete example, we consider a dataset about riboflavin (vitamin B2) production in Bacillus subtilis with 71 observations and 4088 variables Bühlmann

et al., 2014, Dezeure et al., 2015b, Geer et al., 2014. The response variable measures the logarithm of the riboflavin production rate and the predictors are logarithm of the expression level of 4088 genes. We will model the data with a high-dimensional linear model and carry out variable screening and inferences with the LOCO path statistic.

We use $T(1,1)$ in this part and obtained bootstrap P-values for each gene after variable screening. We screened in 342 genes with $T_j(1,1) > 0$, $j = 1, \ldots, 4088$. Based on our bootstrapped P-values, our method found the following 9 significant genes at 0.05 significance level: ARGF_at, XHLA_at, XHLB_at, XTRA_at, YCKE_at, YEBC_at, YOAB_at, YXLD_at and YYBG_at. Using the P-values based on the desparsified LASSO results in 0 significant genes in Dezeure et al., 2015b. Figures 1.7 and 1.8 show the variable importance for a small portion of genes. We will see only a few genes have large variable importance, while most genes have variable importance less than 1%.

Table 1.3 shows all variables with importance 1%, where YXLD_at and YOAB_at have the largest variable importance. Both genes are also tested significant using our bootstrap procedure.

Table 1.3: The first 10 most important genes.

Genes	Importance	P-value
YOAB_at	10.7%	0.0084
YXLD_at	10.3%	0.0084
ARGF_at	5.8%	0.0168
LYSC_at	5.2%	0.0924
YEBC_at	5.2%	0.0616
XHLA_at	5.1%	0.0140
YCKE_at	5.1%	0.0084
YDDK_at	4.4%	0.0560
SPOVAA_at	2.9%	0.1482
XHLB_at	2.7%	0.0194

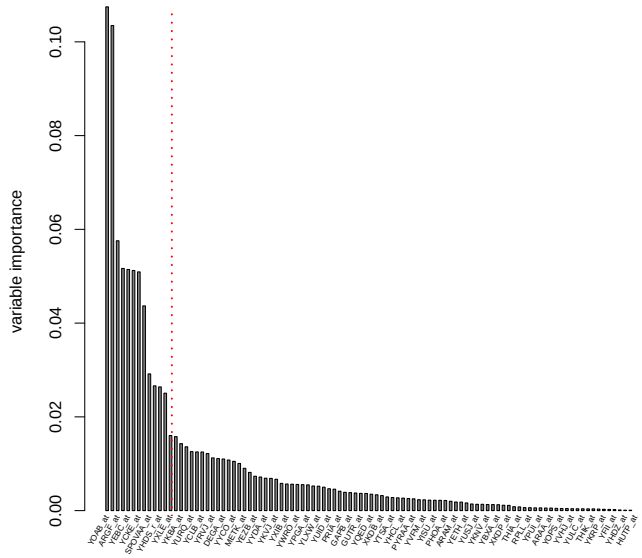

Figure 1.7: The first 100 most important genes. The vertical dotted line marks the variable importance at 1%.

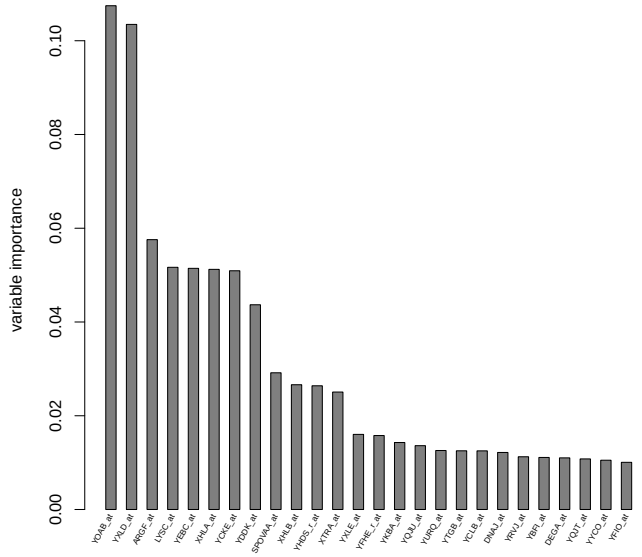

Figure 1.8: All genes with variable importance $> 1\%$.

1.6 Discussion

Our LOCO path statistic provides a new way to do variable screening and statistical inference in linear models. For variable screening, our method does not require the selection of tuning parameters and can achieve a greater probability of selecting a set of covariates that contains the true model than both SIS and ISIS. For statistical inference, our method provides reliable P-values in both high and low-dimensional settings. Overall, the proposed bootstrap method controls the size and in some cases achieves higher power than the desparsified LASSO of Geer et al., 2014. Moreover, our method can be used to test hypothesis simultaneously involving multiple coefficients. We believe the LOCO path idea can be readily extended to other settings.

Consider the regularization optimization problem

$$\hat{\beta} = \hat{\beta}(\lambda) := \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \, L(Y, \mathbf{X}\beta) + \lambda J(\beta), \tag{1.9}$$

where $L(\cdot)$ is a pre-defined loss function, $\lambda > 0$ is a tuning parameter which controls the level of regularization, and $J(\cdot)$ is a penalty function on β. The solution path $\hat{\beta}(\lambda)$ could be viewed as a 1-to-p mapping $\lambda \mapsto \hat{\beta}(\lambda)$ taking values in $(0, \infty)$ and returning values in $\mathbb{R}^p$. Since our measure of feature importance and variable screening procedure relies on the solution path only, we can easily adapt our method to (1.9), which includes logistic regression, Poisson regression and Cox models. Appropriate bootstrap methods for calibrating hypothesis tests would have to be worked out under each setting, which we leave to future work.

Chapter 2

A Generalized framework for High-Dimensional Inference based on the Leave-One-Covariate-Out regularization path

2.1 Introduction

The increasingly rapid emergence of high dimensional data, where the number of variables p may be larger than the sample size n, has necessitated the development of new statistical methodologies. In many areas, especially in genomics and biology, the number of genes measured will be substantially larger than the number of samples. Financial data, including data from high frequency trading, is also a source of high dimensional data.

In the high-dimensional regime, classical regression methodologies that work under lower-dimensional settings may fail. One way to handle high-dimensionality is to impose some sparsity constraints on these classical techniques. The LASSO Tibshirani, 1997 is a typical example for the linear and generalized linear regression problem, which can provide consistent estimators under some sparsity constraints. Many other variants of LASSO Zou and Hastie, 2005, Zou, 2006 and Tibshirani, Taylor, et al., 2011 have been proposed, which are suited to different problems under various settings.

Although the LASSO and other sparse estimators have become very popular in high-dimensional settings, it is hard to provide straightforward statistical inference

procedures based on these estimators, since their distributions are very complicated. The adaptive LASSO estimator Zou, 2006 has a limiting normal distribution under some conditions with the appropriate choice of tuning parameter. The desparsified LASSO, also has a limiting normal distribution Geer et al., 2014, Zhang and Cheng, 2017. However, these methods may be sensitive to choices of tuning parameters. Other techniques, including sample splitting Wasserman and Roeder, 2009, Meinshausen et al., 2009, bootstrapping, Das et al., 2019, Chatterjee and Lahiri, 2011, Chatterjee, Lahiri, et al., 2013 and sequential testing Lockhart et al., 2014 have also been proposed and provide valid inference under some conditions.

The proposed Leave-One-Covariate-Out(LOCO) solution path statistic, which has been shown to work well in the case of linear regression with a continuous response variable, provides a novel way to measure variable importance in high-dimensional settings. The LOCO path idea can also be used to construct a test statistic for testing hypothesis about regression coefficients. By bootstrapping the null distribution of the LOCO path statistic, it also provides a solid inference procedure for high-dimensional linear regression. In this paper, we will extend the LOCO solution path statistic to generalized linear models and to more general hypotheses. We would also consider other solution paths other than LASSO solutions path.

We organize this paper as follows: Section 2.2 illustrates the LOCO path statistic and Section 2.3 describes how we modify the LOCO path statistic for testing different hypothesis under different models. Section 2.4 presents simulation results and Sections 2.5 applies our method on different real data sets. Section 2.6 provides additional discussion.

2.2 Methodology

We consider the generalized linear model(GLM) when the number of covariates is large. Suppose we have data $\mathbf{X} = [X_1^T, X_2^T, \ldots, X_n^T]^T$ with $X_i \in \mathbb{R}^p, i = 1, \ldots, n$

and $Y \in \mathbb{R}^n$. The GLM assumes the distribution of Y belongs to the canonical-form exponential family with the following density function

$$f(Y) = \exp\left(Y\beta^T\mathbf{X} - b\left(\beta^T\mathbf{X}\right)\right)c(Y).$$

where $b(\cdot)$ and $c(\cdot)$ are some known function We also focus on the canonical link function for simplicity, so that

$$\mathbb{E}(Y_i|X_i = x) = b'((x^T\beta)) = \mu(x^T\beta), \tag{2.1}$$

where $\mu(\cdot)$ is the inverse link function and $\beta \in \mathbb{R}^p$ is a vector of unknown regression coefficients. We focus on the high-dimensional cases: $p > n$, but our method is also applicable to the lower-dimensional case, $p \leq n$.

The proposed Leave-One-Covariate-Out(LOCO) path statistic based on the LASSO estimator is a way to measure variable importance for linear models by calculating the change in the LASSO solution path due to removing one covariate from the model. We extend the LOCO path statistic to generalized linear models. To assess the importance of covariate j, we compare the complete solution path

$$\hat{\beta} := \hat{\beta}(\lambda) = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}}\, L(Y, \mathbf{X}\beta) + \lambda J(\beta), \quad \lambda > 0 \tag{2.2}$$

to the LOCO solution path, given by

$$\hat{\beta}^{(-j)} := \hat{\beta}^{(-j)}(\lambda) = \underset{\beta\in\mathbb{R}^p, \beta_j=0}{\operatorname{argmin}}\, L(Y, \mathbf{X}\beta) + \lambda J(\beta), \quad \lambda > 0, \tag{2.3}$$

for each $j \in \{1, \ldots, p\}$, where $L(\cdot)$ is a pre-defined loss function and $J(\cdot)$ is a penalty function. If covariate X_j is important, its importance will be reflected in a large difference between the paths $\hat{\beta}$ and $\hat{\beta}^{(-j)}$, whereas if it is not important, the difference between the paths $\hat{\beta}$ and $\hat{\beta}^{(-j)}$ will be small.

The LASSO solution path $\hat{\beta}$ can be viewed as a function taking values in $(0, \infty)$ and returning values in $\mathbb{R}^p$. As discussed in Chapter 1, the LOCO path statistic is defined as

$$T_j(s, t) = ||\hat{\beta} - \hat{\beta}^{(-j)}||_{s,t}, \tag{2.4}$$

where $\hat{\beta}$, $\hat{\beta}^{(-j)}$ is the full LASSO solution path and LOCO LASSO path, respectively. And the quantity $\|\cdot\|_{s,t}$ is defined as follows.

For a function $f : [0,\infty) \to \mathbb{R}^p$, such that $f(x) = (f_1(x), \ldots, f_p(x))^T$ with $f_j : [0,\infty) \to \mathbb{R}$, for $j = 1, \ldots, p$, we construct the following quantity. For all $\lambda > 0$, let

$$\|f\|_{q,q} = \begin{cases} (\sum_{k=1}^{p} \int_0^\infty |f_k(\lambda)|^q d\lambda)^{1/q}, & 0 < q < \infty \\ \max\limits_{1\le k\le p} \sup\limits_{\lambda>0} |f_k(\lambda)|, & q = \infty \end{cases}$$

Then the LOCO path statistic for covariate j is $T_j(s,t) = \|\hat{\beta} - \hat{\beta}^{(-j)}\|_{s,t}$.

The LASSO solution path is piecewise linear for linear regression models hence we can calculate $T_j(s,t)$ exactly. Rosset and Zhu, 2007 proposed sufficient conditions under which the LASSO solution path is piecewise linear: if $L(\cdot)$ is piecewise quadratic and $J(\cdot)$ is piecewise linear. Hence, for common generalized linear models such as logistic regression and Poisson regression, the solution path is *not* piecewise linear. If we consider more general penalty functions J, such as elastic net or the group LASSO penalty, the solution path is also not piecewise linear. Therefore, we need to calculate $T_j(q)$ in an approximate way by specifying a fine grid of λ.

Variable screening in ultra-high-dimensional settings

The ultra-high dimensional problem for linear regression is discussed by Fan and Lv, 2008, where the dimensionality p grows exponentially ($\log(p) = O(n^\zeta)$) as n grows. For such problems, preliminary variable screening is often done to reduce the dimension of the data. Sure Independence Screening(SIS) and Iterative Sure Independence Screening(ISIS) are proposed Fan and Lv, 2008. SIS and ISIS are also extended to generalized linear models Fan and Lv, 2010 by ranking maximum marginal likelihood estimators (MMLE).

Our method naturally adapts to ultra-high dimensional generalized linear models. We suggest the following screening procedure: Compute the solution path with all variables in the model. Then remove one variable at a time and compute the LOCO

solution path; compute the values $T_1(s,t),\ldots,T_p(s,t)$, which compare the solution path based on the full set of covariates to the LOCO solution paths. Then screen out variables for which $T_j(s,t) \leq \epsilon$, where ϵ is a user-specified threshold. Choosing $\epsilon = 0$ discards only those variables which never enter the solution path. We can also rank $T_j(s,t)$ and only select the top K variables, where we might choose K to be the sample size n.

Computational details

For linear models with continuous response, the solution path is piecewise linear so we can compute the entire LASSO solution path and calculate the exact value of $T_j(q)$. For most generalized linear models, the solution path is not piecewise linear, we need to compute it over a grid of λ and approximate $T_j(q)$.

To construct the grid of λ, we will make a sequence of λ decreasing from some pre-determined $\lambda_{\max}$ to $\lambda_{\min}$ on the logarithm scale. For linear models, λ_{max} is determined by $\frac{1}{n}\max_i |X_i^T Y|$. And for logistic models, $\lambda_{max} = \max_i \frac{1}{n}|((Y-\bar{Y})(1-\bar{Y}))^T X_i|$. For the choice of λ_{min}, we will let $\lambda_{\min} = \epsilon\lambda_{\max}$. In practice, we usually choose $\epsilon = 0.001$ and $K = 100$ and it usually gives very good approximations.

Hypothesis test

We now consider using the LOCO path idea to test hypotheses of the form

$$H_0\colon\ \beta_j = \beta_{0,j} \text{ for all } j \in \mathcal{A} \text{ versus } H_1\colon\ \beta_j \neq \beta_{0,j} \text{ for some } j \in \mathcal{A}, \tag{2.5}$$

for some $\{\beta_{0,j}, j \in \mathcal{A}\}$, where $\mathcal{A} \subset \{1,\ldots,p\}$. We first calculate the solution path with all variables included. Next, we compute the solution path subject to the constraint specified by the null hypothesis, which is given by

$$\hat{\beta}_0 := \hat{\beta}_0(\lambda) = \underset{\beta\in\mathbb{R}^p, \beta_j=\beta_{0,j}, j\in\mathcal{A}}{\operatorname{argmin}} L(Y, \mathbf{X}\beta) + \lambda J(\beta). \tag{2.6}$$

We then suggest as a test statistic for testing H_0 versus H_1 the quantity

$$T_0(s,t) = \|\hat{\beta} - \hat{\beta}_0\|_{s,t}, \tag{2.7}$$

which compares the solution paths $\hat{\beta}_0$ and $\hat{\beta}$. For testing the hypotheses

$$H_0\text{: } \beta_j = \beta_0 \text{ versus } H_1\text{: } \beta_j \neq \beta_0,$$

for some $j \in \{1, \ldots, p\}$, we have $\hat{\beta}_0 = \hat{\beta}^{(-j)}$, so that the test statistic $T_0(s,t)$ is equal to the LOCO path variable importance statistic $T_j(s,t) = \|\hat{\beta} - \hat{\beta}^{(-j)}\|_{s,t}$.

A bootstrap estimator of the null distribution

In order to test the hypotheses in (2.5) using the test statistic $T_0(s,t)$ in (2.7), we need to estimate the distribution of $T_0(s,t)$ under H_0. We propose estimating this null distribution using a parametric bootstrap procedure.

In Chapter 1, the adaptive LASSO estimator is proposed as an initial estimator of β. In this paper, we propose to use the adaptive elastic net as an initial estimator for $\tilde{\beta}$, which is better suited to settings in which the columns of $\mathbf{X}$ are highly correlated.

$$\hat{\beta}^{aenet} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}}\, L(Y, \mathbf{X}\beta) + \gamma \sum_{j=1}^{p} \hat{w}_j \left(\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j| \right), \tag{2.8}$$

where the tuning parameter γ is selected via 10-fold cross validation and the weights $\hat{w}_1, \ldots, \hat{w}_p$ are given by

$$\hat{w}_j = 1/|\hat{\beta}_j^L|, \quad j = 1, \ldots, p,$$

where $\hat{\beta}_1^L, \ldots, \hat{\beta}_p^L$ are the LASSO or elastic net estimates of $\beta_1, \ldots, \beta_p$ from (2.2) under the 10-fold cross validation choice of λ. In our simulations studies, we mainly used adaptive LASSO($\alpha = 1$) and it appears to work well. For the $p < n$ case the least-squares estimator could be used, though even in the lower dimensional case, we still recommend using the adaptive LASSO or elastic net estimator when p is close to n.

After obtaining $\tilde{\beta}$, we set a working initial $\tilde{\beta}_{H_0}$ as $\tilde{\beta}_{j,H_0} = \beta_{0,j}$ for $j \in \mathcal{A}$ and $\tilde{\beta}_{j,H_0} = \tilde{\beta}_j^{Ada}$ for $j \in \mathcal{A}^c$. The working initial $\tilde{\beta}_{H_0}$ will be used in the parametric bootstrap procedure.

The bootstrapped $\tilde{Y}^*$ will be sampled via

$$\tilde{Y}_i^* \sim \exp\left(Y\tilde{\beta}_{H_0}^T\mathbf{X}_i - b\left({\tilde{\beta}_{H_0}}^T\mathbf{X}_i\right)\right)c(Y), \quad i = 1, \ldots, n,$$

where $\tilde{Y}_i^*$ is the i-th entry of the $n \times 1$ vector $\tilde{Y}^*$.

For testing the hypotheses in (2.5), the bootstrap versions $\hat{\beta}^*$ and $\hat{\beta}_0^*$ of $\hat{\beta}$ and $\hat{\beta}_0$ are constructed as

$$\hat{\beta}^* := \hat{\beta}^*(\lambda) = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}}\, L(\tilde{Y}^*, \mathbf{X}\beta) + \lambda J(\beta), \tag{2.9}$$

and

$$\hat{\beta}_0^* := \hat{\beta}_0^*(\lambda) = \underset{\beta \in \mathbb{R}^p, \beta_j = \beta_{0,j}, j \in \mathcal{A}}{\operatorname{argmin}} L(\tilde{Y}^*, \mathbf{X}\beta) + \lambda J(\beta) \tag{2.10}$$

respectively. Then the bootstrap version of $T_0(s,t) = \|\hat{\beta} - \hat{\beta}_0\|_{s,t}$ is given by

$$T_0^*(s,t) = \|\hat{\beta}^* - \hat{\beta}_0^*\|_{s,t}.$$

Given a large number B of Monte-Carlo replicates of $T_0^*(s,t)$, denoted by, say, $T_0^{*,(1)}(s,t) < \cdots < T_0^{*,(B)}(s,t)$, when ordered, our bootstrap-based test of H_0 at significance level α has decision rule

$$\text{Reject } H_0 \text{ if and only if } T_0(s,t) > T_0^{*,(\lfloor B(1-\alpha)\rfloor)},$$

where $T_0^{*,(\lfloor B(1-\alpha)\rfloor)}$ is the Monte-Carlo approximation to the bootstrap estimator of the upper α-quantile of the null distribution of $T_0(s,t)$, and $\lfloor\cdot\rfloor$ is the floor function. We could also obtain a bootstrapped P-value as

$$B^{-1}\sum_{i=1}^{B} I\left\{T_0^{*,(i)}(s,t) > T_0(s,t)\right\},$$

where $I(\cdot)$ is the indicator function.

Now we will establish some theoretical results to justify the effectiveness of our method. Consider for linear regression model

$$Y = \mathbf{X}\beta + \epsilon,$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ with sample size n and number of covariates p. We first start with a special case with orthogonal design $n^{-1}\mathbf{X}^T\mathbf{X} = \mathbf{I}$. In that case, the LASSO estimator can be solved analytically:

$$\widehat{\beta}_j(\lambda) = \begin{cases} z_j - \lambda, & \text{if } z_j > \lambda \\ 0, & \text{if } |z_j| \leq \lambda \\ z_j + \lambda, & \text{if } z_j < -\lambda \end{cases},$$

where z_j is the OLS estimator, and $z_j \sim \mathcal{N}(\beta_j, \sigma^2/n)$.

Hence we can derive the sampling distribution of $\widehat{\beta}_j(\lambda)$ for fixed λ as follows.

$$\mathbb{P}(\widehat{\beta_j}(\lambda) \leq x) = \begin{cases} \Phi(\frac{x+\lambda-\beta_j}{\sigma/\sqrt{n}}), & \text{if } x > 0 \\ \Phi(\frac{\lambda-\beta_j}{\sigma/\sqrt{n}}) - \Phi(\frac{-\lambda-\beta_j}{\sigma/\sqrt{n}}), & \text{if } x = 0 \\ \Phi(\frac{x-\lambda-\beta_j}{\sigma/\sqrt{n}}), & \text{if } x < 0 \end{cases}$$

Hence $\widehat{\beta}_j(\lambda)$ is a biased estimator for β_j since $\mathbb{E}(\widehat{\beta}_j(\lambda)) \neq \beta_j$, unless $\beta_j = 0$.

In the adaptive LASSO paper Zou, 2006, the authors consider the lasso estimates, $\hat{\boldsymbol{\beta}}^{(n)}$

$$\hat{\boldsymbol{\beta}}^{(n)}(\lambda) = \arg\min_{\boldsymbol{\beta}} \|Y - \mathbf{X}\beta\|_2^2 + \lambda_n \|\beta\|_1 \tag{2.11}$$

where λ_n varies with n. In their lemma 1, they proved if $\lambda_n/n \to \lambda_0 \geq 0$, then $\hat{\boldsymbol{\beta}}^{(n)} \to_p \arg\min V_1$ where

$$V_1(\mathbf{u}) = (\mathbf{u} - \boldsymbol{\beta}^*)^T \mathbf{C} (\mathbf{u} - \boldsymbol{\beta}^*) + \lambda_0 \|\mathbf{u}\|_1, \tag{2.12}$$

$\boldsymbol{\beta}^*$ is the true regression coefficients and $\frac{1}{n}\mathbf{X}^T\mathbf{X} \to \mathbf{C}$.

Inspired by $V_1\mathbf{u}$, we denote the following the sample path:

$$\hat{\boldsymbol{\beta}}(\lambda) = \arg\min_{\boldsymbol{\beta}} \frac{1}{n} \|Y - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_1 ,$$

And for each λ in the sample path, we denote the following the population path:

$$\tilde{\boldsymbol{\beta}}(\lambda) = \arg\min_{\boldsymbol{\beta}} \frac{1}{n} (\beta - \beta^*)^T \mathbf{X}^T \mathbf{X} (\beta - \beta^*) + \lambda \|\beta\|_1 .$$

We consider a simple model

$$Y = Z\beta + \epsilon,$$

where $\beta \in \mathbb{R}$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Suppose we have data $(z_i, y_i), i = 1, \ldots, n$, we want to solve

$$\underset{\beta}{\text{minimize}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - z_i\beta)^2 + \lambda|\beta| \right\}.$$

It can be shown the sample path $\widehat{\beta}(\lambda) = \mathcal{S}_\lambda(\hat{\beta}^{LS})$, where $\hat{\beta}^{LS}$ is the least square estimator. And for the population path, we need to solve

$$\underset{\beta}{\text{minimize}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (z_i\beta^* - z_i\beta)^2 + \lambda|\beta| \right\}.$$

We can also show the population path $\tilde{\beta}(\lambda) = \mathcal{S}_\lambda(\beta^*)$, and hence

$$\sqrt{n}\{\hat{\boldsymbol{\beta}}(\lambda) - \tilde{\boldsymbol{\beta}}(\lambda)\} = \sqrt{n}\{\mathcal{S}_\lambda(\hat{\beta}^{LS}) - \mathcal{S}_\lambda(\beta^*)\}.$$

Hence if $\beta^* \neq 0$, for $\lambda < |\hat{\beta}^{LS}|$, we have

$$\sqrt{n}\{\mathcal{S}_\lambda(\hat{\beta}^{LS}) - \mathcal{S}_\lambda(\beta^*)\} \to_d \mathcal{N}(0, \sigma^2).$$

This means the LASSO estimator is consistent. For orthogonal design case, the proof is similar.

And our $T_j(1) = \frac{1}{2}|\hat{\beta}_j^{LS}|^2$ and the population version $\tilde{T}_j(1) = \frac{1}{2}|\hat{\beta}_j^*|^2$. Hence $\sqrt{n}\{T_j(1) - \tilde{T}_j(1)\}$ has asymptotic Normal distribution. This shows although the LASSO estimator $\hat{\beta}(\lambda)$ may not have asymptotic Normal distribution for all λ and its consistency depends on λ. After integrating λ out, we may have a consistent estimator back. Extensions to more general settings still worth our investigation.

2.3 Extension to more general problems

We consider testing more general hypothesis of the form H_0: $D\beta = d$ vs H_1: $D\beta \neq d$, where D is a $m \times p$ matrix with $\text{rank}(D) = m$. We will restrict $m < p$ since otherwise the null hypothesis reduces to a null hypothesis of the form in Section 2.2. To test $D\beta = d$, we borrow some ideas from the generalized LASSO Tibshirani, Taylor, et al., 2011 and describe as follows.

First, we augment the matrix D to be a full rank $p \times p$ matrix $\tilde{D} = \begin{bmatrix} D \\ I \end{bmatrix}$, and we define the vector $\tilde{d} = \begin{bmatrix} d \\ \mathbf{0} \end{bmatrix}$. Then we let $\theta = (\theta_1, \theta_2)^T = \tilde{D}\beta - \tilde{d}$. So we can solve $\beta = \tilde{D}^{-1}(\theta + \tilde{d})$. And then we can follow 2.2 and compare the full solution path

$$\hat{\theta} := \hat{\theta}(\lambda) = \underset{\theta \in \mathbb{R}^p}{\text{argmin}}\, L(Y, \mathbf{X}\widetilde{D}^{-1}(\theta + d)) + \lambda J(\theta),$$

with the constrained solution path

$$\hat{\theta}_0 := \hat{\theta}_0(\lambda) = \underset{\theta \in \mathbb{R}^p, \theta_1 = 0}{\text{argmin}}\, L(Y, \mathbf{X}\widetilde{D}^{-1}(\theta + d)) + \lambda J(\theta).$$

A simple example would be testing H_0: $\beta_1 = \beta_2$ vs H_1: $\beta_1 \neq \beta_2$ in the linear regression model $Y = \mathbf{X}\beta + \epsilon$. In that case, $D = (1, -1)$ and $d = 0$. The augmented $\tilde{D}$ and $\tilde{D}^{-1}$ are $\begin{bmatrix} 1 & -1 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$ and $\begin{bmatrix} 1 & 1 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$, respectively.

Hence, after transformation, the new design matrix $\mathbf{X}\widetilde{D}^{-1}$ is basically $(X_1, X_1 + X_2, X_3, \ldots, X_p)$. And for the transformed regression $Y = \mathbf{X}\widetilde{D}^{-1}\theta + \epsilon$, we can easily test $\theta_1 = 0$ using the methodology discussed in Section 2.2.

2.4 Simulation studies

We now study via simulation the effectiveness of the LOCO path statistic as a variable screening tool in GLM as well as the properties of our proposed LOCO-path-based tests of hypotheses which use the parametric bootstrap to estimate the null distributions of the test statistics. We first present the variable screening results.

Variable screening

We study the performance of the our variable screening procedure described in Section 2.2 by generating data from the Logistic regression model

$$\mathbb{E}(Y|\mathbf{X}) = \mu(\beta X_1 + \beta X_2 + \beta X_3),$$

where $\mu(x) = \exp(x)/(1+\exp(x))$, with a total of p predictors $X_1, \ldots, X_p$ in the model. The settings are similar to those considered in Chapter 1. The rows of the design matrix are generated as independent multivariate normal random vectors with covariance matrix $\Sigma = (\rho^{|i-j|})_{1\leq i,j\leq p}$, where $\rho = 0, 0.1, 0.5$ and 0.9. Models with $\beta = 1, 2, 3$, $p = 100$, $n = 20$, and $p = 1000$, $n = 50$ are considered. To compare our results with SIS and ISIS, we utilized the R package `SIS` Saldana and Feng, 2018. We simulated 200 data sets and for each model we calculate $T_j(1,1)$ and $T_j(2,2)$ for $j = 1, 2, \ldots, p$ and select the top $n-1$ covariates, selecting the same number of covariates with SIS and ISIS in order to make a fair comparison. For our method, we utilized the R package `glmnet` Friedman et al., 2010 to calculate our test statistic.

In Table 2.1 we show the proportion of times that the true model is contained in the set of selected covariates by our method and by the SIS and ISIS variable screening methods. In most cases, the model selected by our LOCO-path-based method contains the true model with greater frequency than that of the SIS and ISIS methods. On the other hand, the success of ISIS depends on the appropriate selection of tuning parameters.

Table 2.1: Proportion of times SIS, ISIS and our method selected a set of covariates containing $\{X_1, X_2, X_3\}$ for Logistic regression model.

Setting	β	$T_j(1,1)$	$T_j(2,2)$	SIS	ISIS
$p = 1000, n = 50, \Sigma = \mathbf{I}_p$	1	0.705	0.705	0.235	0.105
	2	0.945	0.945	0.640	0.420
	3	0.975	0.975	0.785	0.645
$p = 1000, n = 50, \Sigma = (0.1^{\lvert i-j \rvert})_{1 \le i,j \le p}$	1	0.780	0.775	0.455	0.210
	2	0.960	0.960	0.770	0.495
	3	0.980	0.980	0.825	0.570
$p = 1000, n = 50, \Sigma = (0.5^{\lvert i-j \rvert})_{1 \le i,j \le p}$	1	0.930	0.930	0.850	0.225
	2	1.000	1.000	1.000	0.390
	3	1.000	1.000	1.000	0.400
$p = 1000, n = 50, \Sigma = (0.9^{\lvert i-j \rvert})_{1 \le i,j \le p}$	1	0.870	0.870	1.000	0.160
	2	0.980	0.980	1.000	0.245
	3	1.000	1.000	1.000	0.250
$p = 100, n = 20, \Sigma = \mathbf{I}_p$	1	0.675	0.675	0.140	0.115
	2	0.870	0.870	0.295	0.265
	3	0.900	0.900	0.365	0.285
$p = 100, n = 20, \Sigma = (0.1^{\lvert i-j \rvert})_{1 \le i,j \le p}$	1	0.785	0.785	0.215	0.150
	2	0.895	0.895	0.365	0.280
	3	0.915	0.910	0.525	0.380
$p = 100, n = 20, \Sigma = (0.5^{\lvert i-j \rvert})_{1 \le i,j \le p}$	1	0.865	0.860	0.570	0.280
	2	0.965	0.960	0.850	0.500
	3	0.985	0.985	0.890	0.490
$p = 100, n = 20, \Sigma = (0.9^{\lvert i-j \rvert})_{1 \le i,j \le p}$	1	0.900	0.895	0.970	0.355
	2	0.975	0.975	1.000	0.345
	3	0.990	0.990	1.000	0.425

Study of power and size of LOCO path tests of hypotheses

Test involving a single coefficient

We first study the size and power of the LOCO path test for testing the hypotheses H_0: $\beta_j = 0$ versus H_1: $\beta_j \neq 0$ for some $j \in \{1, \ldots, p\}$, where the rejection region of the test is calibrated using the parametric bootstrap procedure described in Section 2.2. We consider the test statistics $T_j(1,1)$, $T_j(2,2)$, and $T_j(\infty,\infty)$. In high-dimensional ($p \geq n$) settings, we compare the empirical size and power of our test based on these statistics with the test based on the desparsified LASSO estimator Geer et al., 2014.

We use the R package `hdi` Dezeure et al., 2015a to obtain the P-value based on the desparsified LASSO estimator using default settings Dezeure et al., 2015b for Logistic regression. Poisson regression is not available for `hdi`. And we utilize the R package `glmnet` Friedman et al., 2010 to implement our method. In low-dimensional ($p < n$) settings, we compare the performance of our tests to that of the Wald type test.

We generate data from the logistic regression and Poisson regression models with

$$E(Y|\mathbf{X}) = \mu(\mathbf{X}\beta),$$

where $\mu(x) = \exp(x)/(1+\exp(x))$ for logistic regression and $\mu(x) = \exp(x)$ for Poisson regression. We consider $p = 80$ and $p = 1000$. For $p = 1000$, we set $\beta = (\beta_1, \ldots, \beta_p)^T$ such that $\beta_2 = \cdots = \beta_{10} = 1$, $\beta_{11} = \cdots = \beta_{1000} = 0$ for logistic regression, $\beta_2 = \beta_3 = 1$, $\beta_4 = \cdots = \beta_{1000} = 0$ for Poisson regression. For $p = 80$, we set $\beta_2 = \beta_3 = 3$, $\beta_4 = \cdots = \beta_{80} = 0$ for logistic regression and $\beta_2 = \beta_3 = 1$, $\beta_4 = \cdots = \beta_{80} = 0$ for Poisson regression. To simulate the power curve, we take different values of $\beta_1 \in \{0/10, 5/10, \ldots, 5\}$ for Logistic regression and $\beta_1 \in \{0/50, 1/50, \ldots, 10/50\}$ for Poisson regression . Each row of $\mathbf{X}$ is generated independently from the multivariate normal distribution $\mathcal{N}(0, \Sigma)$, where we consider different choices of the $p \times p$ covariance matrix Σ. For each choice of Σ and for each value of β_1, we generate $N = 500$ data sets and with each data set we test H_0: $\beta_1 = 0$ versus H_1: $\beta_1 \neq 0$. For each data set, we draw $B = 500$ bootstrap samples to estimate the null distribution. We record the proportion of rejections of H_0 at the $\alpha = 0.05$ significance level.

The empirical size of the simulation for Logistic regression testing H_0: $\beta_1 = 0$ under $p = 1000$ and $p = 80$, is given in Table 2.2 and 2.4. We also recorded the empirical size of the test based on the desparsified LASSO estimator. For $p = 1000$, our method tends to be too conservative for some designs. However, the size of desparsified LASSO estimator is even worse. For $p = 80$, the size of our method is well-controlled, while the desparsified LASSO estimator is too conservative.

The empirical size of the simulation for Poisson regression testing H_0: $\beta_1 = 0$ under $p = 1000$ and $p = 80$, is given in Table 2.3 and 2.5. The `hdi` package only supports linear and Logistic regression currently so we did not show the desparsified LASSO results. It is clear that our method nicely controlled the size under different choices of Σ and different quantities $T_1(1,1)$, $T_1(2,2)$ and $T_1(\infty,\infty)$.

The Logistic regression empirical power curves of our test based on the LOCO path statistics $T_1(1,1)$, $T_1(2,2)$ and $T_1(\infty,\infty)$ as well as of the test based on the desparsified LASSO under settings $n = 1000$ and $p = 80$ over the values $\beta_1 \in \{0/50, 5/10, \ldots, 5\}$ are depicted in Figures 2.1 and 2.3. For most cases, $T_1(1,1)$ and $T_1(2,2)$ have a higher power than $T_1(\infty,\infty)$, while $T_1(\infty,\infty)$ loses some power under the correlated design. For $p = 1000$, under different designs, our method outperformed desparsified LASSO and our method achieves higher power than desparsified LASSO as the correlation increases. For $p = 80$, our method achieves comparable power with desparsified LASSO as the correlation increases. We also noticed that for $p = 80$, the desparsified LASSO may fail, although the frequency is very rare (3-5 out of 500 simulations). It is also interesting to note that we cannot do Wald-type test even for some lower-dimensional case ($n = 100, p = 80$). The maximum likelihood estimator is undefined with very high probability due to complete or quasi-complete separation.

Overall, for the Logistic regression, our methods achieves comparable or higher power, with size better controlled, compared to the desparsified LASSO method.

The Poisson regression empirical power curves of our test based on the LOCO path statistics $T_1(1,1)$, $T_1(2,2)$ and $T_1(\infty,\infty)$ under settings $n = 1000$ and $p = 80$ over the values $\beta_1 \in \{0/10, 1/50, \ldots, 10/50\}$ are depicted in Figures 2.2 and 2.4. For $p = 80$, we also included the power curves using the Wald-type test. For most cases, $T_1(1,1)$ and $T_1(2,2)$ have comparable and highest power, while $T_1(\infty,\infty)$ loses some power under the correlated design. For $p = 80$, our method achieves significantly higher power than the Wald-type test as the correlation increases.

Table 2.2: Logistic regression empirical size for testing H_0: $\beta_1 = 0$ under different correlation design with $n = 100$, $p = 1000$.

Design	Method	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
$\Sigma = \mathbf{I}_p$	$T_1(1,1)$	0.076	0.062	0.038	0.008
	$T_1(2,2)$	0.082	0.060	0.038	0.014
	$T_1(\infty,\infty)$	0.054	0.046	0.026	0.006
	De-sparsified	0.128	0.068	0.030	0.004
$\Sigma = (0.5^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.082	0.048	0.022	0.006
	$T_1(2,2)$	0.096	0.056	0.030	0.014
	$T_1(\infty,\infty)$	0.132	0.070	0.036	0.010
	De-sparsified	0.032	0.016	0.008	0.000
$\Sigma = (0.9^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.198	0.112	0.068	0.040
	$T_1(2,2)$	0.240	0.136	0.084	0.026
	$T_1(\infty,\infty)$	0.232	0.154	0.086	0.036
	De-sparsified	0.078	0.038	0.016	0.006
$\Sigma = (0.5^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.038	0.018	0.008	0.002
	$T_1(2,2)$	0.048	0.028	0.016	0.004
	$T_1(\infty,\infty)$	0.038	0.012	0.002	0.000
	De-sparsified	0.018	0.006	0.000	0.000
$\Sigma = (0.8^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.062	0.036	0.016	0.000
	$T_1(2,2)$	0.082	0.032	0.016	0.006
	$T_1(\infty,\infty)$	0.054	0.020	0.006	0.002
	De-sparsified	0.034	0.008	0.006	0.000

Table 2.3: Poisson regression empirical size for testing H_0: $\beta_1 = 0$ under different correlation design with $n = 100$, $p = 1000$.

Design	Method	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
$\Sigma = \mathbf{I}_p$	$T_1(1,1)$	0.182	0.106	0.058	0.022
	$T_1(2,2)$	0.198	0.096	0.050	0.008
	$T_1(\infty,\infty)$	0.188	0.088	0.038	0.006
$\Sigma = (0.5^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.194	0.102	0.054	0.010
	$T_1(2,2)$	0.248	0.108	0.048	0.010
	$T_1(\infty,\infty)$	0.242	0.124	0.062	0.020
$\Sigma = (0.9^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.264	0.148	0.066	0.022
	$T_1(2,2)$	0.266	0.174	0.120	0.048
	$T_1(\infty,\infty)$	0.246	0.146	0.084	0.036
$\Sigma = (0.5^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.200	0.096	0.050	0.010
	$T_1(2,2)$	0.184	0.076	0.038	0.016
	$T_1(\infty,\infty)$	0.192	0.102	0.062	0.006
$\Sigma = (0.8^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.186	0.096	0.066	0.014
	$T_1(2,2)$	0.180	0.096	0.056	0.008
	$T_1(\infty,\infty)$	0.152	0.076	0.038	0.010

Table 2.4: Logistic regression empirical size for testing H_0: $\beta_1 = 0$ under different correlation design with $n = 100$, $p = 80$.

Design	Method	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
$\Sigma = \mathbf{I}_p$	$T_1(1,1)$	0.200	0.104	0.056	0.018
	$T_1(2,2)$	0.190	0.088	0.060	0.026
	$T_1(\infty,\infty)$	0.208	0.110	0.066	0.032
	De-sparsified	0.090	0.032	0.014	0.000
$\Sigma = (0.5^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.188	0.100	0.054	0.020
	$T_1(2,2)$	0.184	0.086	0.050	0.016
	$T_1(\infty,\infty)$	0.220	0.112	0.062	0.026
	De-sparsified	0.120	0.058	0.020	0.002
$\Sigma = (0.9^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.200	0.118	0.078	0.040
	$T_1(2,2)$	0.190	0.118	0.066	0.024
	$T_1(\infty,\infty)$	0.216	0.124	0.084	0.028
	De-sparsified	0.187	0.089	0.050	0.012
$\Sigma = (0.5^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.166	0.080	0.038	0.012
	$T_1(2,2)$	0.166	0.086	0.044	0.016
	$T_1(\infty,\infty)$	0.154	0.072	0.032	0.006
	De-sparsified	0.102	0.042	0.016	0.002
$\Sigma = (0.8^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.144	0.084	0.050	0.008
	$T_1(2,2)$	0.176	0.082	0.032	0.010
	$T_1(\infty,\infty)$	0.204	0.082	0.036	0.010
	De-sparsified	0.145	0.050	0.024	0.006

Table 2.5: Poisson regression empirical size for testing H_0: $\beta_1 = 0$ under different correlation design with $n = 100$, $p = 80$.

Design	Method	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
$\Sigma = \mathbf{I}_p$	$T_1(1,1)$	0.188	0.088	0.044	0.002
	$T_1(2,2)$	0.188	0.110	0.056	0.008
	$T_1(\infty,\infty)$	0.198	0.096	0.048	0.004
	Wald	0.126	0.058	0.026	0.002
$\Sigma = (0.5^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.212	0.108	0.060	0.010
	$T_1(2,2)$	0.186	0.090	0.046	0.018
	$T_1(\infty,\infty)$	0.210	0.104	0.044	0.010
	Wald	0.068	0.036	0.010	0.000
$\Sigma = (0.9^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.218	0.098	0.058	0.018
	$T_1(2,2)$	0.230	0.098	0.048	0.018
	$T_1(\infty,\infty)$	0.162	0.064	0.030	0.008
	Wald	0.040	0.016	0.006	0.000
$\Sigma = (0.5^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.180	0.090	0.050	0.024
	$T_1(2,2)$	0.218	0.090	0.044	0.018
	$T_1(\infty,\infty)$	0.206	0.108	0.056	0.012
	Wald	0.084	0.046	0.020	0.000
$\Sigma = (0.8^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.162	0.076	0.026	0.010
	$T_1(2,2)$	0.210	0.118	0.062	0.010
	$T_1(\infty,\infty)$	0.210	0.126	0.078	0.022
	Wald	0.072	0.026	0.004	0.000

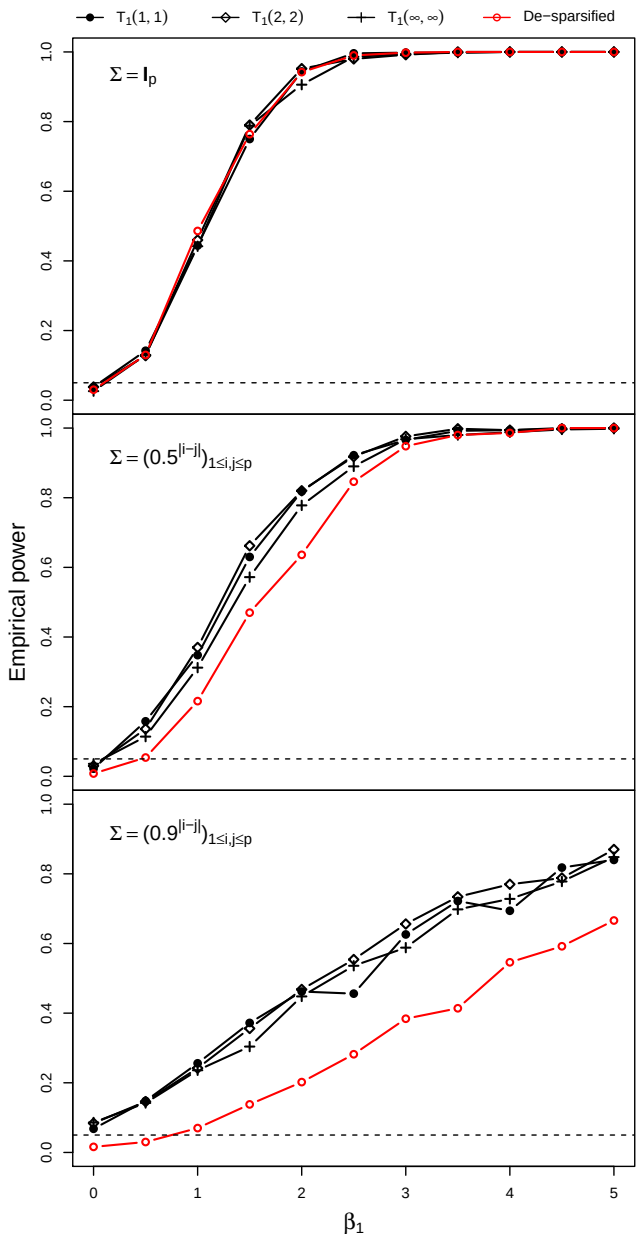

Figure 2.1: Logistic regression empirical power for testing H_0: $\beta_1 = 0$ vs H_1: $\beta_1 \neq 0$ under different correlation design with $n = 100$, $p = 1000$ (from top to bottom: $\Sigma = \mathbf{I}_p$, $\Sigma = (0.5^{|i-j|})_{1\leq i,j\leq p}$, and $\Sigma = (0.9^{|i-j|})_{1\leq i,j\leq p}$).

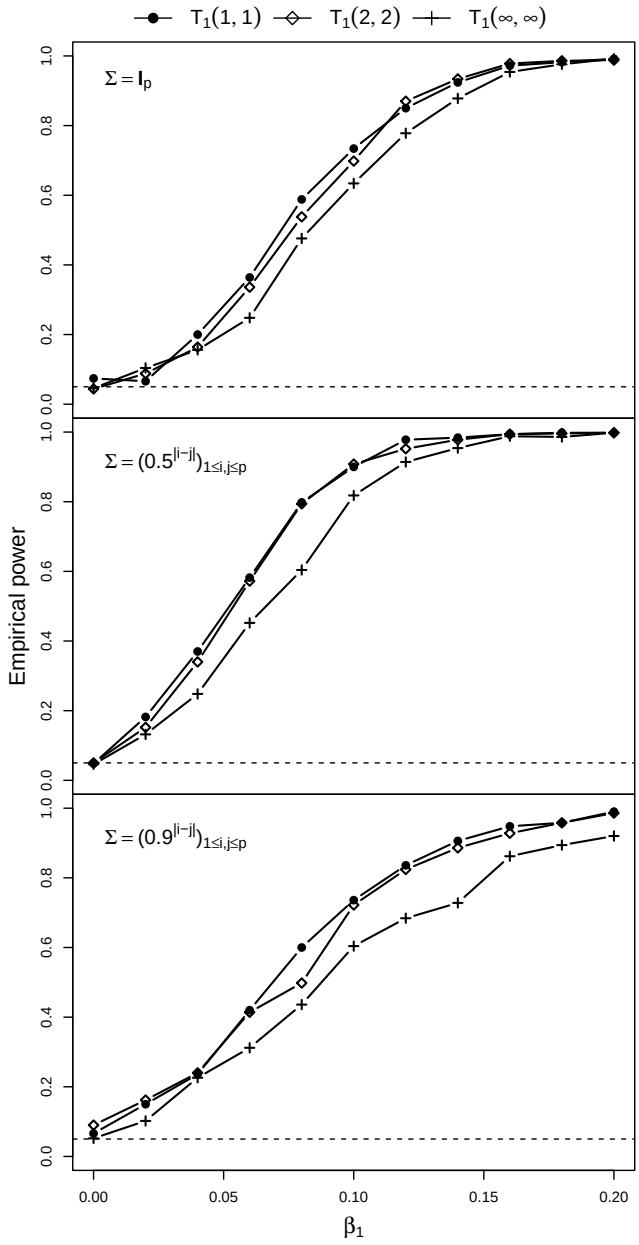

Figure 2.2: Poisson regression empirical power for testing H_0: $\beta_1 = 0$ vs H_1: $\beta_1 \neq 0$ under different correlation design with $n = 100$, $p = 1000$ (from top to bottom: $\Sigma = \mathbf{I}_p$, $\Sigma = (0.5^{|i-j|})_{1\leq i,j\leq p}$, and $\Sigma = (0.9^{|i-j|})_{1\leq i,j\leq p}$).

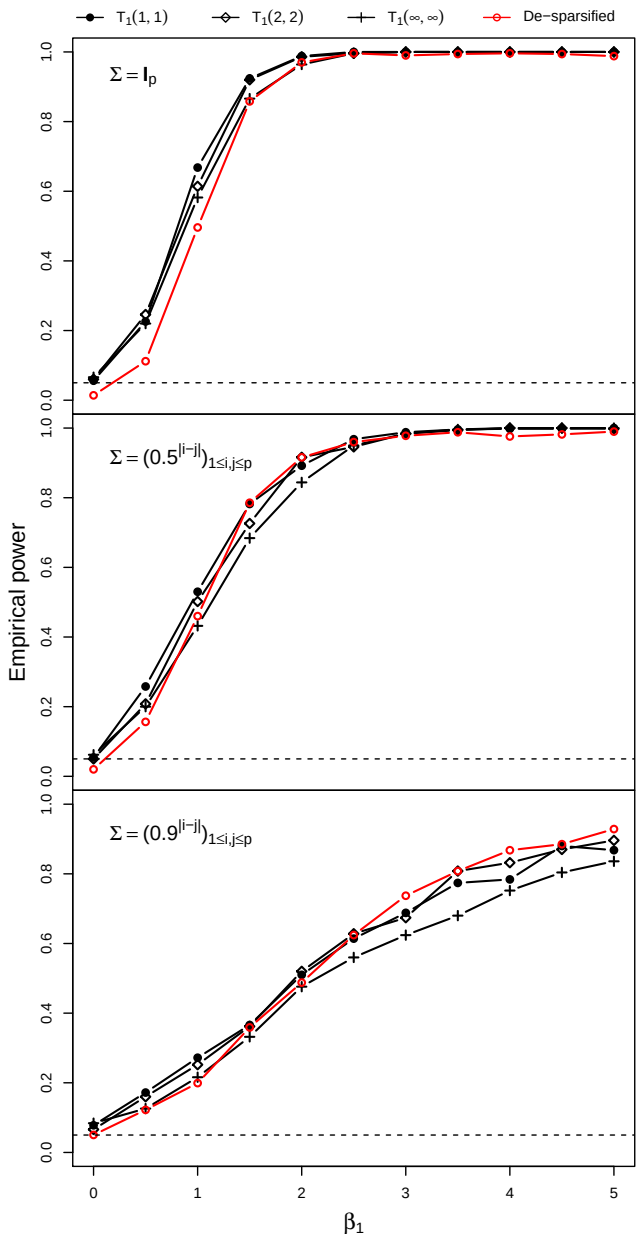

Figure 2.3: Logistic regression empirical power for testing H_0: $\beta_1 = 0$ vs H_1: $\beta_1 \neq 0$ under different correlation design with $n = 100$, $p = 80$ (from top to bottom: $\Sigma = \mathbf{I}_p$, $\Sigma = (0.5^{|i-j|})_{1\leq i,j\leq p}$, and $\Sigma = (0.9^{|i-j|})_{1\leq i,j\leq p}$).

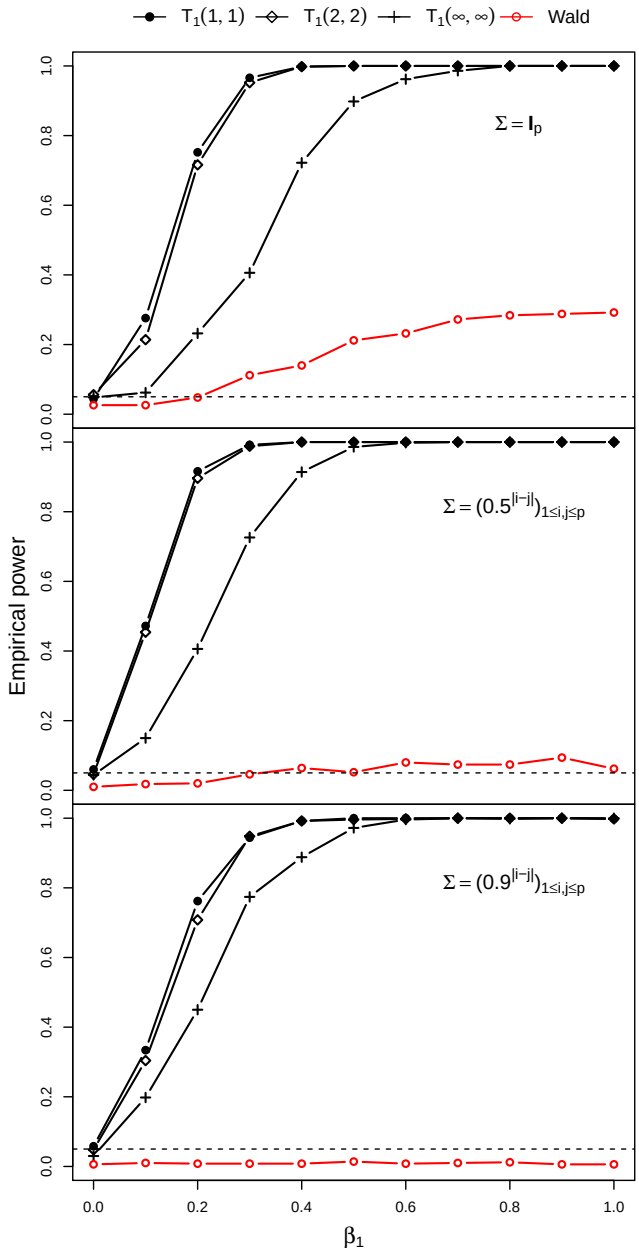

Figure 2.4: Poisson regression empirical power for testing H_0: $\beta_1 = 0$ vs H_1: $\beta_1 \neq 0$ under different correlation design with $n = 100$, $p = 80$ (from top to bottom: $\Sigma = \mathbf{I}_p$, $\Sigma = (0.5^{|i-j|})_{1\leq i,j\leq p}$, and $\Sigma = (0.9^{|i-j|})_{1\leq i,j\leq p}$).

Test involving multiple coefficients

We generate data from similar model as in Section 2.4, with

$$E(Y|\mathbf{X}) = \mu(\mathbf{X}\beta),$$

with $n = 100$ and $\beta = (\beta_1, \ldots, \beta_p)^T$. For $p = 1000$, we set $\beta_2 = \cdots = \beta_{10} = 1$, $\beta_{11} = \cdots = \beta_{1000} = 0$ for Logistic regression, $\beta_2 = \beta_3 = 1$, $\beta_4 = \cdots = \beta_{1000} = 0$ for Poisson regression. For $p = 80$, we set $\beta_2 = \beta_3 = 3$, $\beta_4 = \cdots = \beta_{80} = 0$ for Logistic regression and $\beta_2 = \beta_3 = 1$, $\beta_4 = \cdots = \beta_{80} = 0$ for Poisson regression. To simulate the power curve, we take different values of $\beta_1 \in \{10/10, 15/10, \ldots, 6\}$ for Logistic regression and $\beta_1 \in \{10/10, 11/10, \ldots, 2\}$ for Poisson regression .

For $p = 1000$, We will test

$$H_0\text{: } \beta_1 = 1,\ \beta_{11} = 0,\ \beta_{12} = 0 \text{ vs } H_1\text{: } \beta_1 \neq 1 \text{ or } \beta_{11} \neq 0 \text{ or } \beta_{12} \neq 0.$$

and for $p = 80$, we will test

$$H_0\text{: } \beta_1 = 1,\ \beta_4 = 0,\ \beta_5 = 0 \text{ vs } H_1\text{: } \beta_1 \neq 1 \text{ or } \beta_4 \neq 0 \text{ or } \beta_5 \neq 0.$$

Other settings remain the same as Section 2.4.

The empirical size of the simulation for Logistic and Poisson regression with $p = 1000$ is given in Table 2.6 and 2.7 under different choices of Σ. We will see the size is well controlled under different designs.

For the $p = 1000$ case, Figure 2.5 and 2.6 shows the power curves of the tests under different choices of Σ for Logistic regression and Poisson regression, respectively. The size is well controlled when H_0 is true, and $T_1(1, 1)$, $T_1(2, 2)$ and $T_1(\infty, \infty)$ achieves similar power.

For the $p = 80$ case, Figure 2.7 and 2.8 shows the power curves of the tests under different choices of Σ for both Logistic regression and Poisson regression. The size is well controlled and $T_1(1, 1)$ and $T_1(2, 2)$ achieves higher power than $T_1(\infty, \infty)$ as correlation increases.

Table 2.6: Logistic regression empirical size for simultaneous testing under different correlation design with $n = 100$, $p = 1000$.

Design	Method	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
$\Sigma = \mathbf{I}_p$	$T_1(1,1)$	0.158	0.098	0.068	0.016
	$T_1(2,2)$	0.174	0.088	0.042	0.010
	$T_1(\infty,\infty)$	0.168	0.090	0.038	0.014
$\Sigma = (0.5^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.172	0.102	0.056	0.020
	$T_1(2,2)$	0.144	0.072	0.052	0.022
	$T_1(\infty,\infty)$	0.148	0.076	0.042	0.012
$\Sigma = (0.9^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.236	0.176	0.118	0.050
	$T_1(2,2)$	0.264	0.172	0.126	0.062
	$T_1(\infty,\infty)$	0.240	0.126	0.078	0.028
$\Sigma = (0.5^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.088	0.042	0.018	0.008
	$T_1(2,2)$	0.088	0.042	0.020	0.008
	$T_1(\infty,\infty)$	0.086	0.036	0.014	0.008
$\Sigma = (0.8^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.090	0.038	0.024	0.008
	$T_1(2,2)$	0.112	0.048	0.016	0.002
	$T_1(\infty,\infty)$	0.096	0.048	0.024	0.006

Table 2.7: Poisson regression empirical size for simultaneous testing under different correlation design with $n = 100$, $p = 1000$.

Design	Method	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
$\Sigma = \mathbf{I}_p$	$T_1(1,1)$	0.182	0.106	0.058	0.022
	$T_1(2,2)$	0.198	0.096	0.050	0.008
	$T_1(\infty,\infty)$	0.188	0.088	0.038	0.006
$\Sigma = (0.5^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.194	0.102	0.054	0.010
	$T_1(2,2)$	0.248	0.108	0.048	0.010
	$T_1(\infty,\infty)$	0.242	0.124	0.062	0.020
$\Sigma = (0.9^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.264	0.148	0.066	0.022
	$T_1(2,2)$	0.266	0.174	0.120	0.048
	$T_1(\infty,\infty)$	0.246	0.146	0.084	0.036
$\Sigma = (0.5^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.200	0.096	0.050	0.010
	$T_1(2,2)$	0.184	0.076	0.038	0.016
	$T_1(\infty,\infty)$	0.192	0.102	0.062	0.006
$\Sigma = (0.8^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.186	0.096	0.066	0.014
	$T_1(2,2)$	0.180	0.096	0.056	0.008
	$T_1(\infty,\infty)$	0.152	0.076	0.038	0.010

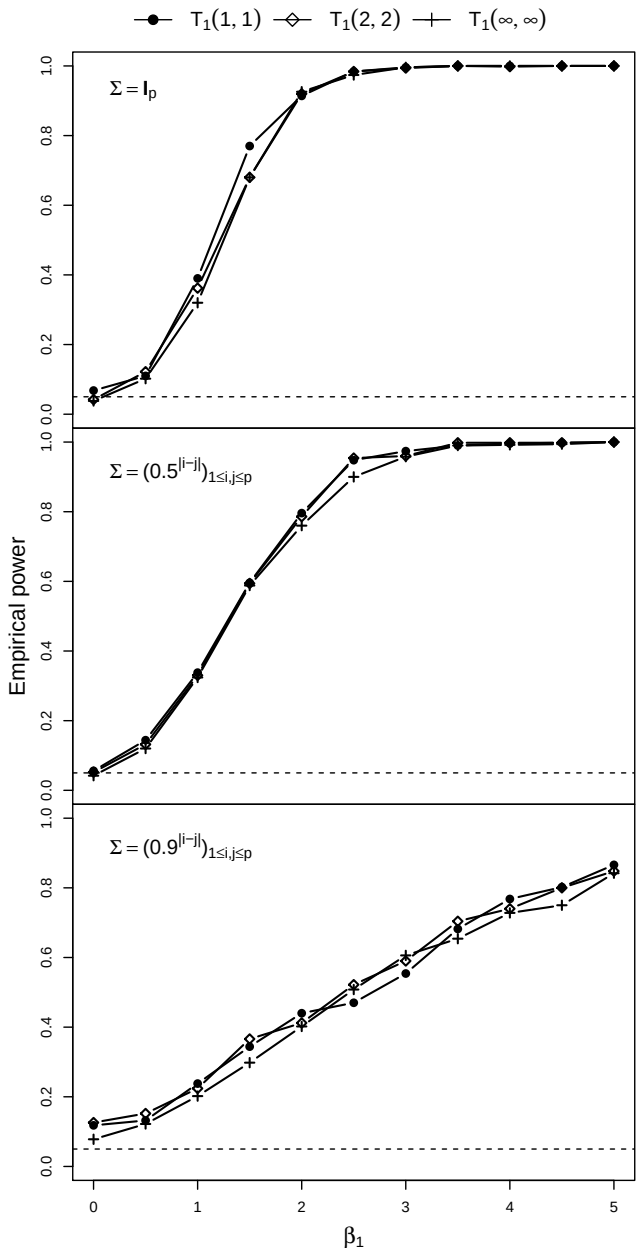

Figure 2.5: Logistic regression empirical power simultaneous testing under different correlation design with $n = 100$, $p = 1000$ (from top to bottom: $\Sigma = \mathbf{I}_p$, $\Sigma = (0.5^{|i-j|})_{1\leq i,j\leq p}$, and $\Sigma = (0.9^{|i-j|})_{1\leq i,j\leq p}$).

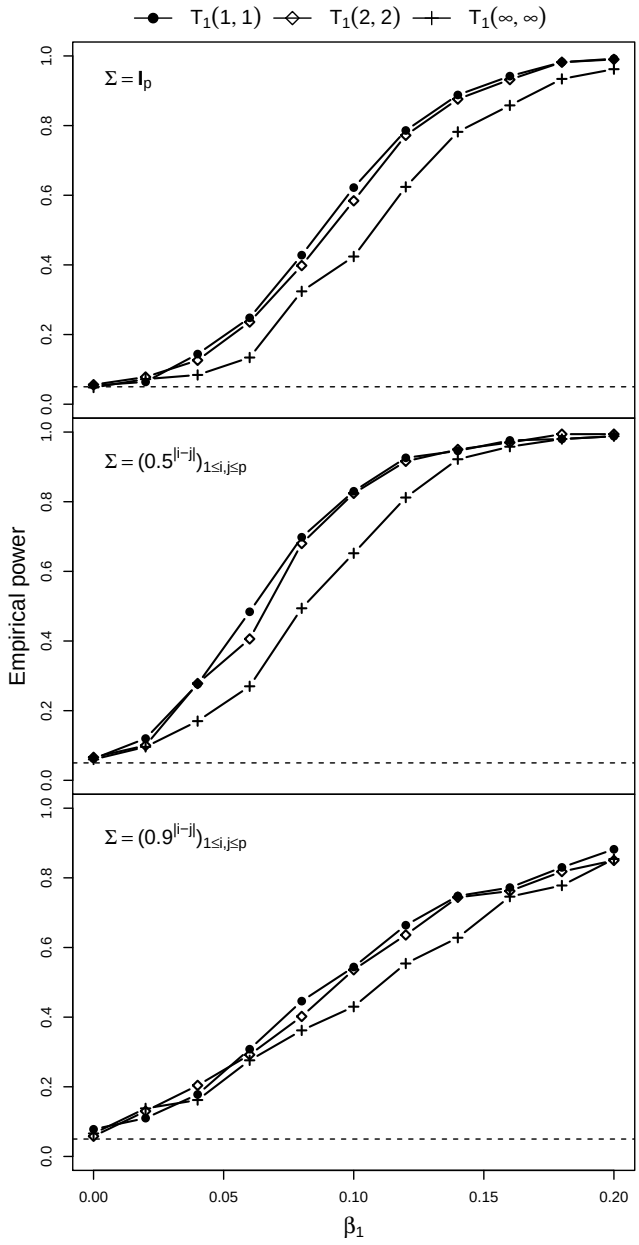

Figure 2.6: Poisson regression empirical power simultaneous testing under different correlation design with $n = 100$, $p = 1000$ (from top to bottom: $\Sigma = \mathbf{I}_p$, $\Sigma = (0.5^{|i-j|})_{1\leq i,j\leq p}$, and $\Sigma = (0.9^{|i-j|})_{1\leq i,j\leq p}$).

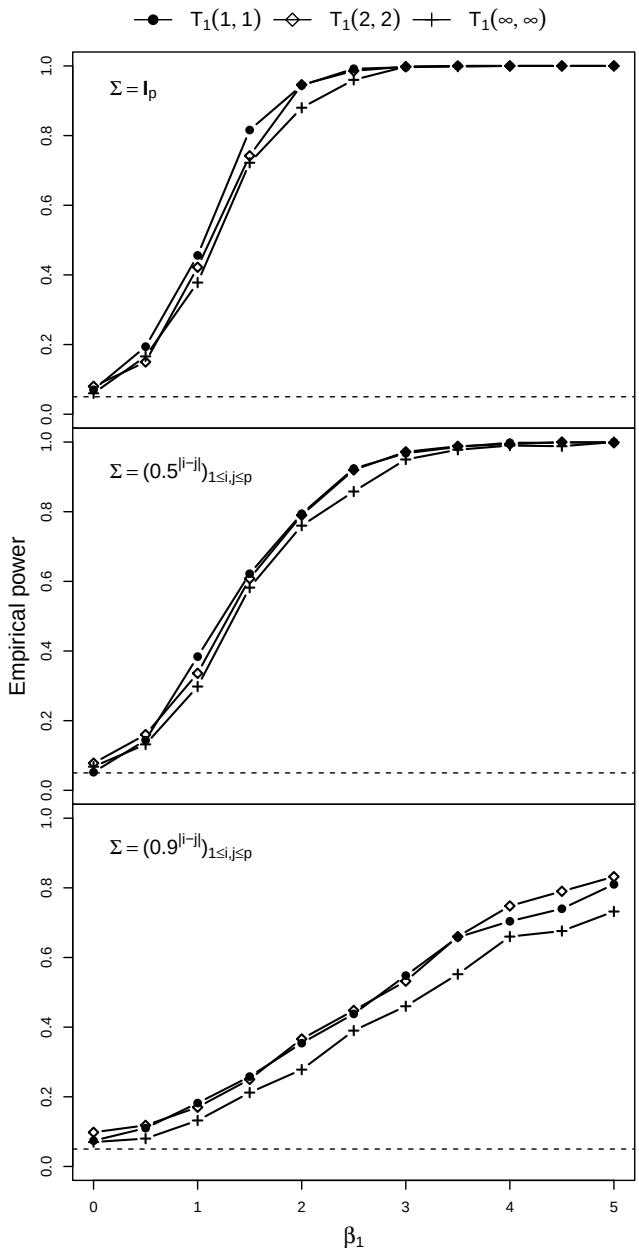

Figure 2.7: Logistic regression empirical power simultaneous testing under different correlation design with $n = 100$, $p = 80$ (from top to bottom: $\Sigma = \mathbf{I}_p$, $\Sigma = (0.5^{|i-j|})_{1\leq i,j\leq p}$, and $\Sigma = (0.9^{|i-j|})_{1\leq i,j\leq p}$).

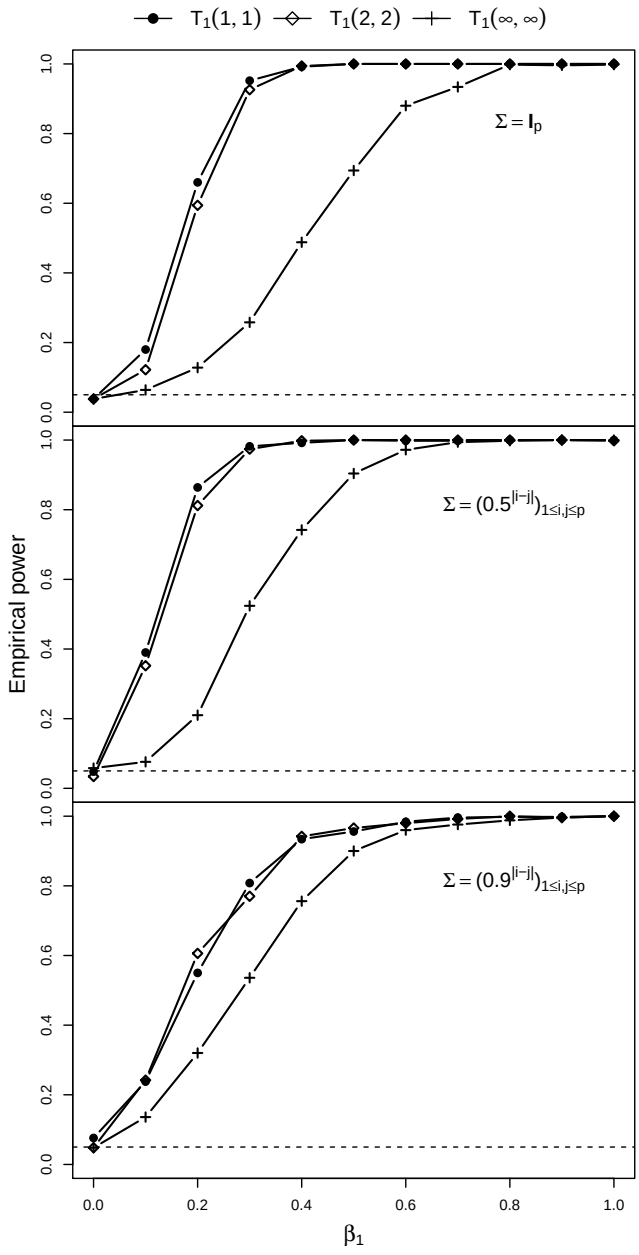

Figure 2.8: Poisson regression empirical power simultaneous testing under different correlation design with $n = 100$, $p = 80$ (from top to bottom: $\Sigma = \mathbf{I}_p$, $\Sigma = (0.5^{|i-j|})_{1\leq i,j\leq p}$, and $\Sigma = (0.9^{|i-j|})_{1\leq i,j\leq p}$).

More general test

In this section, we will consider testing H_0: $\beta_1 = \beta_2$ vs H_1: $\beta_1 \neq \beta_2$ as discussed in Section 2.4. For simplicity, we generate data from the linear regression model

$$Y = \mathbf{X}\beta + \epsilon,$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I}_n)$ with $n = 100$ and $\beta = (\beta_1, \ldots, \beta_p)^T$. For $p = 1000$, we set $\beta_2 = \cdots = \beta_{11} = 1$, $\beta_{12} = \cdots = \beta_{1000} = 0$, and $\beta_1 \in \{1, 11/10, \ldots, 2\}$. For $p = 80$, we set $\beta_2 = \beta_3 = \beta_4 = 1$. Other simulation settings remain the same as in section 2.4. We utilized R package `lars` Hastie and Efron, 2013 to implement our method, as it allows for exact computation of the test statistic, due to piecewise linearity of the LASSO path in the continuous response linear model. We compare it to the classical t-test in lower-dimensional settings.

The empirical size of the simulation for H_0: $\beta_1 = \beta_2$ under $p = 1000$, is given in Table 2.8 under different choices of Σ. It is clear that our method nicely controlled the size under different choices of Σ and different quantities $T_1(1, 1)$, $T_1(2, 2)$ and $T_1(\infty, \infty)$.

The empirical power curves of our test based on the LOCO path statistics $T_1(1, 1)$, $T_1(2, 2)$ and $T_1(\infty, \infty)$ under settings $n = 1000$ and $p = 80$ over the values $\beta_1 \in \{0/10, 1/10, \ldots, 1\}$ are depicted in Figures 2.9 and 2.10. For $p = 80$, we compared it to the classical T-test. For most cases, $T_1(1, 1)$ and $T_1(2, 2)$ have the highest power, while $T_1(\infty, \infty)$ loses a lot of power under the correlated design. Our method achieved much greater power than the T-test under different designs.

Table 2.8: Empirical power for testing $\beta_1 = \beta_2$ under different correlation design with $n = 100$, $p = 1000$.

Design	Method	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
$\Sigma = \mathbf{I}_p$	$T_1(1,1)$	0.224	0.126	0.064	0.010
	$T_1(2,2)$	0.192	0.110	0.060	0.014
	$T_1(\infty,\infty)$	0.216	0.114	0.062	0.020
$\Sigma = (0.5^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.242	0.114	0.052	0.012
	$T_1(2,2)$	0.196	0.082	0.042	0.016
	$T_1(\infty,\infty)$	0.184	0.098	0.052	0.008
$\Sigma = (0.9^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.146	0.076	0.042	0.014
	$T_1(2,2)$	0.172	0.086	0.048	0.016
	$T_1(\infty,\infty)$	0.206	0.094	0.042	0.014
$\Sigma = (0.5^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.150	0.078	0.036	0.010
	$T_1(2,2)$	0.176	0.082	0.042	0.008
	$T_1(\infty,\infty)$	0.142	0.076	0.042	0.012
$\Sigma = (0.8^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.132	0.062	0.036	0.022
	$T_1(2,2)$	0.130	0.076	0.052	0.028
	$T_1(\infty,\infty)$	0.134	0.082	0.058	0.028

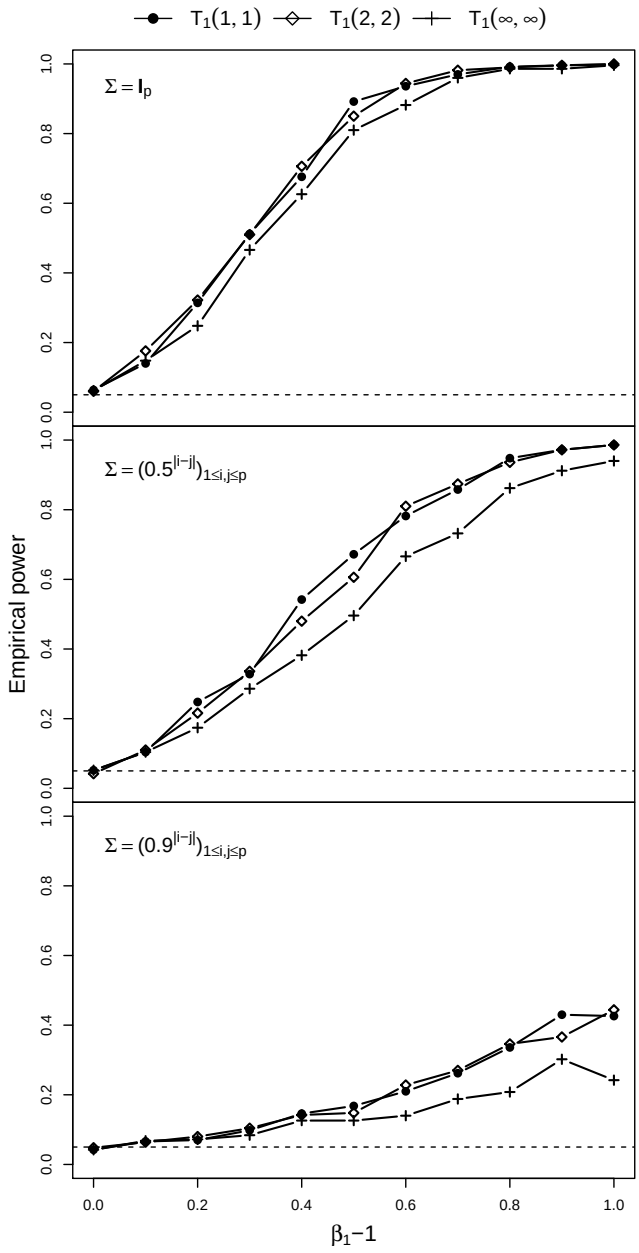

Figure 2.9: Empirical power for testing $\beta_1 = \beta_2$ under different correlation design with $n = 100$, $p = 1000$ (from top to bottom: $\Sigma = \mathbf{I}_p$, $\Sigma = (0.5^{|i-j|})_{1\leq i,j\leq p}$, and $\Sigma = (0.9^{|i-j|})_{1\leq i,j\leq p}$).

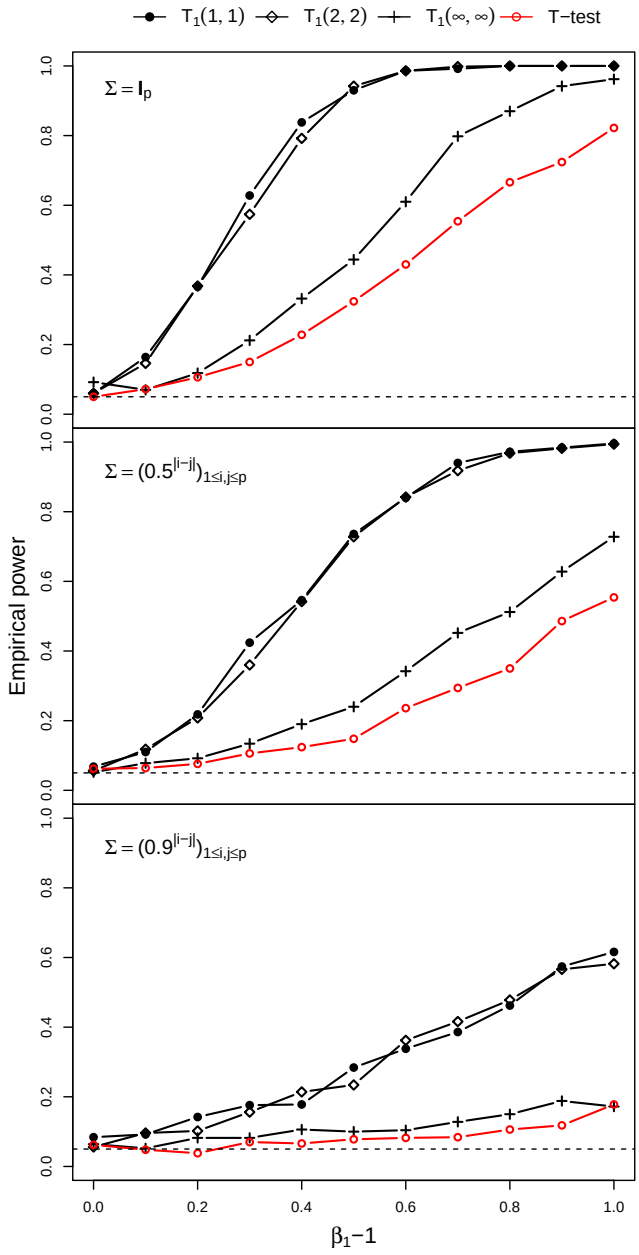

Figure 2.10: Empirical power for testing $\beta_1 = \beta_2$ under different correlation design with $n = 100$, $p = 80$ (from top to bottom: $\Sigma = \mathbf{I}_p$, $\Sigma = (0.5^{|i-j|})_{1\leq i,j\leq p}$, and $\Sigma = (0.9^{|i-j|})_{1\leq i,j\leq p}$).

2.5 Real data analysis

To present some applications of our method, we consider several publicly available datasets and explored whether our method could find some important genes. We will

first do variable screening and apply our inference procedure afterwards. We will also apply desparsified LASSO on these datasets for comparison.

Leukemia: The Leukemia dataset Golub et al., 1999 was obtained from Affymetrix oligonucleotide microarrays, which measured gene expression levels for $n = 72$ patients. Each patients are either suffering from acute lymphoblastic leukemia(ALL) or acute myeloid leukemia (AML). This dataset does not have a control group, just two type of patient samples. And we have $p = 7129$ genes measurements for each patient.

Prostate: The Prostate tumor dataset Singh et al., 2002 measured gene expression levels for $n = 102$ patients who are suffering from prostatectomy. Sample of each patients are either classified as normal or tumor. And we have $p = 12600$ genes measurements for each patient.

Colon: The Colon cancer dataset Alon et al., 1999 collected, measured and selected $p = 2000$ gene expression levels for 40 tumor and 22 normal colon tissues($n = 62$).

Lymphoma: This dataset Dudoit et al., 2002 measured genes expression levels for $n = 62$ samples and yield $p = 4026$ gene expressions. Among these 62 samples, we have 42 samples of diffuse large B-cell lymphoma, 9 samples of follicular lymphoma and 11 samples of chronic lymphocytic leukemia. This dataset does not have a control group as well, just three type of patient samples. And we merged follicular lymphoma and chronic lymphocytic leukemia as one category to make this dataset more balanced.

All these datasets were normalized on the log scale. We use $T(1, 1)$ in this part and obtained bootstrap P-values for each gene after variable screening. The variable screening procedure reduced the dimension of dataset from 7129 to 41 from Leukemia dataset.

For Leukemia dataset, Colon dataset and Lymphoma dataset, neither our method nor the desparsified LASSO found any significant genes. For the Prostate datasets,

our method found X83543 and X07732 significant with P-value 0.0014 and P-value < 0.0001 while the desparsified LASSO only found X07732 with P-value 0.003. And we found X07732 is an external transcribed spacer, a type of non-coding mRNA. And X83543 corresponds to APXL gene for human being.

2.6 Discussion

We extend the LOCO path statistic to generalized linear models and more general hypothesis testing scenario. For variable screening, our method does not require the selection of tuning parameters and can achieve a greater probability of selecting a set of covariates that contains the true model than both SIS and ISIS.

For statistical inference, our method provides reliable P-values in both high and lower-dimensional settings. Overall, the proposed bootstrap method controls the size and in some cases achieves higher power than the desparsified LASSO Geer et al., 2014. Moreover, our method can be used to test hypothesis simultaneously involving multiple coefficients.

We use simulated results to show the effectiveness of our method. We also proved the effectiveness of our method under some simple settings. Rigorous proof of the consistency of our bootstrap procedure requires deep understanding of the behavior of the solution path in both lower and high-dimensional case, which is worth our future investigation.

Chapter 3

Extensions of the Leave-One-Covariate-Out solution path statistic to sparse Gaussian graphical models

3.1 Introduction

We consider Gaussian graphical models, under which

$$\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(n)} \,\mathrm{i.\,i.\,d} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}),$$

where $\boldsymbol{x}_{(i)} \in \mathbb{R}^p$, $i \in 1, \ldots, n$. If Σ is non-singular, an interesting descriptor of the p-dimensional Gaussian distribution is a graph with p nodes and a set of undirected edges, connecting pairs of variables with non-zero conditional covariates, after accounting for the effects of the remaining variables.

Between any two different nodes j and k, there is an undirected edge if and only if $\Sigma^{-1}_{jk} \neq 0$. An interpretation is that: $\boldsymbol{x}_{(j)}$ and $\boldsymbol{x}_{(k)}$ are conditionally dependent given all other variables $\boldsymbol{x}_{(i),i\neq j,k}$ if and only if $\Sigma^{-1}_{jk} \neq 0$. Hence, it is of interest to estimate Σ^{-1}, which is also known as the precision matrix, if we want to learn the graphical structure of these p nodes. This can also be viewed as model selection in Gaussian graphical models or covariance selection Dempster, 1972. Forward or backward stepwise selection are considered standard methods for covariance selection Yuan and Lin, 2007. However, these methods suffer from computational issues and fail to adjust for multiple comparisons Edwards, 2012. The problem especially challenging in the high-dimensional case $p > n$.

For high-dimensional linear regression, LASSO Tibshirani, 1996 and other LASSO based regularization techniques Zou and Hastie, 2005, Zou, 2006 have proved to be successful in variable selection. The graphical LASSO introduced regularization to the graphical models and maximized the penalized log-likelihood

$$\log\det\Theta - \mathrm{tr}(S\Theta) - \rho\|\Theta\|_1,$$

where $\|\Theta\|_1 = \sum_{i,j}|\Theta_{ij}|$. It was first proposed by Meinshausen, Bühlmann, et al. Meinshausen, Bühlmann, et al., 2006, which uses neighborhood selection for each node. Different algorithms Banerjee et al., 2008, Yuan and Lin, 2007 are proposed by solving the dual problem of the graphical LASSO. The glasso algorithm Friedman et al., 2008 borrowed from Yuan and Lin, 2007 and Banerjee et al., 2008 proposed to use a block coordinate descent algorithm, which is very efficient in solving the graphical LASSO problem. Others proposed modifications to glasso Mazumder and Hastie, 2012a Mazumder and Hastie, 2012b to address the convergence issues and improve its performance in large-scale networks. Guo et al. Guo et al., 2011 further extended to joint estimation of multiple graphical models.

However, there is not very much work focusing on the information contained in the LASSO solution path for graphical models. The Leave-One-Covariate-Out(LOCO) solution path was proposed to provide variable importance measurement and statistical inference for linear and generalized linear models. The LOCO solution path considers measuring the change in LASSO solution path due to removal of one covariate from the model. LOCO path compares the full solution path

$$\hat{\beta} := \hat{\beta}(\lambda) = \underset{\beta\in\mathbb{R}^p}{\mathrm{argmin}}\, L(Y, \mathbf{X}\beta) + \lambda J(\beta), \quad \lambda > 0$$

to the LOCO solution path, given by

$$\hat{\beta}^{(-j)} := \hat{\beta}^{(-j)}(\lambda) = \underset{\beta\in\mathbb{R}^p, \beta_j=0}{\mathrm{argmin}}\, L(Y, \mathbf{X}\beta) + \lambda J(\beta), \quad \lambda > 0,$$

where $L(Y, \mathbf{X}\beta)$ is a loss function and $J(\beta)$ is a penalty function. The LOCO solution path is the regularization soluton path when covariate X_j is removed. For non-

zero coefficients its importance will be reflected by a large difference between two paths. Bootstrap procedures are provided for statistical inference. In this paper, we further extend the Leave-One-Covariate-Out solution path to graphical LASSO estimators in order to measure the importance of an edge in the graph by comparing the glasso solution path to a glasso solution path in which that edge is not allowed to enter the path. This paper is organized as follows: Section 3.2 defines the LOCO path statistic for graphical models and Section 3.3 presents simulation results and Section 3.4 provides some real data examples. Section 3.5 presents some additional discussions. Section 3.6 provides additional discussion.

3.2 Methodology

We first review the glasso algorithm. Suppose we have n i.i.d. samples $\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(n)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{x}_{(i)} \in \mathbb{R}^p$, $i \in 1, \ldots, n$. Let $\Theta = \Sigma^{-1}$ and let S be the empirical covariance matrix. To estimate the inverse covariance matrix, we want to maximize the penalized log-likelihood

$$\hat{\Theta} = \underset{\Theta \succ 0}{\operatorname{argmax}} \left\{ \log\det\Theta - \operatorname{tr}(S\Theta) - \rho\|\Theta\|_1 \right\}. \tag{3.1}$$

The glasso algorithm Friedman et al., 2008 is proposed to maximize (3.1) by iteratively solving a LASSO problem Banerjee et al., 2008

$$\arg\max_{\beta} \left\{ \frac{1}{2} \left\| W_{11}^{1/2}\beta - b \right\|^2 + \rho\|\beta\|_1 \right\}, \tag{3.2}$$

where $b = W_{11}^{-1/2} s_{12}$, W is the current estimate of Σ, and W_{11} and s_{12} are obtained by partitioning W and S as

$$W = \begin{pmatrix} W_{11} & w_{12} \\ w_{12}^T & w_{22} \end{pmatrix}, \quad S = \begin{pmatrix} S_{11} & s_{12} \\ s_{12}^T & s_{22} \end{pmatrix}.$$

The value $w_{12} = W_{11}\beta$ gives the solution for w_{12}. This means we can solve (3.1) in the manner of a block coordinate descent algorithm.

Now we put it under the framework of the Leave-One-Covariate-Out solution path. The graphical LASSO estimator (3.1) can be viewed as mapping from $(0, \infty)$ to the space of positive definite matrices. But in this case, since the matrix is symmetric, we can restrict our attention to the upper triangle of Θ, which measures the correlation between different variables. To measure the importance of the edge corresponding to Θ_{kl}, we consider the solution path for the upper triangle entry

$$\hat{\Theta}_{ij,i<j} = \underset{\Theta \succ 0}{\operatorname{argmax}} \left\{ \log\det\Theta - \operatorname{tr}(S\Theta) - \rho\|\Theta\|_1 \right\}. \tag{3.3}$$

For the LOCO path, we consider the constrained solution path

$$\hat{\Theta}^{-(k,l)}_{ij,i<j} = \underset{\Theta \succ 0, \Theta_{kl}=0}{\operatorname{argmax}} \left\{ \log\det\Theta - \operatorname{tr}(S\Theta) - \rho\|\Theta\|_1 \right\}, \tag{3.4}$$

in which the edge connecting variables k and l is never allowed to enter.

To solve the constrained optimization problem, we propose a simple modification of (3.4), by imposing an infinite penalty on Θ_{kl} and solving

$$\hat{\Theta}^{-(k,l)}_{ij,i<j} = \underset{\Theta \succ 0}{\operatorname{argmax}} \left\{ \log\det\Theta - \operatorname{tr}(S\Theta) - \rho \sum_{i \neq k, j \neq l} |\Theta_{ij}| - \gamma|\Theta_{kl}| \right\}, \tag{3.5}$$

where $\gamma \to \infty$. In practice, we can just fix γ to be some very large number.

And then we measure the change in the solution path due to imposing $\Theta_{kl} = 0$ by computing

$$T_{k,l}(q) = ||\hat{\Theta}_{ij,i<j} - \hat{\Theta}^{-(k,l)}_{ij,i<j}||_{q,q},$$

where for a function $f : [0, \infty) \to \mathbb{R}^p$ with each entry $f_j : [0, \infty) \to \mathbb{R}$, for $j = 1, \ldots, p$,

$$\|f\|_{q,q} = \begin{cases} \left(\sum_{k=1}^p \int_0^\infty |f_k(\lambda)|^q d\lambda\right)^{1/q}, & 0 < q < \infty \\ \max\limits_{1 \le k \le p} \sup\limits_{\lambda > 0} |f_k(\lambda)|, & q = \infty \end{cases}.$$

Note this is the same quantity from Chapter 1.

Variable screening for precision matrix

Our statistic $T_{k,l}(q)$ serves as a variable importance measure for Θ_{kl}. We can further use it for variable screening.

We first standardize $T_{k,l}(q)$, $k < l, k = 1, \ldots, p$ and $l = 2, \ldots, p$ by setting

$$\bar{T}_{k,l}(q) = T_{k,l}(q) \left(\sum_{k<l} T_{k,l}(q) \right)^{-1}.$$

For variable screening, we set a threshold ϵ and only screen in entry kl if the precision matrix $\bar{T}_{k,l}(q) > \epsilon$. Or we can rank $\bar{T}_{k,l}(q)$ and only screen the component of Θ in the top K entries. In practice, ϵ can be the 95% quantile of $\bar{T}_{k,l}(q)$ so you only screen in 5% variables. And K can be 100 or depends on how many entries of Θ we want to be non-zero.

Computational details

For linear regression models, the LASSO solution path is piecewise linear Rosset and Zhu, 2007. For the graphical LASSO, the solution path is not piecewise linear. Hence we cannot compute $T_{k,l}(q)$ exactly. But we can approximate $T_{k,l}(q)$ by specifying a grid of λ.

To create the grid of λ, we construct a sequence of λ values decreasing from some pre-determined $\lambda_{\max}$ to $\lambda_{\min}$ with $\lambda_{\max}$ determined by $\max_{i,j} |S|$, where S is the sample covariance matrix Friedman et al., 2019. And λ_{min} is determined by λ_{max}/K, where K is the length of the grid. In practice, we can use $K = 50$ and it usually provides good approximation.

To solve (3.5), we utilized the R package `glasso` Friedman et al., 2019, which allows the users to impose different penalties for each entry of Θ. We may not be able to set $\gamma = \infty$, but in practice settings we have found that setting $\gamma = 10^4$ usually works well.

We can also measuring the importance of some entry of Σ, the covariance matrix, using the LOCO path idea. Although this may not be related to graphical models, it is interesting to show the adeptness of our LOCO path idea.

Suppose we want to know whether some entry of $\Sigma_{kl} = 0$. Then we compare (3.3) to the constrained solution path

$$\hat{\Theta}^{-(k,l)}_{ij,i<j} = \underset{\Theta \succ 0, \Sigma_{kl}=0}{\operatorname{argmax}} \left\{ \log\det\Theta - \operatorname{tr}(S\Theta) - \rho\|\Theta\|_1 \right\}. \tag{3.6}$$

To solve (3.6), we propose a modified glasso algorithm. To measure the importance of Σ_{kl}, we remove the corresponding rows and columns from W and S in the block coordinate descent loop, if the loop updates $\hat{\Sigma}_{kl}$. And then we solve (3.2) and update $w_{12} = W_{11}\beta$. By doing so, we can continue updating the w_{12} entry of W and at the same time, we can impose the constraint $\Sigma_{kl} = 0$ in the coordinate descent loop.

Here is the algorithm in detail:

1. Set $W = S + \rho I$. The diagonal of W remains unchanged in what follows.

2. For each $j = 1, 2, \ldots p, 1, 2, \ldots p, \ldots$, solve the lasso problem (3.2) which takes as input the inner products W_{11} and s_{12}. If a loop updates $\hat{\Sigma}_{kl}$, we remove the corresponding rows and columns from W_{11} and s_{12} . This gives a $p-2$ vector solution $\hat{\beta}^{-(kl)}$. For the entries of $\hat{\beta}$ which will update $\hat{\Sigma}_{kl}$, we will set it equal to 0. All other entries we set as $\hat{\beta} = \hat{\beta}^{-(kl)}$. Hence we have a $p-1$ vector solution $\hat{\beta}$. We then fill in the corresponding row and column of W using $w_{12} = W_{11}\hat{\beta}$. By doing so, W_{kl} will always be 0.

3. Continue until convergence.

And then we measure the change in the solution path

$$T_{k,l}(q) = ||\hat{\Theta}_{ij,i<j} - \hat{\Theta}^{-(k,l)}_{ij,i<j}||_q,$$

which will give us a measure of importance for Σ_{kl}.

3.3 Simulation results

We now study via simulation the effectiveness of the LOCO path statistic as a tool in recovering the dependence structure in Gaussian graphical models.

We generate data from the following model

$$\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(n)} \,\mathrm{i.\,i.\,d} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}),$$

where $p = 50$ and $n = 100$ and 1000. We considered small p since large p will be computationally expensive. We follow similar settings in Luo et al., 2014 and consider two types of precision matrix, given as simulation A and simulation C in that paper. The data generation procedure are the same as in Luo et al., 2014. We will calculate $\bar{T}_{k,l}(2)$ in the variable screening and keep all edges between variable pairs for which $\bar{T}_{k,l}(2) > \epsilon$, where we vary ϵ in order to produce an ROC curve. We compare our procedure to the GRASS algorithm Luo et al., 2014. We simulate 250 datasets for each model. And we will record the false positive rate and true positive rate among 250 simulations.

To generate the precision matrix, we consider two steps. First we generate two types of edge set $\mathcal{E}$. For random graph, we set $(i, j) \in \mathcal{E}$ with probability 0.01 for all $i < j$. This corresponds to the simulation A in Luo et al., 2014. For a non-random graph, we set $(i, j) \in \mathcal{E}$ for all $|i - j| \leq 2$ and $(i, j) \notin \mathcal{E}$ for all $|i - j| > 2$. And this corresponds to the simulation C in Luo et al., 2014. In the second step, we generate a $p \times p$ matrix $\mathbf{A}$, where

$$A_{ij} = A_{ji} = \begin{cases} 1 & \text{for } i = j \\ \text{Unif } [-0.3, 0.7] & \text{for } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

Then we create a positive definite matrix $\boldsymbol{\Sigma}^{-1} = \mathbf{A} + (0.1 - \lambda_{\min}(\mathbf{A}))\,\mathbf{I}$, where $\lambda_{\min}(\mathbf{A})$ gives the smallest eigenvalue of $\mathbf{A}$.

Figures 3.1 and 3.2 present ROC curves for our LOCO path procedure and the GRASS algorithm under different settings with $n = 100$ and $n = 1000$. For $n = 100$, the performance of our procedure and the GRASS is pretty similar, regardless of different type of precision matrix. For $n = 1000$, our method has higher AUC than the GRASS for precision matrix type A. And for type C, the performance between them are also very close.

Figure 3.3 provides another example comparing glasso estimates to our LOCO path statistic. And we consider $\Theta = \Sigma^{-1}$ as $\Theta_{i,i+10} = s$, $\Theta_{i,i} = 1$. Our method (without screening) captured all non-zero entries while the glasso marks irrelevant entries as important.

For variable importance calculation of covariance matrix, we consider identical settings except for the fact that we are estimating Σ now. Figure 3.4 provides an example comparing our LOCO path statistic to true covariance matrix. Our method (without screening) captured all non-zero entries of Σ.

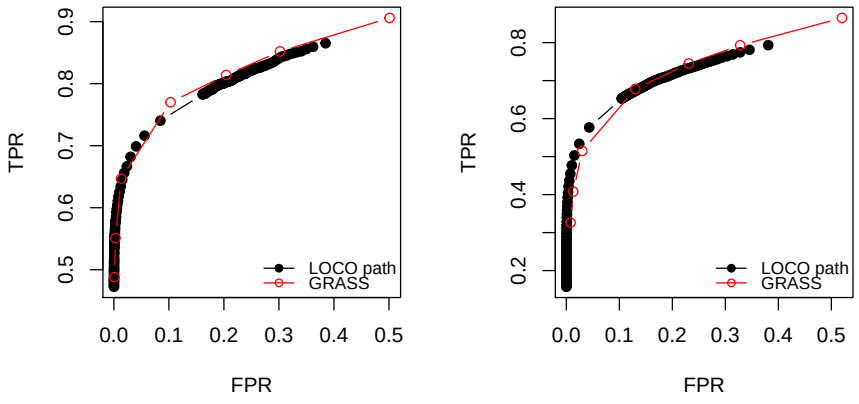

Figure 3.1: ROC curve comparison between LOCO path and GRASS for $n = 100$. Left: simulation A. Right: simulation C.

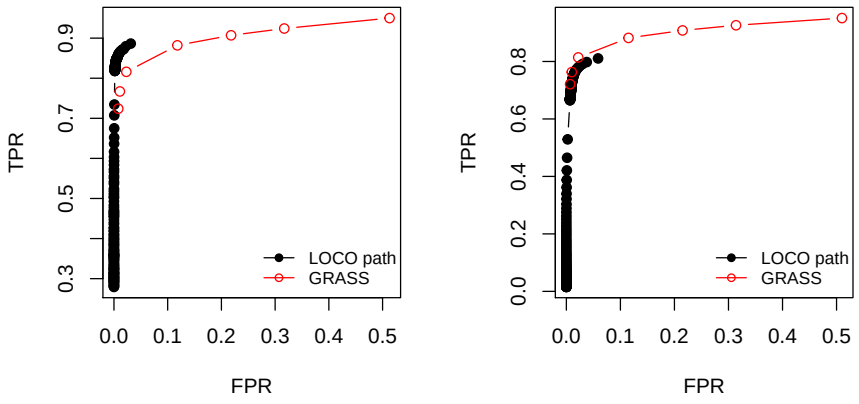

Figure 3.2: ROC curve comparison between LOCO path and GRASS for $n = 1000$. Left: simulation A. Right: simulation C.

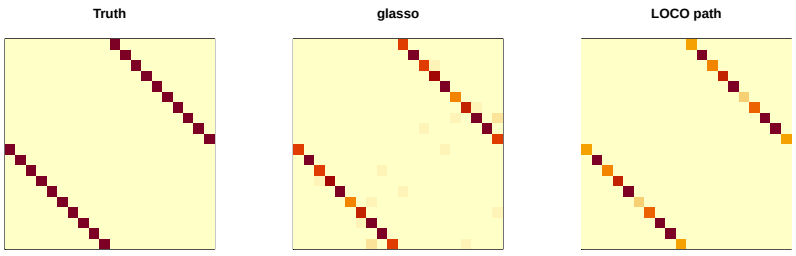

Figure 3.3: Comparison between the glasso estimates and our LOCO path statistic for estimating precision matrix. Left: truth. Middle: glasso. Right: LOCO path statistic.

3.4 Real data analysis

To further present the application of our method, we consider two publicly available datasets and explored whether our method could find some interesting graphical structure.

We consider the flow cytometry dataset Sachs et al., 2005 which was also analyzed in Friedman et al., 2008. This dataset contains $p = 11$ proteins and $n = 7466$ cells. The results of applying our LOCO path statistic to this data set is shown in

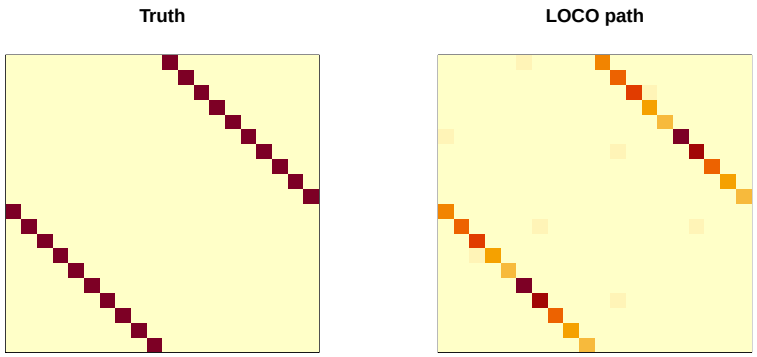

Figure 3.4: LOCO path variable importance of covariance matrix, compared to the truth. Left: truth. Right: LOCO path statistic.

Figure 3.5. We calculate $\bar{T}_{k,l}(2)$ in the variable screening step and we considered 4 different screening threshold ϵ, which is determined by different quantiles of $\bar{T}_{k,l}(2)$, $k = 1, \ldots, 10$, $l = 2, \ldots, 11$ and $k < l$. Figure 3.6 shows the LOCO path statistic calculated for all variable pairs. We will see only a few pairs have large variable importance, while most variable pairs have variable importance close to 0.

We also consider the riboflavin dataset with 71 observations and 4088 variables, which measures gene expression of Bacillus subtilis Bühlmann et al., 2014. And as discussed in Bühlmann et al., 2014, we will also analyze a smaller version of it (riboflavinV100), which only consider the top 100 genes with largest empirical variance. For this dataset, We calculate $\bar{T}_{k,l}(2)$ in the variable screening step and we considered 4 different screening quantile q. Figure 3.7 shows the screening results. There is some agreement between the graph for screening quantile $q = 0.95$ and the graph presented in Bühlmann et al., 2014. Figure 3.8 shows the LOCO path statistic calculated for the top 200 most important variable pairs.

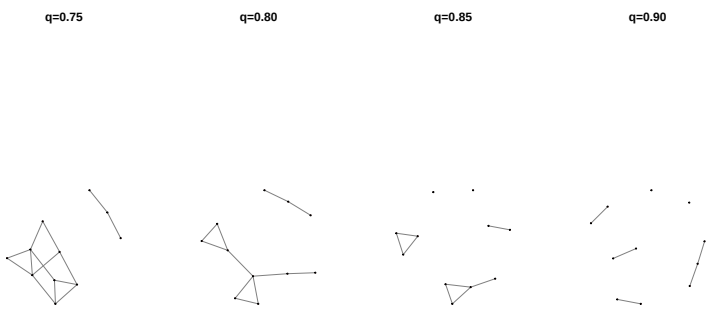

Figure 3.5: Flow cytometry dataset: undirected graph from LOCO path statistic with different values of quantile q.

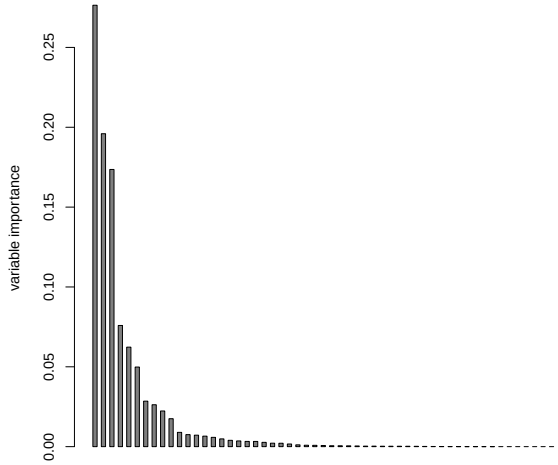

Figure 3.6: Flow cytometry dataset: LOCO path statistic for all variable pairs.

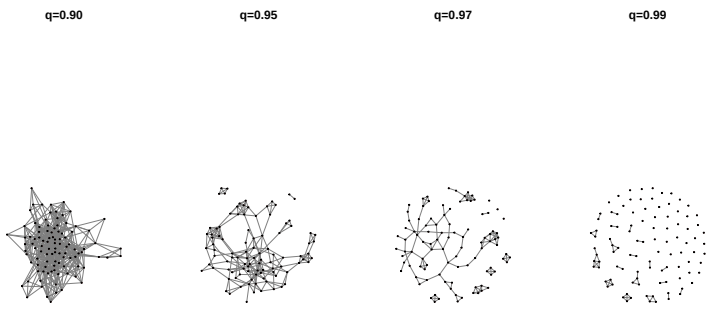

Figure 3.7: RiboflavinV100 dataset: undirected graph from LOCO path statistic with different values of quantile q

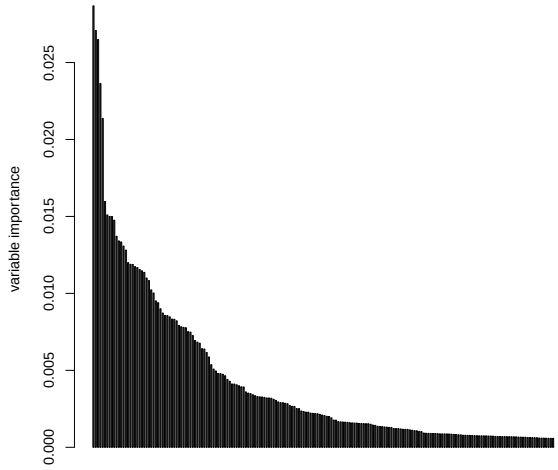

Figure 3.8: RiboflavinV100 dataset: LOCO path statistic for the top 200 variable pairs.

3.5 Discussion

We extend the LOCO path statistic to Gaussian graphical models. For estimating the sparse precision matrix, our method does not require the selection of tuning

parameters and can achieve a greater probability of finding the true dependence structure than the graphical lasso algorithm and the GRASS algorithm.

However, we only consider the calculation of variable importance measures using the LOCO path idea. Valid statistical inference procedures should be considered for our method. It may be interesting to test H_0: $\Theta_{ij} = 0$ v.s H_1: $\Theta_{ij} \neq 0$ where $i \neq j$. Although it may be hard to provide solid theoretical justification, a well designed bootstrap procedure is worth our further investigation.

Bibliography

Alon, Uri, Barkai, Naama, Notterman, Daniel A, Gish, Kurt, Ybarra, Suzanne, Mack, Daniel, and Levine, Arnold J (1999). "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays". In: *Proceedings of the National Academy of Sciences* 96.12, pp. 6745–6750.

Banerjee, Onureena, Ghaoui, Laurent El, and d'Aspremont, Alexandre (2008). "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data". In: *Journal of Machine learning research* 9.Mar, pp. 485–516.

Breiman, Leo (2001). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

Bühlmann, Peter, Kalisch, Markus, and Meier, Lukas (2014). "High-dimensional statistics with a view toward applications in biology". In: *Computational Statistics* 29, pp. 407–430.

Chatterjee, Arindam, Lahiri, Soumendra N, et al. (2013). "Rates of convergence of the adaptive LASSO estimators to the oracle distribution and higher order refinements by the bootstrap". In: *The Annals of Statistics* 41.3, pp. 1232–1259.

Chatterjee, Arindam and Lahiri, Soumendra Nath (2011). "Bootstrapping lasso estimators". In: *Journal of the American Statistical Association* 106.494, pp. 608–625.

Das, Debraj, Gregory, Karl, Lahiri, SN, et al. (2019). "Perturbation bootstrap in adaptive lasso". In: *The Annals of Statistics* 47.4, pp. 2080–2116.

Dempster, Arthur P (1972). "Covariance selection". In: *Biometrics*, pp. 157–175.

Dezeure, Ruben, Bühlmann, Peter, Meier, Lukas, and Meinshausen, Nicolai (2015a). "High-Dimensional Inference: Confidence Intervals, p-values and R-Software hdi". In: *Statistical Science* 30.4, pp. 533–558.

Dezeure, Ruben, Bühlmann, Peter, Meier, Lukas, Meinshausen, Nicolai, et al. (2015b). "High-dimensional inference: Confidence intervals, p-values and R-software hdi". In: *Statistical Science* 30.4, pp. 533–558.

Dudoit, Sandrine, Fridlyand, Jane, and Speed, Terence P (2002). "Comparison of discrimination methods for the classification of tumors using gene expression data". In: *Journal of the American statistical association* 97.457, pp. 77–87.

Edwards, David (2012). *Introduction to graphical modelling*. Springer Science & Business Media.

Fan, Jianqing, Feng, Yang, and Song, Rui (2011). "Nonparametric independence screening in sparse ultra-high-dimensional additive models". In: *Journal of the American Statistical Association* 106.494, pp. 544–557.

Fan, Jianqing and Lv, Jinchi (2008). "Sure independence screening for ultrahigh dimensional feature space". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70.5, pp. 849–911.

Fan, Jianqing and Lv, Jinchi (2010). "A selective overview of variable selection in high dimensional feature space". In: *Statistica Sinica* 20.1, p. 101.

Fan, Jianqing, Song, Rui, et al. (2010). "Sure independence screening in generalized linear models with NP-dimensionality". In: *The Annals of Statistics* 38.6, pp. 3567–3604.

Fisher, Aaron, Rudin, Cynthia, and Dominici, Francesca (2018). "All Models are Wrong but many are Useful: Variable Importance for Black-Box, Proprietary, or Misspecified Prediction Models, using Model Class Reliance". In: *arXiv preprint arXiv:1801.01489*.

Friedman, Jerome, Hastie, Trevor, and Tibshirani, Rob (2019). *glasso: Graphical Lasso: Estimation of Gaussian Graphical Models*. R package version 1.11. URL: `https://CRAN.R-project.org/package=glasso`.

Friedman, Jerome, Hastie, Trevor, and Tibshirani, Robert (2008). "Sparse inverse covariance estimation with the graphical lasso". In: *Biostatistics* 9.3, pp. 432–441.

Friedman, Jerome, Hastie, Trevor, and Tibshirani, Robert (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software* 33.1, pp. 1–22.

Geer, Sara Van de, Bühlmann, Peter, Ritov, Ya'acov, Dezeure, Ruben, et al. (2014). "On asymptotically optimal confidence regions and tests for high-dimensional models". In: *The Annals of Statistics* 42.3, pp. 1166–1202.

Golub, Todd R, Slonim, Donna K, Tamayo, Pablo, Huard, Christine, Gaasenbeek, Michelle, Mesirov, Jill P, Coller, Hilary, Loh, Mignon L, Downing, James R, Caligiuri, Mark A, et al. (1999). "Molecular classification of cancer: class dis-

covery and class prediction by gene expression monitoring". In: *science* 286.5439, pp. 531–537.

Guo, Jian, Levina, Elizaveta, Michailidis, George, and Zhu, Ji (2011). "Joint estimation of multiple graphical models". In: *Biometrika* 98.1, pp. 1–15.

Hastie, Trevor and Efron, Brad (2013). *lars: Least Angle Regression, Lasso and Forward Stagewise*. R package version 1.2. URL: `https://CRAN.R-project.org/package=lars`.

Javanmard, Adel and Montanari, Andrea (2014). "Confidence intervals and hypothesis testing for high-dimensional regression". In: *The Journal of Machine Learning Research* 15.1, pp. 2869–2909.

Ke, Tracy, Jin, Jiashun, and Fan, Jianqing (2014). "Covariance assisted screening and estimation". In: *Annals of Statistics* 42.6, pp. 2202–2242.

Lei, Jing, G'Sell, Max, Rinaldo, Alessandro, Tibshirani, Ryan J, and Wasserman, Larry (2018). "Distribution-free predictive inference for regression". In: *Journal of the American Statistical Association* 113.523, pp. 1094–1111.

Lockhart, Richard, Taylor, Jonathan, Tibshirani, Ryan J, and Tibshirani, Robert (2014). "A significance test for the lasso". In: *Annals of Statistics* 42.2, pp. 413–468.

Luo, Shikai, Song, Rui, and Witten, Daniela (2014). "Sure screening for Gaussian graphical models". In: *arXiv preprint arXiv:1407.7819*.

Mammen, Enno (2012). *When does bootstrap work?: asymptotic results and simulations*. Vol. 77. Springer Science & Business Media.

Mazumder, Rahul and Hastie, Trevor (2012a). "Exact covariance thresholding into connected components for large-scale graphical lasso". In: *Journal of Machine Learning Research* 13.Mar, pp. 781–794.

Mazumder, Rahul and Hastie, Trevor (2012b). "The graphical lasso: New insights and alternatives". In: *Electronic journal of statistics* 6, p. 2125.

Meinshausen, Nicolai, Bühlmann, Peter, et al. (2006). "High-dimensional graphs and variable selection with the lasso". In: *The annals of statistics* 34.3, pp. 1436–1462.

Meinshausen, Nicolai, Meier, Lukas, and Bühlmann, Peter (2009). "P-values for high-dimensional regression". In: *Journal of the American Statistical Association* 104.488, pp. 1671–1681.

Rosset, Saharon and Zhu, Ji (2007). "Piecewise linear regularized solution paths". In: *The Annals of Statistics*, pp. 1012–1030.

Sachs, Karen, Perez, Omar, Pe'er, Dana, Lauffenburger, Douglas A, and Nolan, Garry P (2005). "Causal protein-signaling networks derived from multiparameter single-cell data". In: *Science* 308.5721, pp. 523–529.

Saldana, Diego Franco and Feng, Yang (2018). "SIS: An R Package for Sure Independence Screening in Ultrahigh-Dimensional Statistical Models". In: *Journal of Statistical Software* 83.2, pp. 1–25. DOI: 10.18637/jss.v083.i02.

Singh, Dinesh, Febbo, Phillip G, Ross, Kenneth, Jackson, Donald G, Manola, Judith, Ladd, Christine, Tamayo, Pablo, Renshaw, Andrew A, D'Amico, Anthony V, Richie, Jerome P, et al. (2002). "Gene expression correlates of clinical prostate cancer behavior". In: *Cancer cell* 1.2, pp. 203–209.

Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1, pp. 267–288.

Tibshirani, Robert (1997). "The lasso method for variable selection in the Cox model". In: *Statistics in Medicine* 16.4, pp. 385–395.

Tibshirani, Ryan J, Taylor, Jonathan, et al. (2011). "The solution path of the generalized lasso". In: *The Annals of Statistics* 39.3, pp. 1335–1371.

Wasserman, Larry and Roeder, Kathryn (2009). "High dimensional variable selection". In: *Annals of Statistics* 37.5A, pp. 2178–2201.

Yuan, Ming and Lin, Yi (2007). "Model selection and estimation in the Gaussian graphical model". In: *Biometrika* 94.1, pp. 19–35.

Zhang, Cun-Hui and Zhang, Stephanie S (2014). "Confidence intervals for low dimensional parameters in high dimensional linear models". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 76.1, pp. 217–242.

Zhang, Xianyang and Cheng, Guang (2017). "Simultaneous inference for high-dimensional linear models". In: *Journal of the American Statistical Association* 112.518, pp. 757–768.

Zou, Hui (2006). "The adaptive lasso and its oracle properties". In: *Journal of the American Statistical Association* 101.476, pp. 1418–1429.

Zou, Hui and Hastie, Trevor (2005). "Regularization and variable selection via the elastic net". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2, pp. 301–320.

Appendix A

Chapter 2 Supplementary Materials

A.1 Exact computation of the LOCO path statistic

We use the fact that the LASSO solution path is piece-wise linear to simplify the computation of the LOCO path statistic. Denote by $\lambda_1, \ldots, \lambda_M$ the set of M knots defines the pieces of a LASSO solution path. For two different LASSO solution paths, the set of knots may be different. Suppose for path $\hat{\beta}$ we have knots $\boldsymbol{\lambda}_1 = \{\lambda_{1,i}, i = 1, 2, \ldots, M_1\}$ and for path $\hat{\beta}^{(-j)}$ we have knots $\boldsymbol{\lambda}_2 = \{\lambda_{2,j}, j = 1, 2, \ldots, M_2\}$.

To calculate $T_j(s,t)$, we need to take the union of $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ so that we have $\boldsymbol{\lambda} = \boldsymbol{\lambda}_1 \cup \boldsymbol{\lambda}_2$. By focusing on small intervals $(\lambda_m, \lambda_{m+1}]$, $m = 1, \ldots, M-1$, we are able to reduce the calculation of $T_j(s,t)$ to the calculation of many simple integrals between two straight lines. Hence the LOCO path statistics $T_j(s,t)$, for $s, t < \infty$, can be expressed as

$$T_j(s,t) = \left(\sum_{k=1}^{p} \left(\sum_{m=1}^{M-1} \int_{\lambda_m}^{\lambda_{m+1}} |\epsilon_{k,m}(\lambda)|^s d\lambda \right)^{\frac{t}{s}} \right)^{\frac{1}{t}}, \tag{A.1}$$

where

$$\epsilon_{k,m}(\lambda) = (\lambda - \lambda_m) \frac{\Delta_{k,m+1} - \Delta_{k,m}}{\lambda_{m+1} - \lambda_m} + \Delta_{k,m}$$

where $\Delta_{k,m} = \hat{\beta}_k^{(-j)}(\lambda_m) - \hat{\beta}_k(\lambda_m)$ and $\Delta_{k,m+1} = \hat{\beta}_k^{(-j)}(\lambda_{m+1}) - \hat{\beta}_k(\lambda_{m+1})$, represent the difference between two straight lines. See Figure A.1 for a depiction of the calculation of $T_j(1,1)$.

From here we just need to solve

$$\int_{\lambda_m}^{\lambda_{m+1}} |\epsilon_{k,m}(\lambda)|^s d\lambda.$$

If $s < \infty$ is even, we have

$$\int_{\lambda_m}^{\lambda_{m+1}} |\epsilon_{k,m}(\lambda)|^s \, d\lambda = \left| \frac{\Delta_{k,m+1}^{s+1} - \Delta_{k,m}^{s+1}}{\Delta_{k,m+1} - \Delta_{k,m}} \right| \frac{\lambda_{m+1} - \lambda_m}{s+1}, \tag{A.2}$$

and if $s < \infty$ is odd, we have

$$\int_{\lambda_m}^{\lambda_{m+1}} |\epsilon_{k,m}(\lambda)|^s \, d\lambda = \begin{cases} \left| \frac{\Delta_{k,m+1}^{s+1} - \Delta_{k,m}^{s+1}}{\Delta_{k,m+1} - \Delta_{k,m}} \right| \frac{\lambda_{m+1} - \lambda_m}{s+1} & \text{if } \Delta_{k,m+1}\Delta_{k,m} > 0 \\ \left| \frac{\Delta_{k,m+1}^{s+1} + \Delta_{k,m}^{s+1}}{\Delta_{k,m+1} - \Delta_{k,m}} \right| \frac{\lambda_{m+1} - \lambda_m}{s+1} & \text{if } \Delta_{k,m+1}\Delta_{k,m} < 0. \end{cases} \tag{A.3}$$

If $s = \infty$ or $t = \infty$, we just compute the maximum of $|\Delta_{k,m}|$ over all $m = 1, 2, \ldots, M$ and $k = 1, 2, \ldots, p$. Hence we can compute $T_j(s,t)$ explicitly.

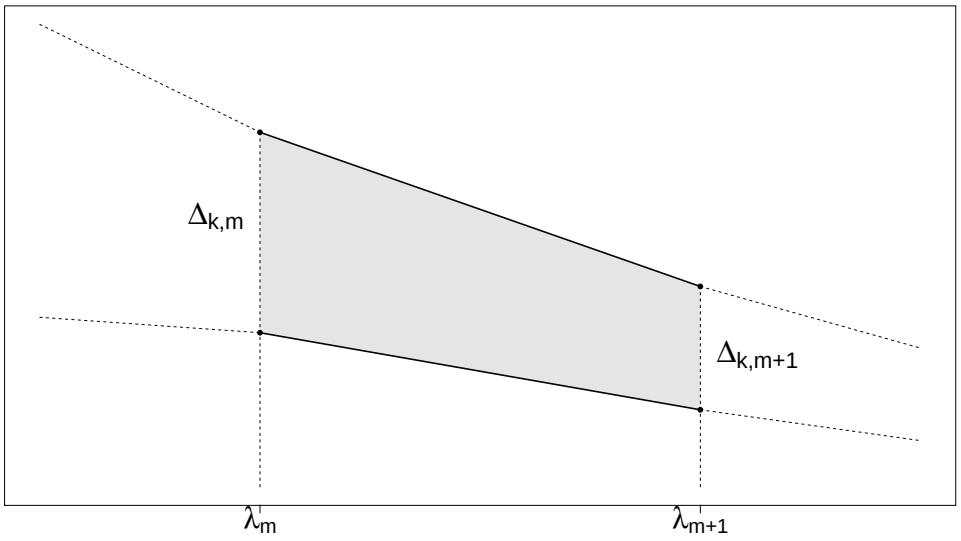

Figure A.1: A detailed look at computing the LOCO path test statistic. Shaded area represents $\int_{\lambda_m}^{\lambda_{m+1}} |\epsilon_{k,m}(\lambda)|^s d\lambda$ with $s = 1$.

A.2 More simulation results

In Table A.1 - A.5 we show all the empirical size of our simulations. In Figure A.2 - A.3 we show some extra empirical size and power curve simulations.

We generate data according to the model

$$Y = \mathbf{X}\beta + \epsilon,$$

and consider three cases with $n = 100$, $p = 12$ and 100. We set $\beta = (\beta_1, \ldots, \beta_p)^T$ such that $\beta_2 = \beta_3 = 1$, $\beta_4 = \cdots = \beta_p = 0$. Other settings are similar to what we described in the Simulation section of our main paper. For $p = 100$, we compare our method to the desparsified LASSO estimator and for $p = 12$, we compare it to the T-test.

For $p = 12$, we achieved higher power compared to the T-test while having the size well controlled. For $p = 100$, $T_1(1, 1)$ achieved comparable power compared to the desparsified LASSO. Although $T_1(\infty, \infty)$ has lower power, but controls size better than the desparsified LASSO.

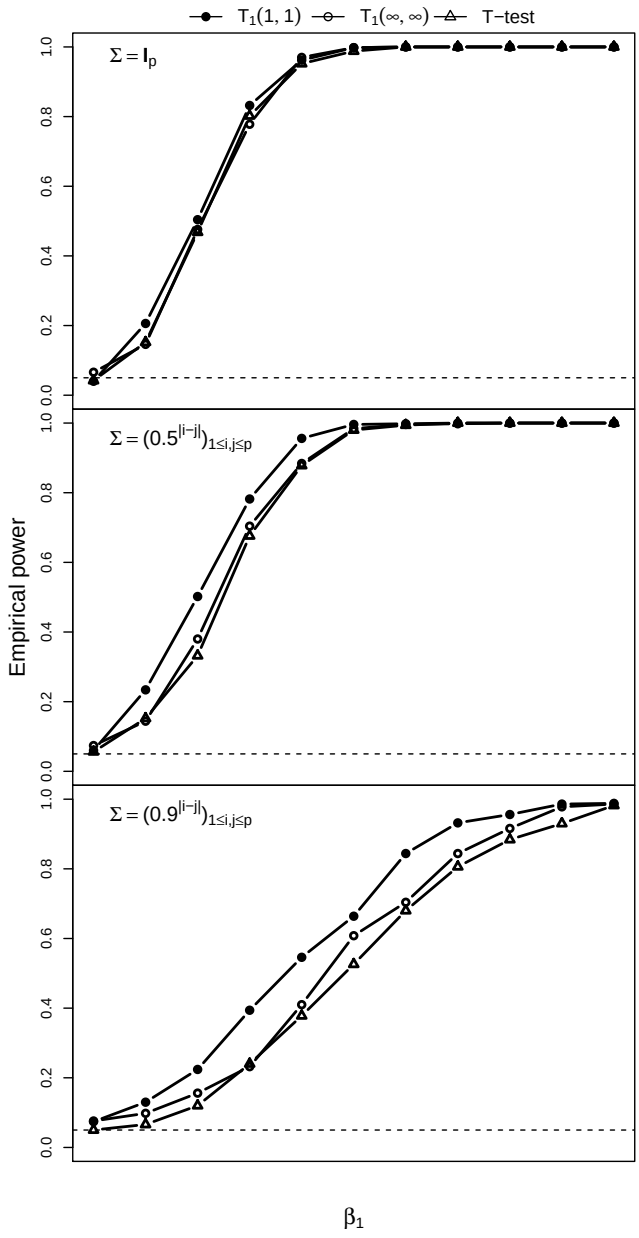

Figure A.2: Empirical power for testing H_0: $\beta_1 = 0$ vs H_1: $\beta_1 \neq 0$ under different correlation design with $n = 100$, $p = 12$.
Upper: $\Sigma = \mathbf{I}_p$, middle: $\Sigma = (0.5^{|i-j|})_{1 \leq i,j \leq p}$, lower: $\Sigma = (0.9^{|i-j|})_{1 \leq i,j \leq p}$

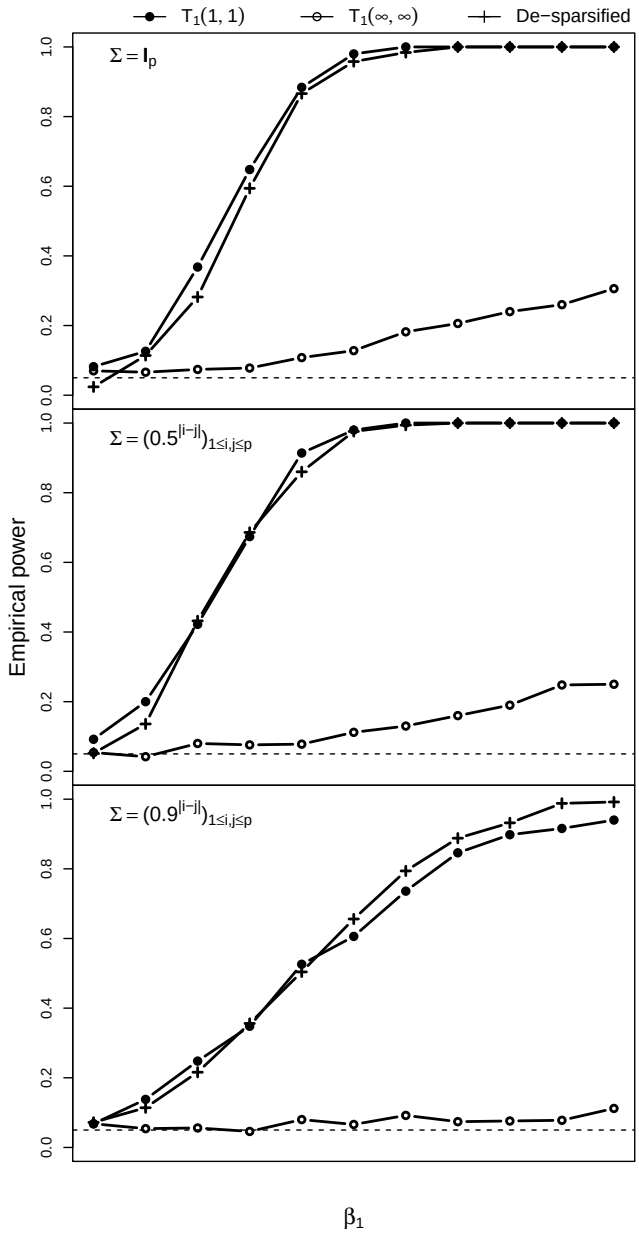

Figure A.3: Empirical power for testing H_0: $\beta_1 = 0$ vs H_1: $\beta_1 \neq 0$ under different correlation design with $n = 100$, $p = 100$.
Upper: $\Sigma = \mathbf{I}_p$, middle: $\Sigma = (0.5^{|i-j|})_{1 \leq i,j \leq p}$, lower: $\Sigma = (0.9^{|i-j|})_{1 \leq i,j \leq p}$

Table A.1: Empirical size of the test under different Σ with $n = 100$, $p = 12$.

Design	Method	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
$\Sigma = \mathbf{I}_p$	$T_1(1,1)$	0.192	0.090	0.04	0.016
	$T_1(2,2)$	0.240	0.118	0.068	0.020
	$T_1(\infty,\infty)$	0.212	0.114	0.066	0.014
	T-test	0.194	0.096	0.042	0.008
$\Sigma = (0.5^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.224	0.110	0.060	0.016
	$T_1(2,2)$	0.226	0.116	0.048	0.020
	$T_1(\infty,\infty)$	0.192	0.108	0.074	0.018
	T-test	0.212	0.098	0.056	0.018
$\Sigma = (0.9^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.200	0.124	0.074	0.024
	$T_1(2,2)$	0.216	0.120	0.082	0.034
	$T_1(\infty,\infty)$	0.236	0.132	0.076	0.018
	T-test	0.186	0.096	0.050	0.012
$\Sigma = (0.5^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.220	0.101	0.060	0.010
	$T_1(2,2)$	0.210	0.106	0.062	0.022
	$T_1(\infty,\infty)$	0.242	0.132	0.058	0.022
	T-test	0.178	0.096	0.050	0.008
$\Sigma = (0.8^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.226	0.128	0.060	0.022
	$T_1(2,2)$	0.204	0.116	0.062	0.016
	$T_1(\infty,\infty)$	0.218	0.096	0.058	0.028
	T-test	0.212	0.102	0.050	0.010

Table A.2: Empirical size of the test under different Σ with $n = 100$, $p = 80$.

Design	Method	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
$\Sigma = \mathbf{I}_p$	$T_1(1,1)$	0.234	0.130	0.066	0.012
	$T_1(2,2)$	0.230	0.110	0.058	0.016
	$T_1(\infty,\infty)$	0.214	0.086	0.040	0.010
	T-test	0.196	0.104	0.038	0.020
$\Sigma = (0.5^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.228	0.102	0.060	0.018
	$T_1(2,2)$	0.240	0.134	0.072	0.016
	$T_1(\infty,\infty)$	0.214	0.112	0.070	0.020
	T-test	0.208	0.094	0.052	0.012
$\Sigma = (0.9^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.210	0.106	0.058	0.018
	$T_1(2,2)$	0.210	0.116	0.058	0.024
	$T_1(\infty,\infty)$	0.192	0.106	0.068	0.018
	T-test	0.206	0.092	0.052	0.012
$\Sigma = (0.5^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.218	0.110	0.064	0.018
	$T_1(2,2)$	0.206	0.106	0.046	0.016
	$T_1(\infty,\infty)$	0.260	0.150	0.072	0.018
	T-test	0.174	0.096	0.054	0.018
$\Sigma = (0.8^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.224	0.112	0.042	0.012
	$T_1(2,2)$	0.228	0.124	0.068	0.016
	$T_1(\infty,\infty)$	0.220	0.114	0.058	0.014
	T-test	0.212	0.112	0.048	0.006

Table A.3: Multiple testing empirical size under different Σ with $n = 100$, $p = 80$.

Design	Method	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
$\Sigma = \mathbf{I}_p$	$T_1(1,1)$	0.270	0.148	0.092	0.024
	$T_1(2,2)$	0.236	0.13	0.068	0.012
	$T_1(\infty,\infty)$	0.232	0.142	0.088	0.018
	F-test	0.204	0.084	0.036	0.004
$\Sigma = (0.5^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.230	0.122	0.064	0.014
	$T_1(2,2)$	0.274	0.140	0.068	0.024
	$T_1(\infty,\infty)$	0.224	0.110	0.066	0.018
	F-test	0.168	0.082	0.036	0.012
$\Sigma = (0.9^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.280	0.148	0.082	0.032
	$T_1(2,2)$	0.240	0.148	0.090	0.032
	$T_1(\infty,\infty)$	0.240	0.116	0.064	0.016
	F-test	0.200	0.100	0.038	0.008
$\Sigma = (0.5^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.242	0.118	0.052	0.010
	$T_1(2,2)$	0.244	0.122	0.058	0.012
	$T_1(\infty,\infty)$	0.208	0.110	0.056	0.016
	F-test	0.204	0.094	0.048	0.006
$\Sigma = (0.8^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.230	0.104	0.062	0.020
	$T_1(2,2)$	0.216	0.102	0.054	0.018
	$T_1(\infty,\infty)$	0.272	0.156	0.088	0.026
	F-test	0.202	0.088	0.034	0.014

Table A.4: Empirical size of the test under different Σ with $n = 100$, $p = 100$.

Design	Method	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
$\Sigma = \mathbf{I}_p$	$T_1(1,1)$	0.244	0.134	0.082	0.018
	$T_1(2,2)$	0.192	0.102	0.046	0.012
	$T_1(\infty,\infty)$	0.236	0.122	0.070	0.024
	Desparsified	0.194	0.102	0.046	0.010
$\Sigma = (0.5^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.252	0.118	0.052	0.022
	$T_1(2,2)$	0.240	0.142	0.076	0.018
	$T_1(\infty,\infty)$	0.212	0.104	0.054	0.012
	Desparsified	0.23	0.118	0.066	0.012
$\Sigma = (0.9^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.234	0.120	0.068	0.026
	$T_1(2,2)$	0.214	0.124	0.066	0.014
	$T_1(\infty,\infty)$	0.220	0.120	0.068	0.014
	Desparsified	0.234	0.120	0.060	0.016
$\Sigma = (0.5^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.256	0.138	0.072	0.014
	$T_1(2,2)$	0.226	0.132	0.050	0.010
	$T_1(\infty,\infty)$	0.204	0.108	0.058	0.024
	Desparsified	0.204	0.096	0.062	0.010
$\Sigma = (0.8^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.242	0.138	0.062	0.010
	$T_1(2,2)$	0.224	0.124	0.064	0.024
	$T_1(\infty,\infty)$	0.214	0.116	0.050	0.010
	Desparsified	0.174	0.090	0.052	0.018

Table A.5: Multiple testing empirical size under different Σ with $n = 100$, $p = 1000$.

Design	Method	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
$\Sigma = \mathbf{I}_p$	$T_1(1,1)$	0.208	0.130	0.070	0.018
	$T_1(2,2)$	0.242	0.124	0.066	0.018
	$T_1(\infty,\infty)$	0.268	0.126	0.068	0.024
$\Sigma = (0.5^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.230	0.128	0.078	0.018
	$T_1(2,2)$	0.184	0.100	0.056	0.018
	$T_1(\infty,\infty)$	0.186	0.100	0.050	0.014
$\Sigma = (0.9^{\lvert i-j\rvert})_{1\le i,j\le p}$	$T_1(1,1)$	0.210	0.134	0.076	0.020
	$T_1(2,2)$	0.230	0.146	0.090	0.024
	$T_1(\infty,\infty)$	0.212	0.102	0.046	0.000
$\Sigma = (0.5^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.184	0.102	0.058	0.012
	$T_1(2,2)$	0.206	0.084	0.044	0.008
	$T_1(\infty,\infty)$	0.176	0.096	0.058	0.016
$\Sigma = (0.8^{\mathbf{1}(i\neq j)})_{1\le i,j\le p}$	$T_1(1,1)$	0.214	0.142	0.098	0.042
	$T_1(2,2)$	0.188	0.086	0.046	0.016
	$T_1(\infty,\infty)$	0.148	0.054	0.026	0.004