On Variable Selection Diagnostics for High-Dimensional Regression Models

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Yanjia Yu

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Yuhong Yang, Adviser

January 2023

Acknowledgements

First and foremost, I would like to thank my advisor, Prof. Yuhong Yang, for his continuous support, advice and guidance over the years. I appreciate that he never gives up on me and always thoughtfully encourages and supports me through ups and downs. He models the way for me in all aspects of research and life with his wisdom and virtue. It has been my extreme honor to have him as my advisor.

I am grateful to the rest of my dissertation committee members Prof. Hui Zou, Prof. Jie Ding and Prof. Wei Pan. Their help and strong support throughout this process have been invaluable to me. I would also like to thank my preliminary exam committee member Prof. Gongjun Xu for his help and support.

Meanwhile, I would like to give my sincere thanks to the excellent faculty in the School of Statistics at the University of Minnesota and in the Department of Statistics and Finance at the University of Science and Technology of China for bringing me into the fascinating world of Statistics, and building a solid statistical foundation in me through their teaching and guidance. Besides, I want to thank the helpful staff, lovely peers and friends, and everyone that I acquainted during my Ph.D. journey. They make this challenging journey memorable and colorful.

Special thanks go to my parents, Xiucai Yu and Yichun Li, for always being so supportive and unconditionally loving.

Last but not least, this dissertation would not have been completed without my loving and supportive husband, Yuwen Gu.

Dedication

To my parents, husband and daughter.

Abstract

Because model selection is ubiquitous in data analysis, the reproducibility of statistical results requires that we be able to evaluate the reliability of the employed model selection method, regardless of the model's apparent good properties. Instability measures have been proposed for evaluating model selection uncertainty. However, low instability does not necessarily indicate that the selected model is trustworthy, because low instability can also arise when a method tends to select an overly parsimonious model. F- and G-measures have become increasingly popular for assessing variable selection performance in theoretical studies and simulation results. However, they are not computable in practice.

In this dissertation work, we propose an estimation method for F- and G-measures and prove their desirable properties of uniform consistency. This gives the data analyst a valuable tool to compare different variable selection methods based on the data at hand. Extensive simulations are conducted to show the very good finite-sample performance of our approach. We apply our methods to several microarray gene expression data sets, with intriguing results.

We also extend the work of Nan and Yang (2014) on variable selection deviation (VSD) measures and Yu et al. (2022) on F- and G-measures to a broader class of models in the exponential dispersion family, including, for example, the Poisson and compound Poisson-gamma models. In particular, we consider the Tweedie family of models that possesses a power mean-variance relationship, for its wide spectrum of applications in fields such as insurance, ecology, political science and health and biomedical studies. We propose methods based on information criteria and adaptive regression by mixing (ARM) to compute

the weights of the candidate models that are adaptive to their predictive performance for the Poisson and Tweedie regression models. Our extensive empirical studies show that the proposed diagnostic measures (including VSD, F- and G-measures) are reasonable metrics of variable selection performance and the weighting methods work very well in recovering the true variable selection deviations.

An R package named **PAVI** is developed to calculate the various variable selection diagnostic measures for all members of the generalized linear models. Three most widely used weighting methods based on AIC, BIC and ARM are supported. Parallel computation mechanism and procedures for dealing with convergence issues of the numerical optimization algorithm in R's `glm` function are implemented to smoothly carry out the weighting procedures. Extensive numerical experiments conducted using this package show that it is stable and delivers expected results.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Due to advancement of data acquisition technologies, massive and complex data are continuously collected everyday from across scientific, financial, and social sectors. A particular challenge for the analysis of these data is to deal with the curse of dimensionality caused by the numerous features therein. To that end, a variety of statistical methods have been proposed, including nowadays the widely used regularization techniques that perform feature or variable selection (see, e.g., the excellent survey by Ding et al., 2018). To name a few, popular regularizers such as the lasso (Tibshirani, 1996), SCAD (Fan and Li, 2001), adaptive lasso (Zou, 2006a), and MCP (Zhang, 2010) are now in the toolbox of every data analyst. The regularized methods are particularly good at solving the so-called high-dimensional data problems, where the dimension of the data can well exceed the number of observations. They not only provide parameter estimates but also perform variable selection, which is a very important factor a data analyst considers to adopt a method when model interpretability is a concern.

Albeit their common use in practice, it is also well recognized that these regularized methods often encounter instability issues. For example, removing some observations

or adding small noises to the data may result in dramatically different selected sets of variables. This uncertainty in variable selection may lead to inconsistent scientific findings or contradictory conclusions.

Instability measures to evaluate variable selection uncertainty have been commonly used in practice to quantify the sensitivity of a variable selection method to small changes of the data due to subsampling, resampling, or perturbations. However, the instability measures may not fully capture the uncertainties in variable selection because an overly parsimonious model (such as the intercept-only model) has very low instability measure but may not be able to fit the data well enough. To that end, Nan and Yang (2014) propose the variable selection deviation measures to fully capture both the false positives and false negatives of a variable selection method. Since the true model is typically unknown in practice, the false positives and negatives are not obtainable in most cases. Nan and Yang (2014) pioneer the adaptive weighting approach to estimating those based on a carefully chosen set of candidate models, with more weights assigned to better-performing models. The proposed weighting methods provide a practical way of estimating the variable selection deviations of a variable selection method in the least squares setting.

In this dissertation, we study more integrative metrics of variable selection uncertainty for better performance assessment of high-dimensional variable identification. In particular, we consider the F- and G-measures that are good summary metrics for integrating both false positives and false negatives. We also generalize the methodology of Nan and Yang (2014) to the generalized linear models, with a particular focus on logistic, Poisson and Tweedie models for their wide applications.

1.2 Dissertation Outline

This dissertation is mainly composed of three chapters (Chapters 2 – 4), which discuss approaches to variable selection performance assessment in the high-dimensional generalized linear models. The main goal of this dissertation is to provide practical methodology and software implementation to calculate deviation measures of various variable selection methods in real data.

In Chapter 2, we propose a method for the performance assessment of variable identification (PAVI), in which we estimate the F- and G-measures based on a combination of multiple candidate models under a proper weighting scheme. Our proposal works for both regression and classification, and applies to both synthetic and real data. Under sensible conditions, we show that our estimates are uniformly consistent in estimating the true F- and G-measures for any set of models to be checked. Two weighting schemes are considered in this chapter: adaptive regression by mixing (ARM, Yang, 2001), and weighting via information criteria (see, e.g., Nan and Yang, 2014).

In Chapter 3, we extend the work of Nan and Yang (2014) on variable selection deviation measures (VSD, VSD^+, VSD^-) and Yu et al. (2022) (and Chapter 2 of this dissertation) on F- and G-measures to a broader class of models in the exponential dispersion family (Jørgensen, 1987; McCullagh and Nelder, 1989; Jørgensen, 1997), including, for example, the Poisson and compound Poisson-gamma models. In particular, we consider the Tweedie family of models that possesses a power mean-variance relationship, for its wide spectrum of applications in many applied fields. Our numerical studies demonstrate that the proposed weighting schemes work well for the high-dimensional generalized linear models considered in this chapter.

In Chapter 4, we present the R package **PAVI** for computing the variable selection

diagnostic measures, such as VSD, VSD^+, VSD^-, precision, recall, F- and G-measures, that are mentioned in previous chapters. We also describe various helper functions for carrying out tasks such as candidate model fitting and weight assignment for the candidate models, using either ARM or an information criterion such as AIC or BIC. Some simulation examples are given to demonstrate the use of the main functions.

Finally, we conclude the dissertation in Chapter 5, with a brief discussion of potential future work.

Chapter 2

Performance Assessment of High-dimensional Variable Identification

2.1 Introduction

Variable selection is of interest in many fields, including bioinformatics, genomics, finance, and economics. In bioinformatics, for example, microarray gene expression data are collected to identify cancer-related biomarkers in order to differentiate affected patients from healthy individuals based on their gene expression profile. The number of variables, p, in typical microarray gene expression data is of 10^{3-5} magnitude, while the number of subjects, n, is of 10^{1-3} magnitude. For problems in which $p \gg n$, the penalized likelihood estimation provides a class of methods for selecting the variables (see, e.g., Fan and Lv, 2010). However, it is well recognized in the literature that model selection methods, including the penalization methods for high-dimensional data, often encounter instability issues (Chatfield, 1995; Draper, 1995; Breiman, 1996a,b; Buckland et al., 1997; Yuan and Yang, 2005; Lim and Yu, 2016). For example, removing a few observations or adding small perturbations to the data may result in dramatically different sets of variables being

selected (Meinshausen and Bühlmann, 2006; Nan and Yang, 2014; Lim and Yu, 2016). This uncertainty in variable selection, as is well known, may have severe practical consequences. On a larger scale, reproducibility is a major problem in the science community (McNutt, 2014; Stodden, 2015).

Variable selection uncertainty is mainly evaluated using instability measures, which test how sensitive a variable selection method is to small changes in the data because of subsampling (Chen et al., 2007), resampling (Breiman, 1996b; Buckland et al., 1997), or perturbations (Breiman, 1996b). However, a low instability measure does not necessarily indicate that a variable selection result is reliable, because low instability can also arise when a method tends to select an overly parsimonious model (e.g., the intercept-only model, in the extreme case).

There is therefore a great need for measures that can fully evaluate the uncertainty of variable selection beyond instability. In variable selection, researchers focus on two types of errors: including unnecessary variables, and excluding important variables. F- and G-measures are popular in the field of information retrieval (Billsus and Pazzani, 1998) for assessing overall variable selection performance (see, e.g., Lim, 2011; Lim and Yu, 2016). Specifically, the F-measure is the harmonic mean of *precision* and *recall*, where precision (or positive predictive value) is defined as the fraction of the selected variables that are true variables, and recall (also known as sensitivity) is defined as the fraction of the true variables that are selected. The G-measure is the geometric mean of precision and recall. By combining precision and recall into one measure, one can evaluate the overall accuracy of a given variable selection method. Clearly, a higher F (or G) value indicates better selection performance, in an overall sense. However, existing approaches calculate the F- (or G-) measure of a given selection method for simulated data only (where the true model

is known), and do not work for real data.

In this work, we propose a method for the **p**erformance **a**ssessment of (high-dimensional) **v**ariable **i**dentification (PAVI), in which we estimate the F- or G-measure based on a combination of multiple candidate models under a proper weighting scheme. Our proposal works for both regression and classification, and applies to both synthetic and real data. Under sensible conditions, we show that our estimates are uniformly consistent in estimating the true F- and G-measures for any set of models to be checked. The candidate models can be very flexible. For example, they can be obtained by penalization using the Lasso (Tibshirani, 1996), smoothly clipped absolute deviations (SCAD) penalty (Fan and Li, 2001), adaptive Lasso (Zou, 2006a), minimax concave penalty (MCP) (Zhang, 2010) or other variable selection techniques. Two weighting schemes are considered in this work: adaptive regression by mixing (Yang, 2001), and weighting via information criteria (see, e.g., Nan and Yang, 2014). In the simulation section, we show the reliable estimation performance of our method for both classification and regression. We further demonstrate our methods by analyzing several microarray gene expression data from real applications. The results of the real data analysis suggest that the PAVI method is very useful for evaluating the variable selection performance of high-dimensional linear-based models. It provides useful information on the reliability and reproducibility of a given model when the true model is unknown. For example, one may justifiably doubt the reproducibility of a model that has very small estimated F- and G- values.

The remainder of the chapter is organized as follows. In Section 2.2, we define the F- and G-measures and introduce our estimation methods. Section 2.3 provides the theoretical justification for the PAVI estimators of the F- and G-measures. Section 2.4 shows how to implement the PAVI method for both regression and classification, including how to obtain

the candidate models and assign weights. Simulation results are presented in Section 2.5. We demonstrate our methods by analyzing three well-studied gene expression data sets in Section 2.6. Section 2.7 concludes the paper. All technical proofs are relegated to Section 2.8 along with additional numerical results.

2.2 Methodology

Let us consider the generalized linear model framework. Denote $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\intercal$ as the $n \times p$ design matrix with $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\intercal$, for $i = 1, \ldots, n$. Let $\mathbf{y} = (y_1, \ldots, y_n)^\intercal$ be the n-dimensional response vector. For a regression with a continuous response, we consider the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon}$ is the vector of n independent errors, and $\boldsymbol{\beta}^* = (\beta_1^*, \ldots, \beta_p^*)^\intercal$ is a p-dimensional coefficient vector of the true underlying model that generates the data. For classification, we consider the binary logistic regression model, for ease of presentation. Let $Y \in \{0, 1\}$ be a binary response variable, and $X \in \mathbb{R}^p$ be a p-dimensional predictor vector. We assume that Y follows the Bernoulli distribution given $X = \mathbf{x}$, with conditional probability

$$\Pr(Y = 1 | X = \mathbf{x}) = 1 - \Pr(Y = 0 | X = \mathbf{x}) = \frac{e^{\mathbf{x}^\intercal \boldsymbol{\beta}^*}}{1 + e^{\mathbf{x}^\intercal \boldsymbol{\beta}^*}}. \tag{2.1}$$

Let $\mathcal{A}^* = \text{supp}(\boldsymbol{\beta}^*) \equiv \{j : \beta_j^* \neq 0\}$ be the index set of the variables in the true model with size $|\mathcal{A}^*|$, where $|\cdot|$ denotes the cardinality of a set. For both regression and classification, we assume that the true model is sparse. In other words, most coefficients in $\boldsymbol{\beta}^*$ are exactly zero, such that $|\mathcal{A}^*|$ is small.

Let $\mathcal{A}^0 = \{j : \beta_j^0 \neq 0\}$ be an index set of all nonzero coefficients from any given variable selection result $\boldsymbol{\beta}^0$. One can use F- and G- measures to evaluate the performance of $\mathcal{A}^0$. F- and G-measures take values between zero and one, where a higher value indicates better performance of the variable selection method. The definitions of F- and G- measures are based on *precision* and *recall.* The precision pr for $\mathcal{A}^0$ is the fraction of true variables in the given model $\mathcal{A}^0$; that is, $pr(\mathcal{A}^0) \equiv pr(\mathcal{A}^0; \mathcal{A}^*) = |\mathcal{A}^0 \cap \mathcal{A}^*|/|\mathcal{A}^0|$. The recall re for $\mathcal{A}^0$ is the fraction of variables in the true model $\mathcal{A}^*$ that are selected; that is, $re(\mathcal{A}^0) \equiv re(\mathcal{A}^0; \mathcal{A}^*) = |\mathcal{A}^0 \cap \mathcal{A}^*|/|\mathcal{A}^*|$. The F-measure for a given model $\mathcal{A}^0$ is defined as the harmonic mean of the precision and recall, and the G-measure is defined as the geometric mean of the two. Specifically,

$$F(\mathcal{A}^0) = F(\mathcal{A}^0; \mathcal{A}^*) \equiv \frac{2 \times pr(\mathcal{A}^0) \times re(\mathcal{A}^0)}{pr(\mathcal{A}^0) + re(\mathcal{A}^0)} = \frac{2|\mathcal{A}^0 \cap \mathcal{A}^*|}{|\mathcal{A}^0| + |\mathcal{A}^*|},$$

and

$$G(\mathcal{A}^0) = G(\mathcal{A}^0; \mathcal{A}^*) \equiv \sqrt{pr(\mathcal{A}^0) \times re(\mathcal{A}^0)} = \frac{|\mathcal{A}^0 \cap \mathcal{A}^*|}{\sqrt{|\mathcal{A}^0| \cdot |\mathcal{A}^*|}}.$$

In a penalized regression, it is well known that when the penalty level is increased, fewer active variables are selected. Therefore, false positives are less likely to happen, whereas false negatives become more likely. By taking the harmonic (or geometric) mean of the precision and recall, the F-measure (or G-measure) integrates both false-positive and false-negative aspects into a single characterization. Given $\mathcal{A}^0$, a high F- or G-measure indicates that the false-positive and false-negative rates are both low. For example, if $\mathcal{A}^* = (1, 1, 1, 0, 0, 0, 0)$ and $\mathcal{A}_1^0 = (1, 1, 1, 0, 0, 0, 1)$, then $pr(\mathcal{A}_1^0) = 3/4$, $re(\mathcal{A}_1^0) = 1$, $F(\mathcal{A}_1^0) = 6/7$, and $G(\mathcal{A}_1^0) = \sqrt{3}/2$. For the same $\mathcal{A}^*$, if we consider a worse case $\mathcal{A}_2^0 = (1, 1, 0, 0, 0, 0, 1)$, then $pr(\mathcal{A}_2^0) = 2/3$, $re(\mathcal{A}_2^0) = 2/3$, $F(\mathcal{A}_2^0) = 2/3$, and $G(\mathcal{A}_2^0) = 2/3$. The F- and G-measures are smaller than those in the first case owing to the existence of

both under-selection and over-selection. In general, F- and G-measures are conservative, in the sense that both are more sensitive to under-selection than they are to over-selection. Specifically, suppose $|\mathcal{A}^*| = m$. If $\mathcal{A}_3^0$ over-selects one variable, then $|\mathcal{A}_3^0| = m+1$, $F(\mathcal{A}_3^0) = 2m/(2m+1)$, and $G(\mathcal{A}_3^0) = \sqrt{m/(m+1)}$. However, if $\mathcal{A}_4^0$ under-selects one variable, then $|\mathcal{A}_4^0| = m-1$, $F(\mathcal{A}_4^0) = (2m-2)/(2m-1)$, and $G(\mathcal{A}_4^0) = \sqrt{(m-1)/m}$. One can easily see that $F(\mathcal{A}_3^0) > F(\mathcal{A}_4^0)$ and $G(\mathcal{A}_3^0) > G(\mathcal{A}_4^0)$.

In real applications, the true model $\mathcal{A}^*$ is usually unknown, and thus we cannot directly know $F(\mathcal{A}^0)$ and $G(\mathcal{A}^0)$ for any given model $\mathcal{A}^0$. However, by borrowing information from a group of given models, we can estimate $F(\mathcal{A}^0)$ and $G(\mathcal{A}^0)$ from the data. Suppose that we have a set of candidate models $\mathbb{S} = \{\mathcal{A}^1, \ldots, \mathcal{A}^K\}$, which can be obtained from a preliminary analysis. When the model size p is small, we can use a full collection of all-subset models $\mathbb{S} = \mathbb{C}$, where

$$\mathbb{C} = \{\emptyset, \{1\}, \ldots, \{p\}, \{1,2\}, \{1,3\}, \ldots, \{1, \ldots, p\}\},$$

where $1, \ldots, p$ represents the indices of the p variables. If p is too large, we can choose $\mathbb{S}$ as a group of models obtained from penalized methods, such as the Lasso, adaptive Lasso, SCAD, and MCP. Define $\mathbf{w} = \{w_1, \ldots, w_K\}$ as the corresponding data-driven weights for $\mathbb{S} = \{\mathcal{A}^1, \ldots, \mathcal{A}^K\}$, where $w_k \geq 0$, for $k = 1, \ldots, K$, and $\sum_{k=1}^K w_k = 1$. In Section 4.1,we further describe how we acquire $\mathbb{S}$ and $\mathbf{w}$. For now, we assume these are already properly acquired. For each $\mathcal{A}^k$, we define the estimated precision and recall for $\mathcal{A}^0$ (relative to $\mathcal{A}^k$) as $pr(\mathcal{A}^0; \mathcal{A}^k) = |\mathcal{A}^0 \cap \mathcal{A}^k|/|\mathcal{A}^0|$ and $re(\mathcal{A}^0; \mathcal{A}^k) = |\mathcal{A}^0 \cap \mathcal{A}^k|/|\mathcal{A}^k|$, and propose the following $\widehat{F}(\mathcal{A}^0)$ using PAVI to estimate $F(\mathcal{A}^0)$:

$$\widehat{F}(\mathcal{A}^0) = \sum_{k=1}^K w_k F(\mathcal{A}^0; \mathcal{A}^k) = 2\sum_{k=1}^K w_k \frac{|\mathcal{A}^0 \cap \mathcal{A}^k|}{|\mathcal{A}^0| + |\mathcal{A}^k|}. \tag{2.2}$$

Similarly, we propose $\widehat{G}(\mathcal{A}^0)$ using PAVI to estimate $G(\mathcal{A}^0)$:

$$\widehat{G}(\mathcal{A}^0) = \sum_{k=1}^{K} w_k G(\mathcal{A}^0; \mathcal{A}^k) = 2 \sum_{k=1}^{K} w_k \frac{|\mathcal{A}^0 \cap \mathcal{A}^k|}{\sqrt{|\mathcal{A}^0| \cdot |\mathcal{A}^k|}}. \tag{2.3}$$

We define the (sample) standard deviation of $\widehat{F}(\mathcal{A}^0)$ as

$$\mathrm{sd}\big(\widehat{F}(\mathcal{A}^0)\big) = \sqrt{\sum_{k=1}^{K} w_k \big(F(\mathcal{A}^0; \mathcal{A}^k) - \widehat{F}(\mathcal{A}^0)\big)^2}. \tag{2.4}$$

Similarly, the (sample) standard deviation of $\widehat{G}(\mathcal{A}^0)$ is

$$\mathrm{sd}\big(\widehat{G}(\mathcal{A}^0)\big) = \sqrt{\sum_{k=1}^{K} w_k \big(G(\mathcal{A}^0; \mathcal{A}^k) - \widehat{G}(\mathcal{A}^0)\big)^2}. \tag{2.5}$$

In (2.2) and (2.3), $\widehat{F}(\mathcal{A}^0)$ and $\widehat{G}(\mathcal{A}^0)$ are estimated using the candidate models $\mathcal{A}^k \in \mathbb{S}$ and weights $w_k \in \mathbf{w}$, for $k = 1, \ldots, K$. Intuitively, if higher weights w_k are assigned to those $\mathcal{A}^k$ that are closer to the true model $\mathcal{A}^*$, then $\widehat{F}(\mathcal{A}^0)$ and $\widehat{G}(\mathcal{A}^0)$ should better approximate the true values of $F(\mathcal{A}^0)$ and $G(\mathcal{A}^0)$, respectively. In Section 4.2, we discuss the methods for computing the weights $\mathbf{w}$ from the data.

2.3 Theory

In this section, we show that the proposed estimators $\widehat{F}$ and $\widehat{G}$ are uniformly consistent for the true F and G, respectively, over the set of all models to be checked. The theory relies on the *weak consistency* (see Definition 2.1 and Nan and Yang, 2014) of the data-dependent model weights $\mathbf{w} = \{w_1, \ldots, w_K\}$, and the *weak inclusion property*, which indicates whether a model screening process is applied to reduce the model list (Definition 2.2).

Definition 2.1 (Weak consistency) The weighting vector $\mathbf{w} = (w_1, \dots, w_K)^\intercal$ is weakly consistent if

$$\frac{\sum_{k=1}^{K} w_k \cdot |\mathcal{A}^k \nabla \mathcal{A}^*|}{|\mathcal{A}^*|} \xrightarrow{p} 0 \text{ as } n \to \infty,$$

where ∇ denotes the symmetric difference between two sets. □

Remark The definition basically says that $\mathbf{w}$ is sufficiently concentrated around the true model $\mathcal{A}^*$, such that the weighted deviation $|\mathcal{A}^k \nabla \mathcal{A}^*|$ eventually diminishes relative to the size of the true model. When the true model is allowed to increase in dimension as n increases, including the denominator $|\mathcal{A}^*|$ in the definition makes the condition more likely to be satisfied. □

The following theorem shows that under the weak consistency condition, the estimators $\widehat{F}$ and $\widehat{G}$ are uniformly consistent (the proof is provided in Section 2.8).

Theorem 2.1 (Uniform consistency of $\widehat{F}$ and $\widehat{G}$)
Suppose the model weighting $\mathbf{w}$ is weakly consistent. Then, $\widehat{F}$ and $\widehat{G}$ based on PAVI are uniformly consistent, in the sense that

$$\sup_{\mathcal{A}^0 \in \mathbb{C}} |\widehat{F}(\mathcal{A}^0) - F(\mathcal{A}^0)| \xrightarrow{p} 0 \quad \text{as } n \to \infty;$$

$$\sup_{\mathcal{A}^0 \in \mathbb{C}} |\widehat{G}(\mathcal{A}^0) - G(\mathcal{A}^0)| \xrightarrow{p} 0 \quad \text{as } n \to \infty.$$

□

From this theorem, we see that if the model weighting focuses mostly on models that are sensibly close to the true model, then our estimated $\widehat{F}$ and $\widehat{G}$ will be close to their respective true values. Clearly, we also have $E|\widehat{F}(\mathcal{A}^0) - F(\mathcal{A}^0)| \to 0$ and $E|\widehat{G}(\mathcal{A}^0) - G(\mathcal{A}^0)| \to 0$, uniformly.

Theorem 2.2 (Uniform convergence of $\mathrm{sd}(\widehat{F})$ and $\mathrm{sd}(\widehat{G})$)
Suppose the model weighting $\mathbf{w}$ is weakly consistent. Then $\mathrm{sd}(\widehat{F})$ and $\mathrm{sd}(\widehat{G})$ based on PAVI converge to zero in probability uniformly, in the sense that

$$\sup_{\mathcal{A}^0 \in \mathbb{C}} |\mathrm{sd}(\widehat{F}(\mathcal{A}^0))| \xrightarrow{p} 0 \quad \text{as } n \to \infty;$$

$$\sup_{\mathcal{A}^0 \in \mathbb{C}} |\mathrm{sd}(\widehat{G}(\mathcal{A}^0))| \xrightarrow{p} 0 \quad \text{as } n \to \infty.$$

□

From this theorem, we see that if the model weighting is sensible, then $\mathrm{sd}(\widehat{F})$ and $\mathrm{sd}(\widehat{G})$ will be close to zero. The results also support the reliability of our PAVI method.

Theorems 2.1 and 2.2 rely on the weak consistency of $\mathbf{w}$. Clearly, when the candidate models in $\mathbb{S}$ are all poor, weak consistency may not be plausible. One can choose all-subset models $\mathbb{C}$ as $\mathbb{S}$ when p is small, because it always contains $\mathcal{A}^*$. However, in the high-dimensional case, it would be computationally infeasible to use $\mathbb{C}$, and a model screening process may be applied (e.g., considering solution paths of model selection methods).

Definition 2.2 (Weak inclusion property) A set of candidate models $\mathbb{S}$ obtained by a model screening process is called weakly inclusive with respect to $\mathbf{w}$ on $\mathbb{C}$ if $\sum_{k \in \mathbb{S}} w_k$ is bounded away from zero in probability. □

Theorem 2.3
Under the assumption that the weighting vector $\mathbf{w}$ on the all-subset models $\mathbb{C}$ is weakly consistent, as long as $\mathbb{S}$ is weakly inclusive, the conclusions of Theorems 2.1 and 2.2 still hold. □

Remarks on this result are given in Section 2.8.

2.4 Implementation

2.4.1 Candidate models

We discuss how to choose the candidate models for computing $\widehat{F}$ and $\widehat{G}$. To obtain the candidate models, we can use a complete collection of all-subset models; that is, we can choose $\mathbb{S} = \mathbb{C}$. However, in the high-dimensional case, where $p \gg n$, it is almost impossible to use all subsets owing to the high computational cost.

Here, we show how to choose the candidate models for linear and logistic regression models in the high-dimensional setting. Similar procedures apply to other likelihood-based models. Given n independent observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ for the pair (X, Y), we can fit the linear or logistic regression model by minimizing the penalized negative log-likelihood

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} -\ell(\boldsymbol{\beta}) + \sum_{j=1}^{p} p_\lambda(\beta_j), \tag{2.6}$$

where $-\ell(\boldsymbol{\beta}) = (2n)^{-1}\sum_{i=1}^n (y_i - \mathbf{x}_i^\intercal\boldsymbol{\beta})^2$ for the linear regression, and

$$-\ell(\boldsymbol{\beta}) = n^{-1}\sum_{i=1}^{n}\{-y_i \log \pi_i - (1-y_i)\log(1-\pi_i)\}$$

for the logistic regression, where $\pi_i = \Pr(Y_i = 1|X_i = \mathbf{x}_i)$ is the probability in (2.1) for observation i. The nonnegative penalty function $p_\lambda(\cdot)$, with $\lambda \in [0, \infty)$, can be the Lasso (Tibshirani, 1996), SCAD (Fan and Li, 2001), MCP (Zhang, 2010), or some other regularizer.

We compute the models $\mathbb{S} = \{\mathcal{A}^{\lambda_1}, \ldots, \mathcal{A}^{\lambda_L}\}$ for the Lasso, SCAD, and MCP on the solution paths $\{\widehat{\boldsymbol{\beta}}^{\lambda_1}, \ldots, \widehat{\boldsymbol{\beta}}^{\lambda_L}\}$ for decreasing sequences of tuning parameters $\{\lambda_1, \ldots, \lambda_L\}$. These models are then combined as a set of candidate models $\mathbb{S} = \{\mathbb{S}_{\text{Lasso}}, \mathbb{S}_{\text{SCAD}}, \mathbb{S}_{\text{MCP}}\}$. One can efficiently compute all solution paths of the Lasso using **glmnet** (Friedman et al., 2010), and those of the SCAD and MCP using **ncvreg** (Breheny and Huang, 2011).

2.4.2 Weighting methods

There are several different methods in the literature for determining the weights $\mathbf{w} = \{w_1, \ldots, w_K\}$. For example, Buckland et al. (1997) and Leung and Barron (2006) proposed information-criterion-based methods for weighting, such as those using the AIC (Akaike, 1973) and BIC (Schwarz, 1978a). Hoeting et al. (1999) proposed the Bayesian model averaging (BMA) method for weighting, and Yang (2001) studied a weighting strategy called the adaptive regression by mixing (ARM), which computes the weights using data splitting and cross-assessment. It is proven in Yang (2001) that the weighting by the ARM delivers the best rate of convergence for regression estimation. Yang (2000) also extend the ARM weighting method to the classification setting. When the number of models in the candidate-model set is fixed, the BMA weighting is consistent (and thus weakly consistent). From Yang (2007), when one properly chooses the data-splitting ratio, the ARM weighting can be consistent. More recently, Lai et al. (2015) proposed Fisher's fiducial-based methods for deriving probability density functions as weights on the set of candidate models. They showed that, under certain conditions, their method is consistent when p is diverging and the size of the true model is either fixed or diverging. Here, we consider only the ARM weighting and a weighting based on an information criterion.

Weighting using ARM for linear regression

To get the ARM weights, we randomly split the data $\mathbf{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ into a training set $\mathbf{D}_1$ and a test set $\mathbf{D}_2$ of (approximately) equal size. We train the linear regression model on $\mathbf{D}_1$ and evaluate its prediction performance on $\mathbf{D}_2$, based on which the weights $\mathbf{w} = \{w_1, \ldots, w_K\}$ can be computed. Let $\boldsymbol{\beta}_s^{(k)}$ be the sub-vector of $\boldsymbol{\beta}^{(k)}$ representing the nonzero coefficients of model $\mathcal{A}^k$, and let $\mathbf{x}_s^{(k)} \in \mathbb{R}^{|\mathcal{A}^k|}$ be the corresponding subset of selected predictors. When

p is large, the ARM weighting performs poorly in terms of measuring the model deviation. One way to fix this is to add a non-uniform prior $e^{-\psi C_k}$ to the weighting computation, with

$$C_k = s_k \log \frac{ep}{s_k} + 2\log(s_k + 2), \tag{2.7}$$

where s_k is the number of non-constant predictors for model k. The first term $s_k \log \frac{ep}{s_k}$ is an upper bound of $\log \binom{p}{s_k}$, which characterizes which model it is among the $\log \binom{p}{s_k}$ possibilities. This is followed by

$$\binom{p}{s_k} = \frac{\prod_{j=0}^{s_k-1}(p-j)}{s_k!} \leq \frac{p^{s_k}}{s_k!} \leq \left(\frac{pe}{s_k}\right)^{s_k}, \tag{2.8}$$

using Stirling's approximation. The second term in (2.7) represents the number of variables to be estimated. From an information-theoretic perspective, C_k can be regarded as an upper bound on the descriptive complexity of model $\mathcal{A}^k$. This concept plays a crucial role in model selection theory (Yang, 1999; Wang et al., 2014; Ye and Yang, 2019). In addition to this interpretation, one can also treat $e^{-\psi C_k}$ as the prior probability assigned to the models, from a Bayesian viewpoint. The constant $\psi > 0$ controls the relative importance of the prior weight on the final weights, which can be specified by the user. From a theoretical point of view, when ψ is bigger than 5.1, the complexity term is big enough to control the selection bias, and results in minimax optimal estimations (Yang, 1999). However, the bound 5.1 is more due to technical reasons. In practice, a smaller choice often works very well. Based on previous works (Nan and Yang, 2014; Ye et al., 2018; Ye and Yang, 2019) and our own numerical studies (see Section 2.8.6), we found that $\psi = 1$ or 2 often delivers the best numerical results.

The ARM weighting method for the linear regression models is summarized in Algo-

rithm 1.

Algorithm 1: ARM weighting procedure for linear regression.

Randomly split $\mathbf{D}$ into a training set $\mathbf{D}_1$ and a test set $\mathbf{D}_2$ of equal size.
For each $\mathcal{A}^k \in \mathbb{S}$, fit a standard linear regression of y on $\mathbf{x}_s^{(k)}$ using the training set $\mathbf{D}_1$ and get the estimated regression coefficient $\widehat{\boldsymbol{\beta}}_s^{(k)}$ and the estimated standard deviation $\widehat{\boldsymbol{\sigma}}_s^{(k)}$.
For each $\mathcal{A}^k$, compute the prediction $\mathbf{x}_s^{(k)\intercal}\widehat{\boldsymbol{\beta}}_s^{(k)}$ on the test set $\mathbf{D}_2$.
Compute the weight w_k for each candidate model $\mathcal{A}^k$:

$$w_k = \frac{e^{-\psi C_k}(\widehat{\boldsymbol{\sigma}}_s^{(k)})^{-n/2}\prod_{(\mathbf{x}_{si}^{(k)},y_i)\in\mathbf{D}_2}\exp(-(\widehat{\boldsymbol{\sigma}}_s^{(k)})^{-2}(y_i-\mathbf{x}_s^{(k)\intercal}\widehat{\boldsymbol{\beta}}_s^{(k)})^2/2)}{\sum_{l=1}^{K} e^{-\psi C_l}(\widehat{\boldsymbol{\sigma}}_s^{(l)})^{-n/2}\prod_{(\mathbf{x}_{si}^{(l)},y_i)\in\mathbf{D}_2}\exp(-(\widehat{\boldsymbol{\sigma}}_s^{(l)})^{-2}(y_i-\mathbf{x}_s^{(l)\intercal}\widehat{\boldsymbol{\beta}}_s^{(k)})^2/2)},$$

for $k = 1, \ldots, K$, where C_k, for $k = 1, \ldots, K$ is defined in (2.7).
Repeat the steps above (with random data splitting) L times to get $w_k^{(l)}$, for $l = 1, \ldots, L$, and get $w_k = \frac{1}{L}\sum_{l=1}^{L} w_k^{(l)}$.

Weighting using ARM for logistic regression

The ARM weighting method for logistic regression models is similar. We summarize it in Algorithm 2.

Algorithm 2: ARM weighting procedure for logistics regression.

Randomly split $\mathbf{D}$ into a training set $\mathbf{D}_1$ and a test set $\mathbf{D}_2$ of equal size.
For each $\mathcal{A}^k \in \mathbb{S}$, fit a standard logistic regression of y on $\mathbf{x}_s^{(k)}$ using the data in $\mathbf{D}_1$ and get the estimated conditional probability function $\hat{p}^{(k)}(\mathbf{x}_s^{(k)})$, for $k = 1, \ldots, K$,

$$\hat{p}^{(k)}(\mathbf{x}_s^{(k)}) \equiv \Pr(Y = 1 | X_s^{(k)} = \mathbf{x}_s^{(k)}) = \exp(\mathbf{x}_s^{(k)\intercal}\widehat{\boldsymbol{\beta}}_s^{(k)})/(1+\exp(\mathbf{x}_s^{(k)\intercal}\widehat{\boldsymbol{\beta}}_s^{(k)})).$$

For each $\mathcal{A}^k$, evaluate $\hat{p}^{(k)}(\mathbf{x}_s^{(k)})$ on the test set $\mathbf{D}_2$.
Compute the weight w_k for each model $\mathcal{A}^k$ in the candidate models:

$$w_k = \frac{e^{-\psi C_k}\prod_{(\mathbf{x}_{s,i}^{(k)},y_i)\in\mathbf{D}_2}\hat{p}^{(k)}(\mathbf{x}_{s,i}^{(k)})^{y_i}\left(1-\hat{p}^{(k)}(\mathbf{x}_{s,i}^{(k)})\right)^{1-y_i}}{\sum_{l=1}^{K} e^{-\psi C_l}\prod_{(\mathbf{x}_{s,i}^{(l)},y_i)\in\mathbf{D}_2}\hat{p}^{(l)}(\mathbf{x}_{s,i}^{(l)})^{y_i}\left(1-\hat{p}^{(l)}(\mathbf{x}_{s,i}^{(l)})\right)^{1-y_i}},\ k = 1, \ldots, K.$$

Repeat the steps above (with random data splitting) L times to get $w_k^{(l)}$, for $l = 1, \ldots, L$, and get $w_k = \frac{1}{L}\sum_{l=1}^{L} w_k^{(l)}$.

Weighting using modified BIC for linear and logistic regression

Information criteria such as the BIC can be used as alternative ways for computing the weights. Let ℓ_k be the maximized likelihood for model k. Recall that the BIC is $I_k^{\text{BIC}} = -2\log\ell_k + s_k \log n$. To accommodate the huge number of models, an extra term was added by Yang and Barron (1998) to reflect the additional price one needs to pay for searching through all the models. Including this extra term, we calculate the weights using a modified BIC (BIC-p) information criterion:

$$w_k = \exp(-I_k/2 - \psi C_k) / \sum_{l=1}^{K} \exp(-I_l/2 - \psi C_l),\ k = 1, \ldots, K. \tag{2.9}$$

2.5 Simulation

In order to study the performance of the estimated F- and G-measures, we conduct simulations for several well-known variable selection methods (for both regression and classification) under various settings. We consider numerical experiments for both the $n < p$ and the $n \geq p$ cases, with specified structural feature correlation (i.e., independent/correlated). We also consider special settings of the true coefficients, such as decaying coefficients.

2.5.1 Setting I: regression models

For the regression case, the response Y is generated from the following model:

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

where $\epsilon \sim N(0, \sigma^2)$. To study how the estimation performance varies with the noise level σ^2, we choose nine σ-values, evenly spaced between 0.01 and 5. The predictors $\mathbf{x}_i$ and the coefficient vector $\boldsymbol{\beta}$ are generated according to the following settings:

Example 2.1

$n = 200$, $p = 8$, $\boldsymbol{\beta} = (3, 1.5, 2, 0, 0, 0, 0, 0)^\intercal$. Predictors $\mathbf{x}_i$, for $i = 1, \ldots, n$ are generated as n independent and identically distributed (i.i.d.) observations from $N(0, \boldsymbol{I}_p)$. □

Example 2.2

Same as Example 2.1, except $n = 1000$. □

Example 2.3

$n = 200$, $p = 2000$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\intercal$, where $(\beta_1, \beta_2, \beta_3) = (3, 1.5, 2)$ and $(\beta_4, \ldots, \beta_{2000})$ are zero. Predictors $\mathbf{x}_i$, for $i = 1, \ldots, n$, are sampled as n i.i.d. observations from $N(0, \boldsymbol{I}_p)$. □

Example 2.4

$n = 200$, $p = 30$, components 1–5 of $\boldsymbol{\beta}$ are 10.5, components 6–10 are 5.5, components 11–15 are 0.5, and the rest are zero. Therefore, there are 15 nonzero predictors, including five large ones, five moderate ones, and five small ones. Predictors $\mathbf{x}_i$, for $i = 1, \ldots, n$, are generated from $X \sim N_p(0, \Sigma)$, with $\Sigma = (0.4^{|j-k|})_{p \times p}$; thus, the pairwise correlation between X_j and X_k is $0.4^{|j-k|}$. □

Example 2.5

$n = 200$, $p = 200$, components 1–5 of $\boldsymbol{\beta}$ are 10.5, components 6–10 are 5.5, components 11–15 are 0.5, and the rest are zero. Predictors $\mathbf{x}_i$, for $i = 1, \ldots, n$, are generated from $X \sim N_p(0, \Sigma)$. The covariance structure Σ is set as follows: the first 15 predictors $(X_1, \ldots, X_{15})$ and the remaining 185 predictors $(X_{16}, \ldots, X_{200})$ are independent. The pairwise correlation between X_j and X_k in $(X_1, \ldots, X_{15})$ is $0.4^{|j-k|}$, with $j, k = 1, \ldots, 15$. The pairwise correlation between X_j and X_k in $(X_{16}, \ldots, X_{200})$ is $0.4^{|j-k|}$, with $j, k = 16, \ldots, 200$. □

We apply four penalized methods, namely, the Lasso, adaptive Lasso, MCP, and SCAD

to the data from Examples 1–5, and denote the resulting models as $\mathcal{A}^{\text{Lasso}}$, $\mathcal{A}^{\text{AdLasso}}$, $\mathcal{A}^{\text{MCP}}$, and $\mathcal{A}^{\text{SCAD}}$, respectively. We use **glmnet** to compute $\mathcal{A}^{\text{Lasso}}$ and $\mathcal{A}^{\text{AdLasso}}$, and **ncvreg** for computing $\mathcal{A}^{\text{MCP}}$ and $\mathcal{A}^{\text{SCAD}}$. Five-fold cross-validation is used for penalty parameter tuning in all these procedures. Because we know the true model $\mathcal{A}^* = \{j : \beta_j \neq 0\}$ in the simulation, we can report the true $F(\mathcal{A}^0)$ and $G(\mathcal{A}^0)$ measures for each model $\mathcal{A}^0 \in \{\mathcal{A}^{\text{Lasso}}, \mathcal{A}^{\text{AdLasso}}, \mathcal{A}^{\text{MCP}}, \mathcal{A}^{\text{SCAD}}\}$. For comparison, we also compute the estimated $\widehat{F}$ and $\widehat{G}$ using two different weighting methods, the ARM and the BIC-p (modified BIC), with prior adjustment $\psi = 1$. The number of observations in the training set used to compute the ARM weight is half of the sample size $\lfloor n/2 \rfloor$, and the corresponding repetition time is 100.

All simulation cases are repeated 100 times, and the corresponding values are computed and averaged. We compare $\widehat{F}(\mathcal{A}^0)$ and $\widehat{G}(\mathcal{A}^0)$ with the true $F(\mathcal{A}^0)$ and $G(\mathcal{A}^0)$ in Figure 2.1 for Example 2.1, and in Figures 2.2–2.5 of Section 2.8 for Examples 2.2–2.5. Overall, $\widehat{F}(\mathcal{A}^0)$ and $\widehat{G}(\mathcal{A}^0)$ using the ARM and the BIC-p weightings well reflect the trends of $F(\mathcal{A}^0)$ and $G(\mathcal{A}^0)$, in the sense that both the true curves and the estimated curves trend down as σ^2 increases. Furthermore, the estimation accuracy drops as σ^2 increases. The estimated $\widehat{F}(\mathcal{A}^0)$ and $\widehat{G}(\mathcal{A}^0)$ properly reflect the true performance of a given $\mathcal{A}^0$. For example, in Figures 2.3, 2.4, and 2.5, we see that the performance of the Lasso deteriorates significantly as σ^2 increases, because it tends to over-select variables under higher noise levels. In contrast, the adaptive Lasso, MCP, and SCAD have more robust performance against high noise levels. $\widehat{F}(\mathcal{A}^0)$ and $\widehat{G}(\mathcal{A}^0)$ correctly reflect these aforementioned facts. From the results, we find that the MCP performs best, with the highest true/estimated F- and G-measures in Examples 2.2–2.5, while the adaptive Lasso performs best in Example 2.1.

Comparing Figures 2.1 and 2.2, we see that the sample size influences the estimation performance: large samples produce more accurate $\widehat{F}(\mathcal{A}^0)$ and $\widehat{G}(\mathcal{A}^0)$. The gain in the

estimation accuracy from an increased sample size is because more information results in better assigned weights on the candidate models.

In Figure 2.5, the over-estimation in the adaptive Lasso, SCAD, and MCP when σ is large occurs because highly weighted candidate models miss several small coefficients variables, which is caused by the decaying coefficients, and worsened by correlation between the variables. For the Lasso, when σ is small, PAVI identifies good candidate models on which to put high weights. Thus, the estimation is good. When σ is larger, the candidate models with high weights miss several true variables. At the same time, the Lasso chooses more redundant variables as σ becomes larger. Therefore, the precision is under-estimated, as is the F-measure.

2.5.2 Setting II: classification models

For the classification case, we randomly generate n i.i.d observations $\{y_i, \mathbf{x}_i\}_{i=1}^n$. Each binary response $y_i \in \{0, 1\}$ is generated from a Bernoulli distribution with conditional probability $\Pr(Y = 1|X = \mathbf{x}_i) = \exp(\mathbf{x}_i^\intercal \boldsymbol{\beta})/\{1 + \exp(\mathbf{x}_i^\intercal \boldsymbol{\beta})\}$. The explanatory variables X and the coefficient vector $\boldsymbol{\beta}$ are set under the same configurations as in Example 2.1–Example 2.5.

The absolute differences between the true and estimated measures,

$$d_F = |\widehat{F}(\mathcal{A}^0) - F(\mathcal{A}^0)| \quad \text{and} \quad d_G = |\widehat{G}(\mathcal{A}^0) - G(\mathcal{A}^0)|,$$

are used to evaluate the estimation performance, where a smaller d_F and d_G indicate better estimation performance.

All simulation cases are repeated 100 times, and the corresponding $F(\mathcal{A}^0)$, $G(\mathcal{A}^0)$, $\widehat{F}(\mathcal{A}^0)$, $\widehat{G}(\mathcal{A}^0)$, d_F, and d_G values are computed and averaged. The results are summarized in Table 2.1 for Example 1, and in Tables 2.7–2.10 of Section 2.8 for Examples 2–5. The standard

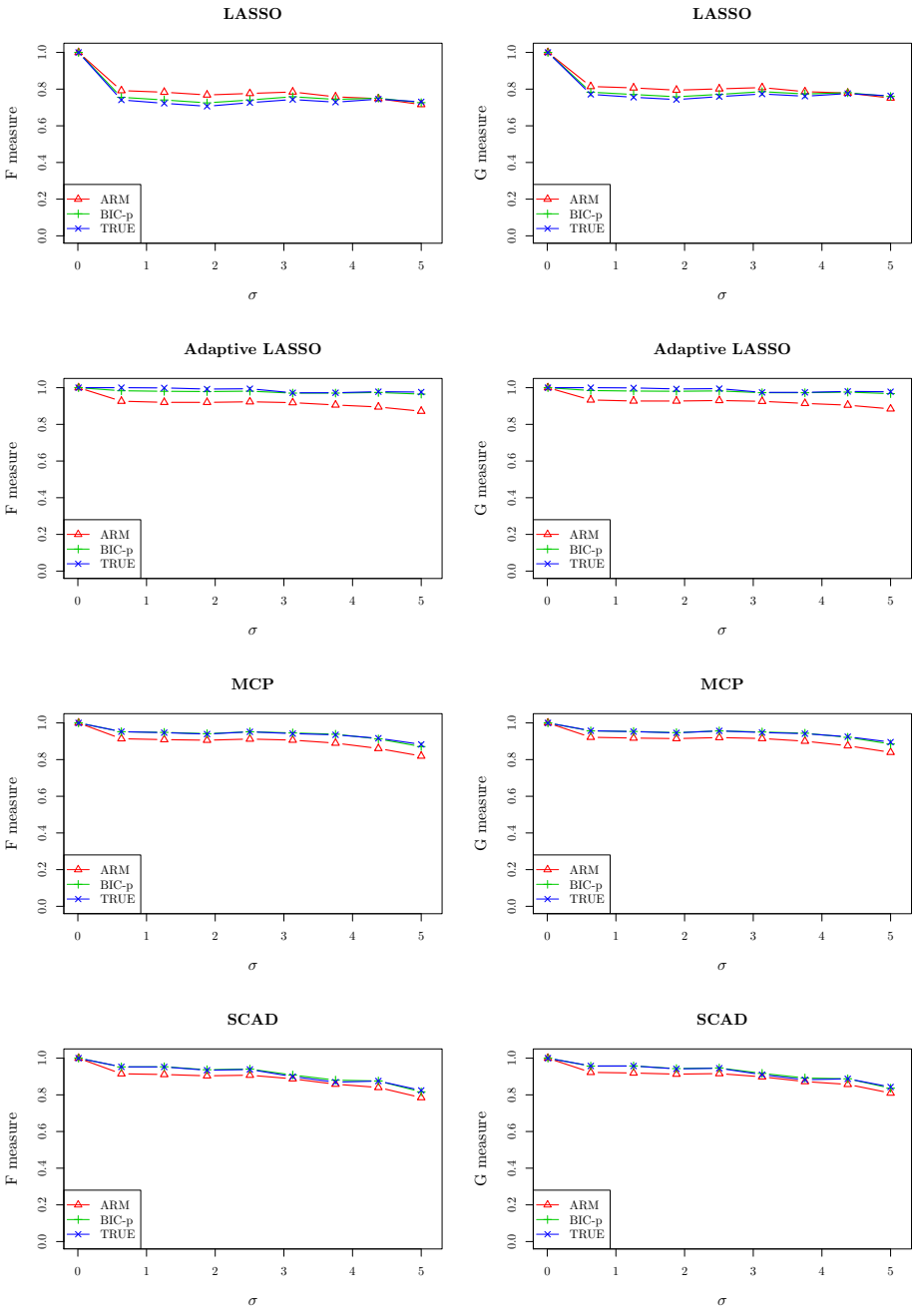

Figure 2.1: Regression case (Example 2.1).

errors are also shown (in parentheses). As shown in the tables, d_F and d_G are generally small, which indicates that the estimated $\widehat{F}(\mathcal{A}^0)$ and $\widehat{G}(\mathcal{A}^0)$ are good approximations of the true $F(\mathcal{A}^0)$ and $G(\mathcal{A}^0)$, respectively. The estimated $\widehat{F}(\mathcal{A}^0)$ and $\widehat{G}(\mathcal{A}^0)$ reflect the true advantage of a given variable selection method. For example, in Table 2.1, and in Tables 2.7–2.10, the adaptive Lasso, MCP, and SCAD have better variable selection performance than that of the Lasso, according to their larger true values of $F(\mathcal{A}^0)$ and $G(\mathcal{A}^0)$. The estimated $\widehat{F}(\mathcal{A}^0)$ and $\widehat{G}(\mathcal{A}^0)$ correctly reflect these differences in performance.

Our estimation method still performs very well in the high-dimensional setting, as can be seen from the small d_F and d_G in Table 2.8. However, the results from Tables 4 and 5 show that the decaying coefficients and feature correlation make the estimation of $\widehat{F}(\mathcal{A}^0)$ and $\widehat{G}(\mathcal{A}^0)$ more difficult. In these two cases, the BIC-p methods tend to overestimate $F(\mathcal{A}^0)$ and $G(\mathcal{A}^0)$ for the MCP and SCAD, whereas the ARM tends to underestimate $F(\mathcal{A}^0)$ and $G(\mathcal{A}^0)$ for the Lasso and adaptive Lasso.

The overestimation problem of the BIC-p method mainly comes from that of the recall part. The final model selected by the SCAD misses several true variables; thus, the true recall is very small. However, if we were to use the heavily weighted candidate models that miss several true variables in the PAVI calculation, the recall would be overestimated.

For the SCAD and ARM combination, using the heavily weighted models that miss several true variables in PAVI will overestimate of the recall and underestimate the precision, although these two effects cancel each other to some degree.

The underestimation by the ARM methods mainly comes from that of the precision part, while the estimated recall is close to (slightly overestimates) the true recall. The Lasso tends to miss true variables and over-select redundant variables in the examples. Thus, the true precision of the Lasso is small.

Table 2.1: Classification case (Example 2.1).

	F	G	d_F	d_G
		Lasso		
True	0.670 (0.010)	0.712 (0.009)		
ARM	0.711 (0.009)	0.747 (0.007)	0.046 (0.003)	0.039 (0.002)
BIC-p	0.687 (0.010)	0.726 (0.008)	0.017 (0.002)	0.014 (0.001)
		AdLasso		
True	0.944 (0.009)	0.949 (0.008)		
ARM	0.899 (0.004)	0.908 (0.004)	0.066 (0.003)	0.060 (0.003)
BIC-p	0.946 (0.007)	0.950 (0.007)	0.018 (0.002)	0.016 (0.001)
		MCP		
True	0.968 (0.009)	0.971 (0.008)		
ARM	0.903 (0.005)	0.913 (0.004)	0.079 (0.003)	0.072 (0.002)
BIC-p	0.961 (0.007)	0.965 (0.006)	0.019 (0.002)	0.017 (0.001)
		SCAD		
True	0.902 (0.012)	0.911 (0.010)		
ARM	0.881 (0.006)	0.892 (0.006)	0.054 (0.003)	0.050 (0.003)
BIC-p	0.911 (0.010)	0.919 (0.009)	0.018 (0.002)	0.016 (0.001)

For the Lasso and BIC combination, using the heavily weighted models that miss several true variables with small coefficients in PAVI overestimates the recall and underestimates the precision, although these two effects cancel each other to some degree.

Both issues are mainly caused by the fact that the candidate models with large weights cannot recover all the variables with small true coefficients. Then, the problem is worsened by the high correlation between the features.

2.6 Real Data

In this section, we apply PAVI using candidate models from several model selection methods to gene expression data for cancer-related biomarker identification. The biomarker selection process is usually under a high-dimensional, small-sample, and high-noise setting involving highly correlated genes (Golub et al., 1999; West et al., 2001). As such, the sets of genes identified may be subject to substantial changes, owing to small perturbations in the data (Baggerly et al., 2004; Henry and Hayes, 2012). Here, we use $\widehat{F}$ and $\widehat{G}$ to evaluate

Table 2.2: Summary of `Colon`, `Leukemia`, `Prostate`.

Data	n	n_1 ($y = 1$)	n_2 ($y = 0$)	p (number of genes)	Data source
`Colon`	62	40	22	2000	Alon et al. (1999)
`Leukemia`	72	25	47	7129	Golub et al. (1999)
`Prostate`	102	52	50	12600	Singh et al. (2002)

such selection uncertainty.

Our goal is to provide a serious and careful analysis of the outcomes of several variable selection methods from multiple angles to understand the key issues of interest. We hope our analysis provides strong enough evidence that the estimated F and G values yield valuable information.

2.6.1 Data description

We consider three well-studied benchmark cancer data sets: `Colon` (Alon et al., 1999), `Leukemia` (Golub et al., 1999), and `Prostate` (Singh et al., 2002). Table 2.2 provides a brief summary.

2.6.2 Methods/models to be examined

Using the three datasets, we compare the variable selection performance of four commonly used penalization methods: the Lasso, adaptive Lasso, MCP, and SCAD. We first obtain the final model $\mathcal{A}^0$ for each method, where the tuning parameter λ is selected using five-fold cross-validation. Then, we use PAVI to estimate $\widehat{F}(\mathcal{A}^0)$ and $\widehat{G}(\mathcal{A}^0)$ with two weighting schemes, ARM and BIC-p. The procedure is repeated 100 times to average out randomness in the tuning parameter selection, and the averages of $\widehat{F}(\mathcal{A}^0)$, $\mathrm{sd}\big(\widehat{F}(\mathcal{A}^0)\big)$ and $\widehat{G}(\mathcal{A}^0)$, $\mathrm{sd}\big(\widehat{G}(\mathcal{A}^0)\big)$ are summarized in Tables 2.3, 2.4, and 2.11. For comparison, we also include several other models studied in the existing literature. Specifically, we consider

Leung and Hung, 2010 (L10), Yang and Song, 2010 (Y10), Chandra and Gupta, 2011 (C11), and Lee and Leu, 2011 (L11) for `Colon`, Leung and Hung, 2010 (L10), Yang and Song, 2010 (Y10), and Ji et al., 2011 (J11; two kinds of models are provided using different importance criteria in this work, denoted by J11^1 and J11^2, respectively) for `Leukemia`, and Leung and Hung, 2010 (L10) and Sharma et al., 2012 (S12) for `Prostate`.

Y10, J11, and S12 use linear-based variable selection techniques without initial variable screening. Specifically, Y10 uses a probit regression model, J11 uses a linear kernel support vector classifier (SVC), and S12 uses the linear discriminant analysis (LDA) technique with nearest centroid classifier (NCC). In contrast, L10, C11, and L11 use nonparametric variable selection techniques: L10 uses a support vector machine (SVM), C11 uses a naïve Bayes classifier (NBC), and SVM, and L11 uses an SVM. In addition, we consider the importance screening method (ImpS) of Ye et al. (2018), which uses sparsity-oriented importance learning for variable screening.

2.6.3 Results

The estimated $\widehat{F}$ and $\widehat{G}$ of each model on `Colon`, `Leukemia`, and `Prostate` are reported in Tables 2.3, 2.4, and 2.11 (in Section 2.8), respectively. We find that ImpS achieves almost the largest estimated $\widehat{F}$ and $\widehat{G}$ on all three data sets. L10 has basically zero $\widehat{F}$ and $\widehat{G}$ for `Colon` and `Prostate`. J11^1 and J11^2 have basically zero $\widehat{F}$ and $\widehat{G}$ for `Leukemia`. (These cases are shown in bold in Tables 2.3, 2.4, and 2.11.) This suggests that, from a logistic regression modeling perspective, they may have chosen "wrong" variables and have very low recall or precision.

Table 2.3: Estimated F- and G-measures and standard deviations for `Colon`. L10 has numerically zero $\widehat{F}$ and $\widehat{G}$ values (shown in bold).

	ARM				BIC-p			
	F	*sd.F*	*G*	*sd.G*	*F*	*sd.F*	*G*	*sd.G*
Lasso	0.147	0.024	0.280	0.022	0.205	0.066	0.332	0.058
AdLasso	0.194	0.165	0.255	0.211	0.309	0.191	0.361	0.209
MCP	0.349	0.045	0.459	0.035	0.460	0.130	0.544	0.093
SCAD	0.149	0.032	0.274	0.039	0.211	0.074	0.331	0.071
ImpS	0.524	0.081	0.596	0.065	0.656	0.176	0.698	0.118
L11	0.111	0.110	0.175	0.175	0.112	0.105	0.157	0.151
Y10	0.103	0.017	0.233	0.018	0.146	0.048	0.276	0.047
C11	0.184	0.020	0.317	0.022	0.223	0.076	0.333	0.082
L10	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**

Table 2.4: Estimated F- and G-measures and standard deviations for `Leukemia`. J11[1] and J11[2] have numerically zero $\widehat{F}$ and $\widehat{G}$ values (shown in bold).

	ARM				BIC-p			
	F	*sd.F*	*G*	*sd.G*	*F*	*sd.F*	*G*	*sd.G*
Lasso	0.083	0.025	0.206	0.026	0.079	0.012	0.203	0.014
AdLasso	0.323	0.044	0.432	0.031	0.322	0.039	0.434	0.033
MCP	0.168	0.170	0.221	0.210	0.061	0.089	0.078	0.108
SCAD	0.094	0.028	0.220	0.028	0.090	0.013	0.216	0.015
ImpS	0.525	0.065	0.591	0.042	0.573	0.129	0.636	0.102
J11[1]	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
J11[2]	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
Y10	0.108	0.014	0.236	0.009	0.105	0.002	0.233	0.012
L10	0.212	0.180	0.265	0.224	0.336	0.089	0.419	0.110

2.6.4 Are the zero $\widehat{F}$ and $\widehat{G}$ values too harsh for the methods?

It is striking that the $\widehat{F}$ and $\widehat{G}$ values for some selections are numerically zero, which seems rather extreme. Does this mean those models are truly poor, or does it mean our performance assessment methodology fails? We examine the matter from three perspectives.

First perspective: the labels of the selected genes

First, let us examine the labels of the selected genes. We obtain the selected genes in the literature. We use five-fold cross-validation for the penalty parameter tuning to obtain selected genes for the penalized regression models. In Tables 2.12, 2.13, and 2.14, the results show that the genes selected by L10 (`Colon` and `Prostate`), J11^1, and J11^2 (`Leukemia`) are mostly not supported by other models. More specifically, the choices of variables by L10, J11^1, and J11^2 in those cases share zero, one, or at most two genes with the other methods, respectively. (These cases are underlined in Tables 2.12, 2.13, and 2.14.)

Second perspective: predictive accuracy

Here, we examine the issue from a predictive accuracy perspective. We randomly split the data set into 4/5 observations for training, and 1/5 observations for testing. We fit the SVM models using the selected genes on the training data using **kernlab** (Karatzoglou et al., 2004), and evaluate the predictive accuracy on the testing data. The procedure is repeated 100 times, and the averaged classification accuracy and "standard errors" (w.r.t. the permutations) are recorded in Table 2.5. Alternatively, we may consider parametric models. We fit the logistic regression using the selected genes (in Table 2.5). We find that L10, J11^1, and J11^2 have worse predictive accuracy (shown in bold in Table 2.5) than that

Table 2.5: Comparisons of classification accuracy on `Colon`, `Leukemia`, and `Prostate` using a logistic regression and an SVM.

Logistic Model					
`Colon`		`Leukemia`		`Prostate`	
ImpS	86.3 (0.8)	ImpS	97.1 (0.3)	ImpS	94.0 (0.4)
Lasso	80.0 (1.0)	Lasso	99.8 (0.1)	Lasso	97.0 (0.4)
AdLasso	85.5 (0.8)	AdLasso	93.9 (0.5)	AdLasso	99.8 (0.1)
MCP	85.1 (0.8)	MCP	99.5 (0.1)	MCP	98.7 (0.2)
SCAD	84.3 (0.8)	SCAD	97.9 (0.3)	SCAD	97.1 (0.2)
L11	80.4 (0.8)	J11[1]	**89.4 (0.8)**	S12	96.5 (0.4)
Y10	90.9 (0.9)	J11[2]	**89.8 (0.7)**	L10	**59.0 (0.8)**
C11	79.6 (1.0)	Y10	91.2 (0.7)		
L10	**83.0 (0.9)**	L10	95.5 (0.4)		
SVM Model					
`Colon`		`Leukemia`		`Prostate`	
ImpS	84.0 (0.9)	ImpS	97.6 (0.3)	ImpS	95.3 (0.4)
Lasso	75.8 (0.9)	Lasso	99.1 (0.2)	Lasso	96.3 (0.4)
AdLasso	79.0 (0.9)	AdLasso	95.8 (0.4)	AdLasso	96.6 (0.3)
MCP	83.1 (1.1)	MCP	99.0 (0.2)	MCP	97.1 (0.3)
SCAD	86.0 (0.9)	SCAD	99.1 (0.2)	SCAD	96.4 (0.3)
L11	79.0 (1.1)	J11[1]	**88.6 (0.8)**	S12	95.5 (0.4)
Y10	78.3 (1.0)	J11[2]	**87.4 (0.9)**	L10	**59.3 (0.9)**
C11	77.1 (0.9)	Y10	90.2 (0.6)		
L10	**72.4 (0.9)**	L10	92.2 (0.6)		

of the simpler model selected by ImpS, supporting the validity of their low $\widehat{F}$ and $\widehat{G}$ values.

Third perspective: traditional model fitting

For the third perspective, we investigate the AIC, BIC, and deviance measures. When comparing models fitted using the maximum likelihood to the same data, the smaller the AIC or BIC value, the better is the model.

From Table 2.6, the model for `Colon` with zero $\widehat{F}$ and $\widehat{G}$ values also has relatively large AIC, BIC, and deviance values (shown in bold in the table) compared with those of the models with large $\widehat{F}$ and $\widehat{G}$ values. The results are similar for the other two data sets, except that the deviance values for `Leukemia` are extremely small, owing to the easy classification nature of the data.

In summary, we see that the low (near zero) $\widehat{F}$ and $\widehat{G}$ values for the investigated sets of

Table 2.6: Estimated AIC, BIC, and deviance for `Colon`, `Leukemia`, and `Prostate`.

`Colon`				`Leukemia`				`Prostate`			
	AIC	BIC	Dev.		AIC	BIC	Dev.		AIC	BIC	Dev.
Lasso	26.0	53.6	0.0		56.0	119.7	0.0		62.0	143.3	0.0
AdLasso	34.9	49.8	20.9		12.0	25.6	0.0		22.0	50.8	0.0
MCP	32.1	44.9	20.1		16.0	34.2	0.0		16.0	36.9	0.0
SCAD	26.0	53.6	0.0		48.0	102.6	0.0		38.0	87.8	0.0
ImpS	35.5	44.1	27.5		8.0	17.1	0.0		12.0	27.7	9.4
L11	51.4	70.5	33.4	J11[1]	**20.0**	**42.7**	**0.0**	S12	36.1	49.2	26.1
Y10	40.0	82.5	0.0	J11[2]	**18.0**	**38.4**	**0.0**	L10	**140.1**	**158.5**	**126.1**
C11	45.2	68.6	23.2	Y10	38.0	81.2	0.0				
L10	**48.6**	**63.5**	**34.6**	L10	10.0	21.3	0.0				

selected genes are supported from the three perspectives. Our PAVI approach provides a valid tool for checking the reliability and reproducibility of a given set of selected variables when the true model is not known. To be fair, we want to emphasize that the poor $\widehat{F}$ and $\widehat{G}$ values of some of the selection methods are based on the logistic regression perspective, although Table 2.5 seems to suggest that a logistic regression works at least as well as an SVM.

2.7 Conclusion

Despite there being many variable selection methods, most investigations of their behaviors are limited to theoretical studies and somewhat scattered simulation results, which may have little to do with a specific data set. There is a severe lack of valid performance measures that are computable based on data alone. This leads to the pessimistic view that for real data, nothing can be said strongly about which method is better for describing the data generation mechanism since no one knows the truth. Sound implementable variable selection diagnostic tools can provide insight into the matter.

Nan and Yang (2014) proposed an approach to investigate how many variables are likely missed, and how many are not quite justifiable for an outcome of a variable selection

process. In real applications, it is often of interest and important to summarize the two types of selection errors into a single measure to characterize the behavior of a variable selection method. As a result, F- and G-measures are gaining in popularity in the model selection literature. If we are given a data set for which several model selection methods are considered, prior to this work, the available model diagnostic tools could only tell us (a) which methods were more unstable, and (b) how many terms are likely missed or unsupported. This information, unlike the F- and G-measures, may not be enough to give one a good sense of the overall model selection performance. In this study, we have advanced this line of research on model selection diagnostics by providing a valid estimation of F- and G-measures.

We have proved that the estimated F- and G-measures are uniformly consistent, as long as the weighting is weakly consistent. The simulation results clearly show that the $\widehat{F}$ and $\widehat{G}$ values based on our PAVI approach nicely characterize the overall performance of the model selection outcomes. This information can be used to compare different methods for the data at hand.

We used three real-data examples to demonstrate the utility of our PAVI methodology. Many variable selection results have been reported in the literature based on these data sets. A careful study with multiple perspectives has provided strong evidence to suggest that some of the variable selection outcomes may be far removed from the best set of variables to use for a logistic regression or an SVM with the given information.

2.8 Proofs and Additional Numerical Studies

In this section, we provide technical proofs for the theorems in the previous sections of this chapter with additional remarks, and give further numerical results, including one on

sensitivity of the complexity parameter ψ and one on the impact of the candidate models.

2.8.1 Proof of Theorem 2.1

Part I: F-measure

Proof Denote by ∇ the symmetric difference between two sets. Estimated F-measure can be rewritten as

$$\widehat{F}(\mathcal{A}^0) = \sum_k w_k F(\mathcal{A}^0; \mathcal{A}^k), \qquad F(\mathcal{A}^0; \mathcal{A}^k) = \frac{|\mathcal{A}^0| + |\mathcal{A}^k| - |\mathcal{A}^0 \nabla \mathcal{A}^k|}{|\mathcal{A}^0| + |\mathcal{A}^k|}.$$

We have

$$\begin{aligned}
&|\widehat{F}(\mathcal{A}^0) - F(\mathcal{A}^0)| = \left| \sum_k w_k F(\mathcal{A}^0; \mathcal{A}^k) - F(\mathcal{A}^0) \right| = \left| \sum_k w_k (F(\mathcal{A}^0; \mathcal{A}^k) - F(\mathcal{A}^0)) \right| \\
&\leq \sum_k w_k |F(\mathcal{A}^0; \mathcal{A}^k) - F(\mathcal{A}^0)| = \sum_k w_k \left| 1 - \frac{|\mathcal{A}^0 \nabla \mathcal{A}^k|}{|\mathcal{A}^0| + |\mathcal{A}^k|} - 1 + \frac{|\mathcal{A}^0 \nabla \mathcal{A}^*|}{|\mathcal{A}^0| + |\mathcal{A}^*|} \right| \\
&= \sum_k w_k \left| \frac{|\mathcal{A}^0| \cdot (|\mathcal{A}^0 \nabla \mathcal{A}^*| - |\mathcal{A}^0 \nabla \mathcal{A}^k|) + |\mathcal{A}^k| \cdot |\mathcal{A}^0 \nabla \mathcal{A}^*| - |\mathcal{A}^*| \cdot |\mathcal{A}^0 \nabla \mathcal{A}^k|}{(|\mathcal{A}^0| + |\mathcal{A}^k|)(|\mathcal{A}^0| + |\mathcal{A}^*|)} \right| \\
&\leq \underbrace{\sum_k w_k \frac{|\mathcal{A}^0| \cdot ||\mathcal{A}^0 \nabla \mathcal{A}^*| - |\mathcal{A}^0 \nabla \mathcal{A}^k||}{(|\mathcal{A}^0| + |\mathcal{A}^k|)(|\mathcal{A}^0| + |\mathcal{A}^*|)}}_{A} + \underbrace{\sum_k w_k \frac{|\mathcal{A}^k| \cdot ||\mathcal{A}^0 \nabla \mathcal{A}^*| - |\mathcal{A}^0 \nabla \mathcal{A}^k||}{(|\mathcal{A}^0| + |\mathcal{A}^k|)(|\mathcal{A}^0| + |\mathcal{A}^*|)}}_{B} \\
&+ \underbrace{\sum_k w_k \frac{||\mathcal{A}^k| - |\mathcal{A}^*|| \cdot |\mathcal{A}^0 \nabla \mathcal{A}^k|}{(|\mathcal{A}^0| + |\mathcal{A}^k|)(|\mathcal{A}^0| + |\mathcal{A}^*|)}}_{C}.
\end{aligned}$$

For ease of notation, we divide the right-most hand side of the above inequality into three parts and denote them by A, B, and C respectively. Note that since $\big||\mathcal{A}^0 \nabla \mathcal{A}^*| - |\mathcal{A}^0 \nabla \mathcal{A}^k|\big| \leq |\mathcal{A}^* \nabla \mathcal{A}^k|$, we have

$$A \leq \sum_k w_k \frac{|\mathcal{A}^0| \cdot |\mathcal{A}^* \nabla \mathcal{A}^k|}{(|\mathcal{A}^0| + |\mathcal{A}^k|)(|\mathcal{A}^0| + |\mathcal{A}^*|)} \leq \sum_k w_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|}.$$

Similarly, it can be shown that

$$B \leq \sum_k w_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|}.$$

Let us now prove a similar bound also holds for C. Specifically, we have

$$\begin{aligned} C &= \sum_k w_k \frac{||\mathcal{A}^k| - |\mathcal{A}^*|| \cdot |\mathcal{A}^0 \nabla \mathcal{A}^k|}{(|\mathcal{A}^0| + |\mathcal{A}^k|)(|\mathcal{A}^0| + |\mathcal{A}^*|)} \leq \sum_k w_k \frac{\big||\mathcal{A}^k| - |\mathcal{A}^*|\big|}{|\mathcal{A}^0| + |\mathcal{A}^*|} \\ &= \sum_k w_k \frac{\big|(|\mathcal{A}^k \backslash \mathcal{A}^*| + |\mathcal{A}^k \cap \mathcal{A}^*|) - (|\mathcal{A}^* \backslash \mathcal{A}^k| + |\mathcal{A}^k \cap \mathcal{A}^*|)\big|}{|\mathcal{A}^0| + |\mathcal{A}^*|} \\ &= \sum_k w_k \frac{\big||\mathcal{A}^k \backslash \mathcal{A}^*| - |\mathcal{A}^* \backslash \mathcal{A}^k|\big|}{|\mathcal{A}^0| + |\mathcal{A}^*|} \leq \sum_k w_k \frac{|\mathcal{A}^k \backslash \mathcal{A}^*| + |\mathcal{A}^* \backslash \mathcal{A}^k|}{|\mathcal{A}^0| + |\mathcal{A}^*|} \\ &= \sum_k w_k \frac{|\mathcal{A}^k \nabla \mathcal{A}^*|}{|\mathcal{A}^0| + |\mathcal{A}^*|} \leq \sum_k w_k \frac{|\mathcal{A}^k \nabla \mathcal{A}^*|}{|\mathcal{A}^*|}. \end{aligned}$$

It follows that for any $\mathcal{A}^0$ in $\mathbb{C}$

$$|\widehat{F}(\mathcal{A}^0) - F(\mathcal{A}^0)| \leq A + B + C \leq 3 \sum_k w_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|}.$$

Therefore,

$$\sup_{\mathcal{A}^0 \in \mathbb{C}} |\widehat{F}(\mathcal{A}^0) - F(\mathcal{A}^0)| \leq 3 \sum_k w_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|}.$$

Now under the assumption that the model weighting w is weakly consistent,

$$\sum_k w_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|} \xrightarrow{p} 0.$$

We have proved $\sup_{\mathcal{A}^0 \in \mathbb{C}} |\widehat{F}(\mathcal{A}^0) - F(\mathcal{A}^0)| \xrightarrow{p} 0$. □

Part II: G-measure

Proof For a given $\mathcal{A}^0$ in $\mathbb{C}$, the estimated G-measure can be rewritten as

$$\widehat{G}(\mathcal{A}^0) = \sum_k w_k G(\mathcal{A}^0; \mathcal{A}^k), \qquad G(\mathcal{A}^0; \mathcal{A}^k) = \frac{|\mathcal{A}^0| + |\mathcal{A}^k| - |\mathcal{A}^0 \nabla \mathcal{A}^k|}{2\sqrt{|\mathcal{A}^0| \cdot |\mathcal{A}^k|}}.$$

Suppose $|\widehat{G}(\mathcal{A}^0)-G(\mathcal{A}^0)|$ does not converge to 0 in probability uniformly over $\mathbb{C}$, then there exist some subsequence $n_1, n_2, \cdots$, $\epsilon_1 > 0, \delta > 0$, $\mathcal{A}^0_{n_j} \in \mathbb{C}$, and sets $\mathcal{S}_{n_j}$, s.t. $P(\mathcal{S}_{n_j}) \geq \delta$ and $|\widehat{G}(\mathcal{A}^0_{n_j}) - G(\mathcal{A}^0_{n_j})| > \epsilon_1$ on $\mathcal{S}_{n_j}$. For ease of notation, we denote $\mathcal{A}^0_{n_j}$ as $\mathcal{A}^0$ in the following proof.

With the above, we first prove that we must have $\frac{|\mathcal{A}^0|}{|\mathcal{A}^*|} \xrightarrow{p} 0$ on $\mathcal{S}_{n_j}$ as $n_j \to \infty$. If not, then there exist $\epsilon_2 > 0$, a subsequence n_{j_l} and sets $\mathcal{N}_{n_{j_l}}$ such that on $\mathcal{N}_{n_{j_l}}$ we have $\frac{|\mathcal{A}^0|}{|\mathcal{A}^*|} > \epsilon_2 > 0$. Then we can actually prove $|\widehat{G}(\mathcal{A}^0) - G(\mathcal{A}^0)| \xrightarrow{p} 0$ on $\mathcal{N}_{n_{j_l}}$ as follows.

By definition of $\widehat{G}$ and G, and $\frac{|\mathcal{A}^0|}{|\mathcal{A}^*|} > \epsilon_2 > 0$ on $\mathcal{N}_{n_{j_l}}$, we have

$$
\begin{aligned}
|\widehat{G}(\mathcal{A}^0) - G(\mathcal{A}^0)| &= |\sum_k w_k G(\mathcal{A}^0; \mathcal{A}^k) - G(\mathcal{A}^0)| \leq \sum_k w_k |G(\mathcal{A}^0; \mathcal{A}^k) - G(\mathcal{A}^0)| \\
&= \sum_k w_k \left| \frac{|\mathcal{A}^0| + |\mathcal{A}^k| - |\mathcal{A}^0 \nabla \mathcal{A}^k|}{2\sqrt{|\mathcal{A}^0| \cdot |\mathcal{A}^k|}} - \frac{|\mathcal{A}^0| + |\mathcal{A}^*| - |\mathcal{A}^0 \nabla \mathcal{A}^*|}{2\sqrt{|\mathcal{A}^0| \cdot |\mathcal{A}^*|}} \right| \\
&\leq \sum_k w_k \frac{|\sqrt{|\mathcal{A}^*|} - \sqrt{|\mathcal{A}^k|}| \cdot ||\mathcal{A}^0| + |\mathcal{A}^k| - |\mathcal{A}^0 \nabla \mathcal{A}^k||}{2\sqrt{|\mathcal{A}^*| \cdot |\mathcal{A}^0| \cdot |\mathcal{A}^k|}} \\
&+ \sum_k w_k \frac{\sqrt{|\mathcal{A}^k|} \cdot ||\mathcal{A}^k| - |\mathcal{A}^*| + |\mathcal{A}^0 \nabla \mathcal{A}^*| - |\mathcal{A}^0 \nabla \mathcal{A}^k||}{2\sqrt{|\mathcal{A}^*| \cdot |\mathcal{A}^0| \cdot |\mathcal{A}^k|}} \\
&\leq \underbrace{\sum_k w_k \frac{|\sqrt{|\mathcal{A}^*|} - \sqrt{|\mathcal{A}^k|}| \cdot ||\mathcal{A}^0| + |\mathcal{A}^k| - |\mathcal{A}^0 \nabla \mathcal{A}^k||}{2\sqrt{|\mathcal{A}^*| \cdot |\mathcal{A}^0| \cdot |\mathcal{A}^k|}}}_{A} \\
&+ \underbrace{\sum_k w_k \frac{||\mathcal{A}^k| - |\mathcal{A}^*||}{2\sqrt{|\mathcal{A}^*| \cdot |\mathcal{A}^0|}}}_{B} + \underbrace{\sum_k w_k \frac{||\mathcal{A}^0 \nabla \mathcal{A}^*| - |\mathcal{A}^0 \nabla \mathcal{A}^k||}{2\sqrt{|\mathcal{A}^*| \cdot |\mathcal{A}^0|}}}_{C}.
\end{aligned}
$$

For notational convenience, we divide the right-most-hand side of the above inequality into three parts and denote them by A, B, and C respectively. For part A, because $|\mathcal{A}^0| + |\mathcal{A}^k| - |\mathcal{A}^0 \nabla \mathcal{A}^k| = 2|\mathcal{A}^0 \cap \mathcal{A}^k|$ and $\big||\mathcal{A}^*| - |\mathcal{A}^k|\big| \leq |\mathcal{A}^* \nabla \mathcal{A}^k|$, together with $|\mathcal{A}^0 \cap \mathcal{A}^k| \leq \sqrt{|\mathcal{A}^0| \cdot |\mathcal{A}^k|}$,

we have

$$A = \sum_k w_k \frac{\big||\mathcal{A}^*| - |\mathcal{A}^k|\big| \cdot |\mathcal{A}^0 \cap \mathcal{A}^k|}{\big(\sqrt{|\mathcal{A}^*|} + \sqrt{|\mathcal{A}^k|}\big)\sqrt{|\mathcal{A}^*| \cdot |\mathcal{A}^0| \cdot |\mathcal{A}^k|}} \leq \sum_k w_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|}.$$

For part B, since $\big||\mathcal{A}^k| - |\mathcal{A}^*|\big| \leq |\mathcal{A}^k \nabla \mathcal{A}^*|$ and $\frac{|\mathcal{A}^0|}{|\mathcal{A}^*|} > \epsilon_2 > 0$ on $\mathcal{N}_{n_{j_l}}$, we have

$$B = \sum_k w_k \frac{\big||\mathcal{A}^k| - |\mathcal{A}^*|\big|}{2\sqrt{|\mathcal{A}^*| \cdot |\mathcal{A}^0|}} \leq \frac{1}{2\sqrt{\epsilon_2}} \sum_k w_k \frac{|\mathcal{A}^k \nabla \mathcal{A}^*|}{|\mathcal{A}^*|}.$$

For part C, it follows from the facts that $\big||\mathcal{A}^0 \nabla \mathcal{A}^*| - |\mathcal{A}^0 \nabla \mathcal{A}^k|\big| \leq |\mathcal{A}^* \nabla \mathcal{A}^k|$ and that $\frac{|\mathcal{A}^0|}{|\mathcal{A}^*|} > \epsilon_2 > 0$ on $\mathcal{N}_{n_{j_l}}$, we have

$$C = \sum_k w_k \frac{\big||\mathcal{A}^0 \nabla \mathcal{A}^*| - |\mathcal{A}^0 \nabla \mathcal{A}^k|\big|}{2\sqrt{|\mathcal{A}^*| \cdot |\mathcal{A}^0|}} \leq \frac{1}{2\sqrt{\epsilon_2}} \frac{\sum_k w_k |\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|}.$$

Consequently, we have that on $\mathcal{N}_{n_{j_l}}$,

$$|\widehat{G}(\mathcal{A}^0) - G(\mathcal{A}^0)| \leq A + B + C \leq (1 + \frac{1}{\sqrt{\epsilon_2}}) \sum_k w_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|}.$$

Under the assumption that the model weighting w is weakly consistent,

$$\sum_k w_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|} \xrightarrow{p} 0,$$

we must have $|\widehat{G}(\mathcal{A}^0) - G(\mathcal{A}^0)| \xrightarrow{p} 0$ on $\mathcal{N}_{n_{j_l}}$. This contradicts with the statement that $|\widehat{G}(\mathcal{A}^0) - G(\mathcal{A}^0)| > \epsilon_1 > 0$ on $\mathcal{S}_{n_j}$. Therefore, we have proved that $\frac{|\mathcal{A}^0|}{|\mathcal{A}^*|} \xrightarrow{p} 0$ on $\mathcal{S}_{n_j}$ under the beginning supposition.

Next, we prove actually we must have $|\widehat{G}(\mathcal{A}^0) - G(\mathcal{A}^0)| \xrightarrow{p} 0$ on $\mathcal{S}_{n_j}$ as $n_j \to \infty$. Because $\frac{|\mathcal{A}^0|}{|\mathcal{A}^*|} \xrightarrow{p} 0$ on $\mathcal{S}_{n_j}$, we can set $\delta_n = \sqrt{\frac{|\mathcal{A}^0|}{|\mathcal{A}^*|}}$, then $\delta_n \xrightarrow{p} 0$ and $\frac{|\mathcal{A}^0|}{|\mathcal{A}^*| \cdot \delta_n} = \delta_n \xrightarrow{p} 0$. Then

$$|G(\mathcal{A}^0)| = \frac{\big||\mathcal{A}^0| + |\mathcal{A}^*| - |\mathcal{A}^0 \nabla \mathcal{A}^*|\big|}{2\sqrt{|\mathcal{A}^*| \cdot |\mathcal{A}^0|}} = \frac{|\mathcal{A}^0 \cap \mathcal{A}^*|}{\sqrt{|\mathcal{A}^0| \cdot |\mathcal{A}^*|}} \leq \sqrt{\frac{|\mathcal{A}^0|}{|\mathcal{A}^*|}} \xrightarrow{p} 0,$$

that is, $G(\mathcal{A}^0) \xrightarrow{p} 0$. Now we prove that we also have $\widehat{G}(\mathcal{A}^0) \xrightarrow{p} 0$ as follows. Observe on $\mathcal{S}_{n_j}$

$$\begin{aligned}\widehat{G}(\mathcal{A}^0) &= \sum_k I(|\mathcal{A}^k| \le |\mathcal{A}^*| \cdot \delta_n) \cdot w_k \frac{|\mathcal{A}^0 \cap \mathcal{A}^k|}{\sqrt{|\mathcal{A}^0| \cdot |\mathcal{A}^k|}} + \sum_k I(|\mathcal{A}^k| > |\mathcal{A}^*| \cdot \delta_n) \cdot w_k \frac{|\mathcal{A}^0 \cap \mathcal{A}^k|}{\sqrt{|\mathcal{A}^0| \cdot |\mathcal{A}^k|}} \\ &\le \sum_k I(|\mathcal{A}^k| \le |\mathcal{A}^*| \cdot \delta_n) \cdot w_k + \sum_k I(|\mathcal{A}^k| > |\mathcal{A}^*| \cdot \delta_n) \cdot w_k \frac{|\mathcal{A}^0 \cap \mathcal{A}^k|}{\sqrt{|\mathcal{A}^0| \cdot |\mathcal{A}^k|}}.\end{aligned}$$

Then because $\sum_k w_k \frac{|\mathcal{A}^k \nabla \mathcal{A}^*|}{|\mathcal{A}^*|} \xrightarrow{p} 0$ and

$$\begin{aligned}&\sum_k w_k \frac{|\mathcal{A}^k \nabla \mathcal{A}^*|}{|\mathcal{A}^*|} \ge \sum_k w_k \frac{||\mathcal{A}^*| - |\mathcal{A}^k||}{|\mathcal{A}^*|} \\ &\ge \sum_k w_k \frac{||\mathcal{A}^*| - |\mathcal{A}^k||}{|\mathcal{A}^*|} \cdot I(|\mathcal{A}^k| \le |\mathcal{A}^*| \cdot \delta_n) \\ &\ge \frac{1}{2} \sum_k w_k \cdot I(|\mathcal{A}^k| \le |\mathcal{A}^*| \cdot \delta_n),\end{aligned}$$

we know $\sum_k I(|\mathcal{A}^k| \le |\mathcal{A}^*| \cdot \delta_n) \cdot w_k \xrightarrow{p} 0$. On $\mathcal{S}_{n_j}$, we also have

$$\begin{aligned}&\sum_k I(|\mathcal{A}^k| > |\mathcal{A}^*| \cdot \delta_n) \cdot w_k \frac{|\mathcal{A}^0 \cap \mathcal{A}^k|}{\sqrt{|\mathcal{A}^0| \cdot |\mathcal{A}^k|}} \\ &\le \sum_k I(|\mathcal{A}^k| > |\mathcal{A}^*| \cdot \delta_n) \cdot w_k \sqrt{\frac{|\mathcal{A}^0|}{|\mathcal{A}^k|}} \\ &\le \sum_k I(|\mathcal{A}^k| > |\mathcal{A}^*| \cdot \delta_n) \cdot w_k \sqrt{\frac{|\mathcal{A}^0|}{|\mathcal{A}^*| \cdot \delta_n}} \\ &\xrightarrow{p} 0,\end{aligned}$$

since $\frac{|\mathcal{A}^0|}{|\mathcal{A}^*| \cdot \delta_n} \xrightarrow{p} 0$ on $\mathcal{S}_{n_j}$. Therefore, we have shown $\widehat{G}(\mathcal{A}^0) \xrightarrow{p} 0$ on $\mathcal{S}_{n_j}$.

Now since we have proved that on $\mathcal{S}_{n_j}$, $G(\mathcal{A}^0) \xrightarrow{p} 0$ and $\widehat{G}(\mathcal{A}^0) \xrightarrow{p} 0$, so $|\widehat{G}(\mathcal{A}^0) - G(\mathcal{A}^0)| \xrightarrow{p} 0$ on $\mathcal{S}_{n_j}$, which contradicts with the beginning supposition that $|\widehat{G}(\mathcal{A}^0) -$

$G(\mathcal{A}^0)| > \epsilon_1 > 0$ on $\mathcal{S}_{n_j}$. Therefore the supposition does not hold, and we have proved the $|\widehat{G}(\mathcal{A}^0) - G(\mathcal{A}^0)|$ does converge to 0 in probability uniformly over $\mathbb{C}$. □

2.8.2 Proof of Theorem 2.2

Part I: standard deviation of F-measure

Proof For any $\mathcal{A}^0$ in $\mathbb{C}$, by definition of the standard deviation of F-measure, we have

$$\begin{aligned}
\mathrm{sd}\big(\widehat{F}(\mathcal{A}^0)\big) &\equiv \sqrt{\sum_k w_k \big(F(\mathcal{A}^0;\mathcal{A}^k) - \widehat{F}(\mathcal{A}^0)\big)^2} \\
&\le \sqrt{\sum_k w_k |F(\mathcal{A}^0;\mathcal{A}^k) - \widehat{F}(\mathcal{A}^0)|} \\
&\le \sqrt{\sum_k w_k |F(\mathcal{A}^0;\mathcal{A}^k) - F(\mathcal{A}^0)| + |F(\mathcal{A}^0) - \widehat{F}(\mathcal{A}^0)|}.
\end{aligned}$$

Using the facts proved in the proof for Theorem 2.1,

$$|\widehat{F}(\mathcal{A}^0) - F(\mathcal{A}^0)| \le \sum_k w_k |F(\mathcal{A}^0;\mathcal{A}^k) - F(\mathcal{A}^0)| \le 3 \sum_k w_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|},$$

we know

$$\mathrm{sd}(\widehat{F}(\mathcal{A}^0)) \le \sqrt{6 \sum_k w_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|}},$$

and

$$\sup_{\mathcal{A}^0 \in \mathbb{C}} \mathrm{sd}(\widehat{F}(\mathcal{A}^0)) \le \sqrt{6 \sum_k w_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|}} \xrightarrow{p} 0$$

under the assumption that the model weighting w is weakly consistent. □

Part II: standard deviation of G-measure

Proof For any $\mathcal{A}^0$ in $\mathbb{C}$, by definition of the standard deviation of G-measure, we have

$$\begin{aligned}\mathrm{sd}\big(\widehat{G}(\mathcal{A}^0)\big) &\equiv \sqrt{\sum_k w_k\big(G(\mathcal{A}^0;\mathcal{A}^k)-\widehat{G}(\mathcal{A}^0)\big)^2}\\ &\le \sqrt{\sum_k w_k|G(\mathcal{A}^0;\mathcal{A}^k)-\widehat{G}(\mathcal{A}^0)|}\\ &\le \sqrt{\sum_k w_k|G(\mathcal{A}^0;\mathcal{A}^k)-G(\mathcal{A}^0)|+|G(\mathcal{A}^0)-\widehat{G}(\mathcal{A}^0)|}.\end{aligned}$$

Using the facts in Theorem 2.1, we have

$$|\widehat{G}(\mathcal{A}^0)-G(\mathcal{A}^0)| \xrightarrow{p} 0.$$

So it suffices to show $\sum_k w_k|G(\mathcal{A}^0;\mathcal{A}^k)-G(\mathcal{A}^0)| \xrightarrow{p} 0$. The arguments are similar to those in the proof of Theorem 2.1. For completeness, the full proof is given below.

Suppose $\sum_k w_k|G(\mathcal{A}^0;\mathcal{A}^k)-G(\mathcal{A}^0)|$ does not converge to 0 in probability uniformly over $\mathbb{C}$, then there exist some subsequence $n_1, n_2, \cdots$, $\epsilon_1>0, \delta>0$, $\mathcal{A}^0_{n_j}\in\mathbb{C}$, and sets $\mathcal{S}_{n_j}$, s.t. $P(\mathcal{S}_{n_j})\ge\delta$ and $\sum_k w_k|G(\mathcal{A}^0_{n_j};\mathcal{A}^k)-G(\mathcal{A}^0_{n_j})|>\epsilon_1$ on $\mathcal{S}_{n_j}$. For ease of notation, we denote $\mathcal{A}^0_{n_j}$ as $\mathcal{A}^0$. We first prove that we must have $\frac{|\mathcal{A}^0|}{|\mathcal{A}^*|}\xrightarrow{p}0$ on $\mathcal{S}_{n_j}$ as $n_j\to\infty$. If not, then there exist $\epsilon_2>0$, a subsequence n_{j_l} and sets $\mathcal{N}_{n_{j_l}}$ such that on $\mathcal{N}_{n_{j_l}}$ we have $\frac{|\mathcal{A}^0|}{|\mathcal{A}^*|}>\epsilon_2>0$. Then we can actually prove $\sum_k w_k|G(\mathcal{A}^0;\mathcal{A}^k)-G(\mathcal{A}^0)|\xrightarrow{p}0$ on $\mathcal{N}_{n_{j_l}}$ as follows. On $\mathcal{N}_{n_{j_l}}$, since $\frac{|\mathcal{A}^0|}{|\mathcal{A}^*|}>\epsilon_2>0$, we have that

$$
\begin{aligned}
&\sum_k w_k |G(\mathcal{A}^0;\mathcal{A}^k) - G(\mathcal{A}^0)| \\
&\le \underbrace{\sum_k w_k \frac{|\sqrt{|\mathcal{A}^*|} - \sqrt{|\mathcal{A}^k|}| \cdot ||\mathcal{A}^0| + |\mathcal{A}^k| - |\mathcal{A}^0 \nabla \mathcal{A}^k||}{2\sqrt{|\mathcal{A}^*| \cdot |\mathcal{A}^0| \cdot |\mathcal{A}^k|}}}_{A} \\
&\quad + \underbrace{\sum_k w_k \frac{||\mathcal{A}^k| - |\mathcal{A}^*||}{2\sqrt{|\mathcal{A}^*| \cdot |\mathcal{A}^0|}}}_{B} + \underbrace{\sum_k w_k \frac{||\mathcal{A}^0 \nabla \mathcal{A}^*| - |\mathcal{A}^0 \nabla \mathcal{A}^k||}{2\sqrt{|\mathcal{A}^*| \cdot |\mathcal{A}^0|}}}_{C} \\
&\le (1 + \frac{1}{\sqrt{\epsilon_2}}) \sum_k w_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|}.
\end{aligned}
$$

Under the assumption that the model weighting w is weakly consistent,

$$\sum_k w_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|} \xrightarrow{p} 0,$$

we must have $\sum_k w_k |G(\mathcal{A}^0;\mathcal{A}^k) - G(\mathcal{A}^0)| \xrightarrow{p} 0$ on $\mathcal{N}_{n_{j_l}}$. This contradicts with the statement that $\sum_k w_k |G(\mathcal{A}^0;\mathcal{A}^k) - G(\mathcal{A}^0)| > \epsilon_1 > 0$ on $\mathcal{S}_{n_j}$. Therefore, we have proved that $\frac{|\mathcal{A}^0|}{|\mathcal{A}^*|} \xrightarrow{p} 0$ on $\mathcal{S}_{n_j}$ under the beginning supposition.

Next, we prove actually we must have $\sum_k w_k |G(\mathcal{A}^0;\mathcal{A}^k) - G(\mathcal{A}^0)| \xrightarrow{p} 0$ on $\mathcal{S}_{n_j}$ as $n_j \to \infty$. Similar to the proof in Theorem 2.1, we can prove that $G(\mathcal{A}^0) \xrightarrow{p} 0$ and $\widehat{G}(\mathcal{A}^0) \xrightarrow{p} 0$ on $\mathcal{S}_{n_j}$. We then have

$$\sum_k w_k |G(\mathcal{A}^0;\mathcal{A}^k) - G(\mathcal{A}^0)| \le \sum_k w_k G(\mathcal{A}^0;\mathcal{A}^k) + G(\mathcal{A}^0) = \widehat{G}(\mathcal{A}^0) + G(\mathcal{A}^0) \xrightarrow{p} 0$$

on $\mathcal{S}_{n_j}$, which contradicts with the beginning supposition that $\sum_k w_k |G(\mathcal{A}^0_{n_j};\mathcal{A}^k) - G(\mathcal{A}^0_{n_j})| > \epsilon_1 > 0$ on $\mathcal{S}_{n_j}$. Therefore, the supposition does not hold, and we have proved that $\sum_k w_k |G(\mathcal{A}^0_{n_j};\mathcal{A}^k) - G(\mathcal{A}^0_{n_j})|$ does converge to 0 in probability uniformly over $\mathbb{C}$. Since

we have

$$\mathrm{sd}\big(\widehat{G}(\mathcal{A}^0)\big) \le \sqrt{\sum_k w_k |G(\mathcal{A}^0;\mathcal{A}^k) - G(\mathcal{A}^0)| + |G(\mathcal{A}^0) - \widehat{G}(\mathcal{A}^0)|} \xrightarrow{p} 0$$

for any $\mathcal{A}^0 \in \mathbb{C}$, we have proved

$$\sup_{\mathcal{A}^0 \in \mathbb{C}} |\mathrm{sd}\big(\widehat{G}(\mathcal{A}^0)\big)| \xrightarrow{p} 0 \qquad \text{as } n \to \infty.$$

□

2.8.3 Proof of Theorem 2.3

Proof When a model screening is used to obtain the reduced candidate model list $\mathbb{S}$, the weights of the models in $\mathbb{S}$ are renormalized as $\widetilde{w}_k = w_k / w_{\mathbb{S}}$, where $w_{\mathbb{S}} = \sum_{k\in\mathbb{S}} w_k$. We next show that this renormalized weighting, though random, is still weakly consistent (in spite of possibly missing the true model in $\mathbb{S}$). Indeed,

$$\sum_{k\in\mathbb{S}} \widetilde{w}_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|} = \left(\sum_{k\in\mathbb{S}} w_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|} \right) / w_{\mathbb{S}} \le \left(\sum_{k\in\mathbb{C}} w_k \frac{|\mathcal{A}^* \nabla \mathcal{A}^k|}{|\mathcal{A}^*|} \right) / w_{\mathbb{S}},$$

which clearly converges to zero in probability under the weak consistency of w and the weak inclusion property of $\mathbb{S}$. Then the arguments for the convergence of $\widehat{F}$ and $\widehat{G}$ in the proofs of Theorems 1 and 2 continue to work. Thus we know that the conclusions of Theorems 2.1 and 2.2 still hold. □

2.8.4 Remarks on Theorem 2.3

Theorem 2.3 relies on a good quality of the set of candidate models obtained from a model screening step. The weak inclusion property demands $\mathbb{S}$ to contain some (good) models with non-vanishing cumulated weight, but does not require $\mathcal{A}^*$ to be in $\mathbb{S}$ with

high-probability. If the true model is really strong, it is not very likely to be missed by $\$$. In contrast, if there are very weak true coefficients, the true model may not be included in $\$$. Fortunately, in this case, as long as the number of small effects is asymptotically negligible compared to the true model size, some models close to $\mathcal{A}^*$ are most likely to be included in $\$$, and the weak inclusion property may hold. For example, suppose the true model size is of order $\log n$ and there are no more than $(\log n)^{1/2}$ small coefficients. Then the models without some of the small-effect variables are likely to receive comparable or even higher weights than the true model. Then, even if the true model is missed in $\$$, the weak inclusion property holds.

In particular, if $\$$ is obtained as the solution path of a penalized method and has the weak inclusion property, the method is said to be *weakly path-inclusive* or *weakly path-consistent*. Note that for a consistent weighting, our definition here on $\$$ is weaker than the path-consistency that requires the true model to be included on the solution path with probability going to 1.

In the high-dimensional case, we can set $\$$ as a large collection of the models obtained from the solution paths of multiple penalized methods, such as (adaptive) Lasso, SCAD and MCP. Specifically, we can obtain the models $\$_{\text{Lasso}}$, $\$_{\text{SCAD}}$, $\$_{\text{MCP}}$ for (adaptive) Lasso, SCAD and MCP respectively on the solution paths $\{\widehat{\boldsymbol{\beta}}^{\lambda_1}, \ldots, \widehat{\boldsymbol{\beta}}^{\lambda_L}\}$ for decreasing sequences of tuning parameters $\{\lambda_1, \ldots, \lambda_L\}$. These models are then combined together as a union of candidate models $\$ = \{\$_{\text{Lasso}}, \$_{\text{SCAD}}, \$_{\text{MCP}}\}$. These penalized methods are good choices, since according to existing theories (Tibshirani, 1996; Zou, 2006a; Fan and Li, 2001; Zhang, 2010), $\$$ produced by the solution paths of these methods ensure path-consistency under certain regularity conditions. In fact, in order to get Theorem 2.3, only one of $\$_{\text{Lasso}}$, $\$_{\text{SCAD}}$ and $\$_{\text{MCP}}$ needs to be weakly path-consistent. Of course, users are not limited to these

options, they can add models obtained from any other weakly path-consistent variable selection methods into $\mathbb{S}$ to further enhance the chance of capturing the true/best model. More details about candidate models are discussed in Section 4.1 of the main paper.

2.8.5 Additional simulation results

Table 2.7: Classification case (Example 2.2).

	F	G	d_F	d_G
		Lasso		
True	0.631 (0.008)	0.680 (0.006)		
ARM	0.697 (0.007)	0.734 (0.006)	0.066 (0.002)	0.054 (0.002)
BIC-p	0.639 (0.008)	0.686 (0.006)	0.008 (0.001)	0.006 (0.001)
		AdLasso		
True	0.989 (0.004)	0.989 (0.004)		
ARM	0.929 (0.002)	0.935 (0.002)	0.067 (0.002)	0.062 (0.002)
BIC-p	0.987 (0.003)	0.988 (0.002)	0.009 (0.001)	0.008 (0.001)
		MCP		
True	0.964 (0.008)	0.967 (0.008)		
ARM	0.922 (0.004)	0.929 (0.004)	0.065 (0.002)	0.059 (0.002)
BIC-p	0.965 (0.008)	0.968 (0.007)	0.009 (0.001)	0.008 (0.001)
		SCAD		
True	0.955 (0.010)	0.960 (0.009)		
ARM	0.919 (0.005)	0.926 (0.004)	0.065 (0.002)	0.059 (0.002)
BIC-p	0.956 (0.009)	0.961 (0.008)	0.009 (0.001)	0.008 (0.001)

Table 2.8: Classification case (Example 2.3).

	F	G	d_F	d_G
		Lasso		
True	0.154 (0.011)	0.278 (0.010)		
ARM	0.129 (0.009)	0.251 (0.009)	0.025 (0.002)	0.028 (0.002)
BIC-p	0.159 (0.011)	0.283 (0.010)	0.010 (0.002)	0.010 (0.002)
		AdLasso		
True	0.712 (0.021)	0.751 (0.018)		
ARM	0.627 (0.020)	0.682 (0.016)	0.091 (0.006)	0.076 (0.005)
BIC-p	0.716 (0.021)	0.754 (0.017)	0.030 (0.006)	0.026 (0.005)
		MCP		
True	0.498 (0.015)	0.576 (0.012)		
ARM	0.433 (0.015)	0.523 (0.012)	0.067 (0.004)	0.056 (0.003)
BIC-p	0.511 (0.015)	0.586 (0.012)	0.026 (0.005)	0.020 (0.004)
		SCAD		
True	0.214 (0.006)	0.344 (0.005)		
ARM	0.183 (0.006)	0.312 (0.006)	0.032 (0.002)	0.033 (0.002)
BIC-p	0.225 (0.007)	0.352 (0.006)	0.017 (0.004)	0.014 (0.003)

Table 2.9: Classification case (Example 2.4).

	F	G	d_F	d_G
		Lasso		
True	0.720 (0.005)	0.734 (0.005)		
ARM	0.493 (0.006)	0.572 (0.004)	0.227 (0.007)	0.163 (0.006)
BIC-p	0.616 (0.006)	0.667 (0.004)	0.109 (0.005)	0.077 (0.005)
		AdLasso		
True	0.794 (0.005)	0.800 (0.005)		
ARM	0.722 (0.006)	0.755 (0.005)	0.081 (0.006)	0.059 (0.005)
BIC-p	0.876 (0.006)	0.883 (0.005)	0.096 (0.006)	0.094 (0.006)
		MCP		
True	0.751 (0.005)	0.770 (0.005)		
ARM	0.793 (0.004)	0.813 (0.004)	0.063 (0.005)	0.056 (0.004)
BIC-p	0.932 (0.005)	0.934 (0.005)	0.182 (0.006)	0.164 (0.005)
		SCAD		
True	0.778 (0.006)	0.789 (0.006)		
ARM	0.755 (0.005)	0.781 (0.004)	0.064 (0.006)	0.055 (0.005)
BIC-p	0.913 (0.006)	0.916 (0.005)	0.141 (0.007)	0.132 (0.006)

Table 2.10: Classification case (Example 2.5).

	F	G	d_F	d_G
		Lasso		
True	0.386 (0.006)	0.440 (0.005)		
ARM	0.223 (0.004)	0.348 (0.004)	0.163 (0.006)	0.093 (0.005)
BIC-p	0.359 (0.006)	0.465 (0.005)	0.039 (0.004)	0.043 (0.003)
		AdLasso		
True	0.726 (0.005)	0.735 (0.005)		
ARM	0.616 (0.008)	0.669 (0.006)	0.118 (0.007)	0.079 (0.005)
BIC-p	0.859 (0.008)	0.865 (0.008)	0.137 (0.007)	0.133 (0.006)
		MCP		
True	0.683 (0.008)	0.695 (0.008)		
ARM	0.639 (0.009)	0.687 (0.007)	0.079 (0.006)	0.063 (0.005)
BIC-p	0.868 (0.008)	0.871 (0.008)	0.186 (0.006)	0.177 (0.006)
		SCAD		
True	0.634 (0.008)	0.637 (0.008)		
ARM	0.506 (0.010)	0.580 (0.008)	0.131 (0.007)	0.072 (0.005)
BIC-p	0.743 (0.009)	0.766 (0.008)	0.110 (0.006)	0.130 (0.006)

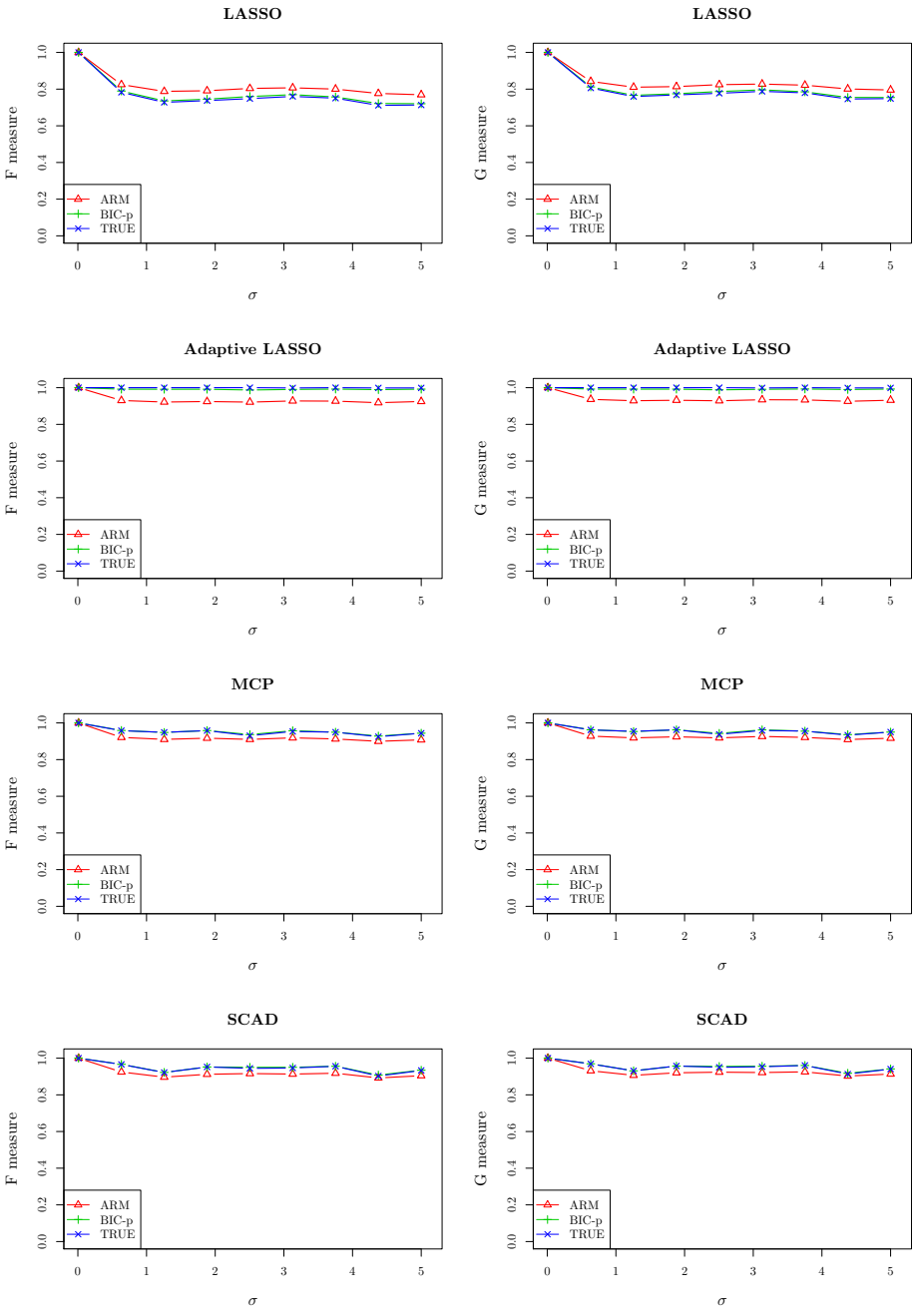

Figure 2.2: Regression case (Example 2.2).

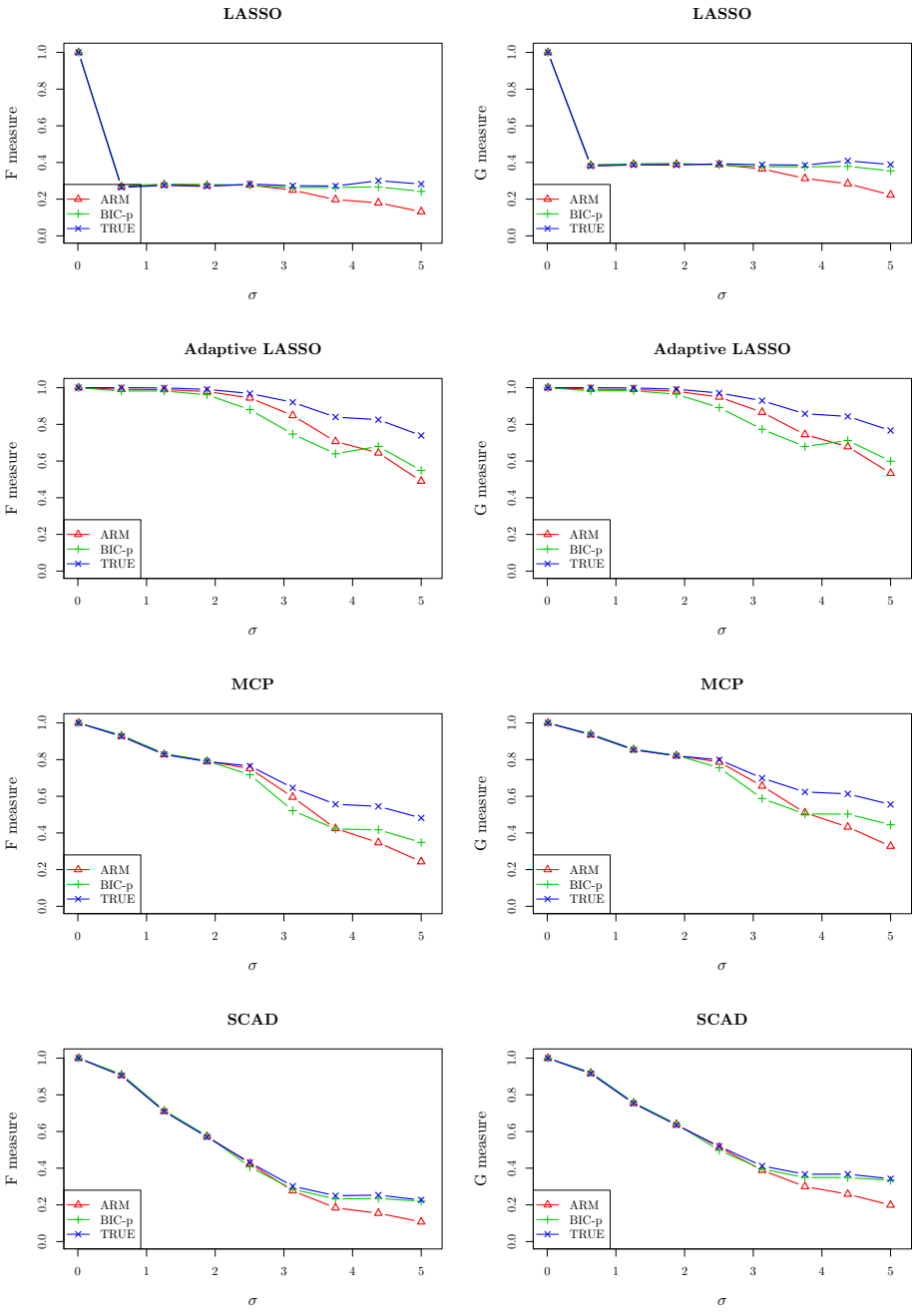

Figure 2.3: Regression case (Example 2.3)

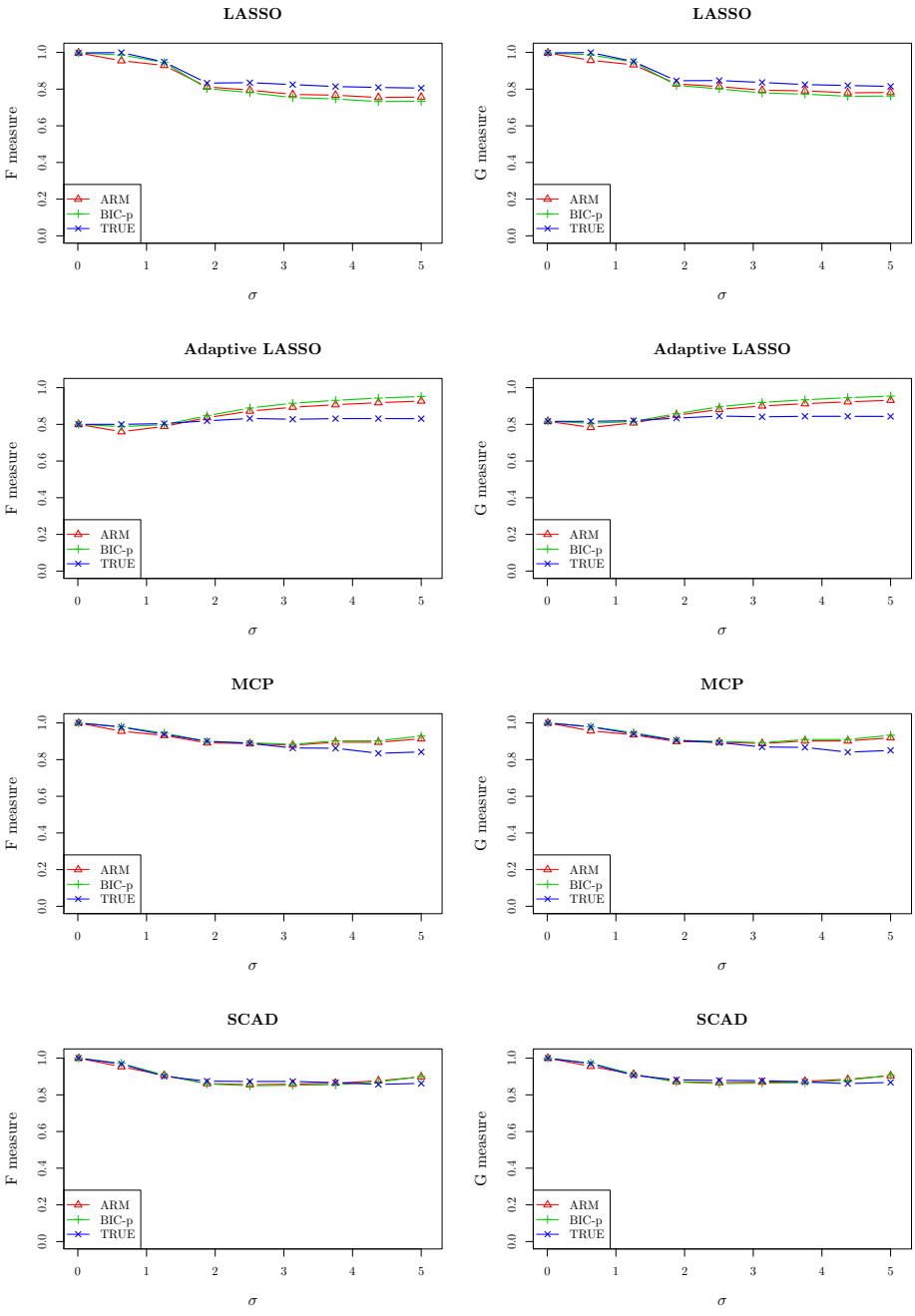

Figure 2.4: Regression case (Example 2.4).

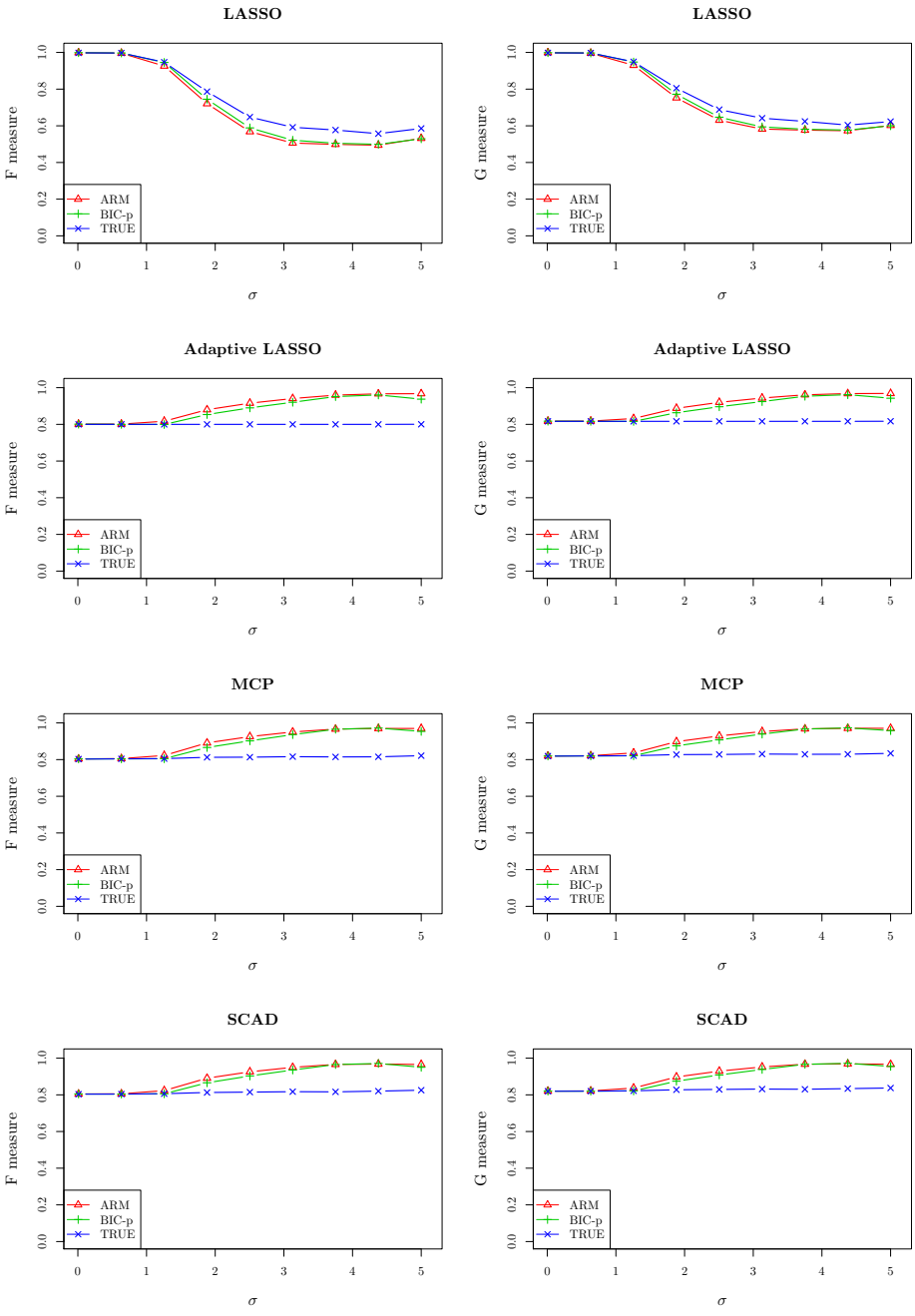

Figure 2.5: Regression case (Example 2.5).

2.8.6 Sensitivity analysis of ψ

In this simulation, we study how the choices of the prior weight parameter ψ impact the estimation performance of PAVI. We only present results for the regression case, since we found that the classification case gives similar results. We adopt the simulation setting of Example 3 defined in Section 5.1, except that we let $\sigma^2 = 1$, $n = 100$ and we vary $p = \{200, 2000\}$. We compare $\widehat{F}(\mathcal{A}^0)$ and $\widehat{G}(\mathcal{A}^0)$ with the true $F(\mathcal{A}^0)$ and $G(\mathcal{A}^0)$ under nine different values of ψ, that is, $\psi \in \{0, 0.5, 1, 2, 4, 6, 8, 10, 20\}$.

All simulation cases are repeated for 100 times and the corresponding values are computed and averaged. The results are shown in Figure 2.6 for $p = 200$ case and 2.7 for $p = 2000$ case. We can see that by using either the ARM or BIC-p weighting with $\psi = 1$ or 2, the estimated $\widehat{F}(\mathcal{A}^0)$ and $\widehat{G}(\mathcal{A}^0)$ can better reflect the true $F(\mathcal{A}^0)$ and $G(\mathcal{A}^0)$ for all four different variable selection methods under evaluation. We observed similar results in other simulation settings. We conclude that overall, under $\psi = 1$ or 2 setting, PAVI is stably reliable in our simulation, while either a too large or too small value of ψ leads to poor estimation performance.

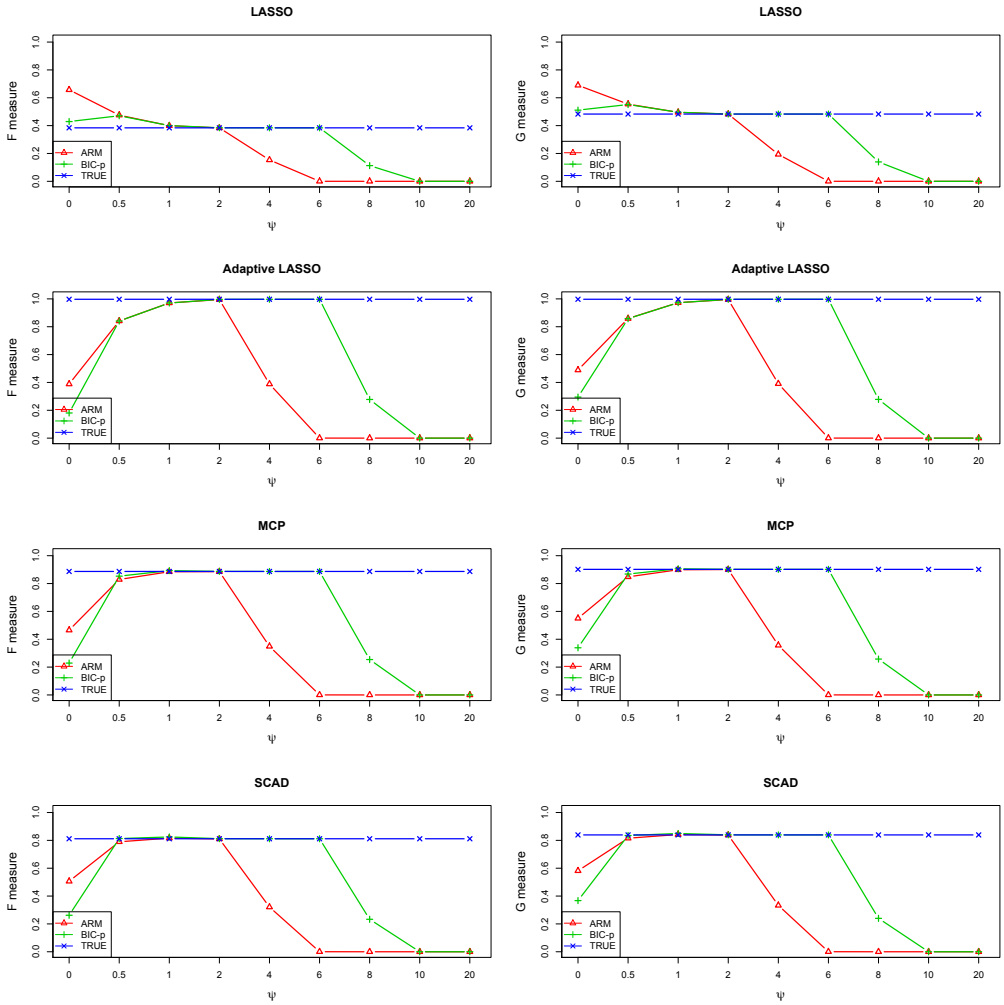

Figure 2.6: Sensitivity analysis of ψ. Regression case, $n = 100$ and $p = 200$.

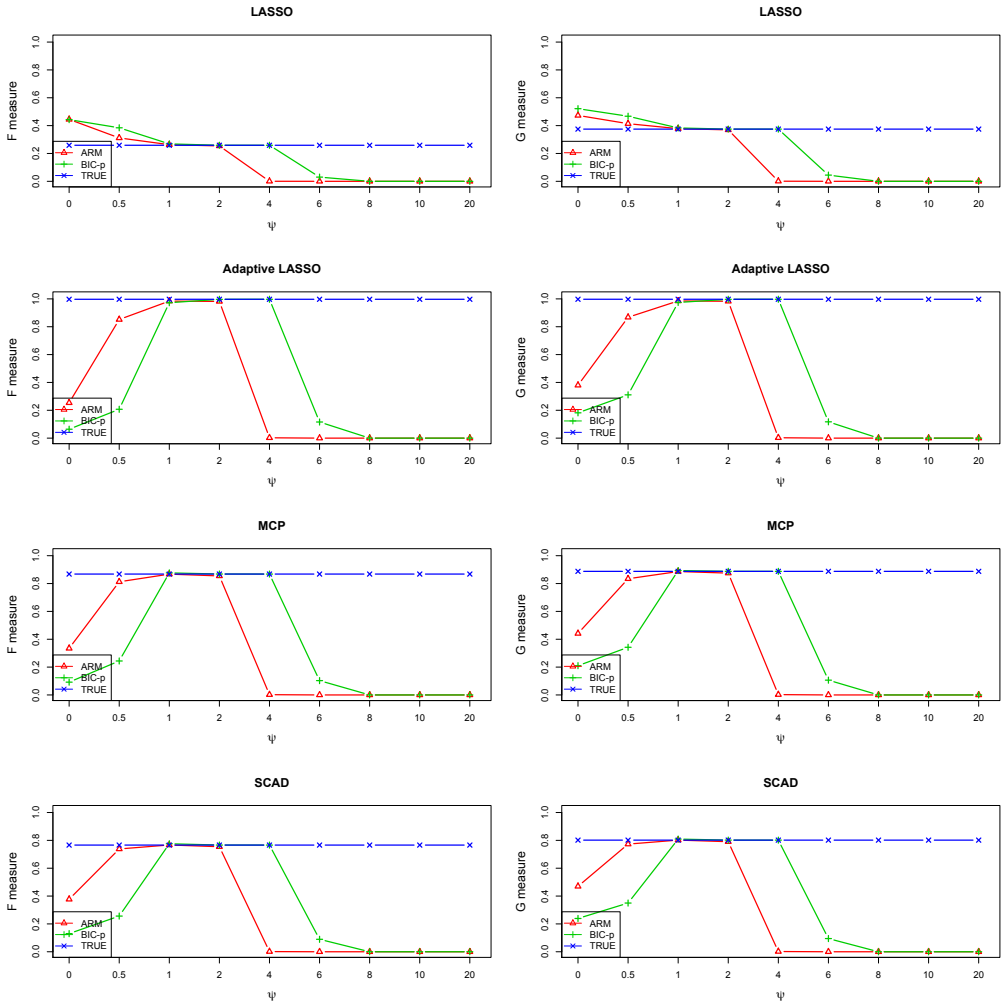

Figure 2.7: Sensitivity analysis of ψ. Regression case, $n = 100$ and $p = 2000$.

2.8.7 Impact of candidate models

In this simulation study, we investigate how the quality of the candidate models impacts the estimation performance of PAVI:

- How heterogeneity of the candidate model $\mathbb{S}$ affects the estimation performance.
- How it affects estimation performance when $\mathbb{S}$ contains/not contain the true model.

We only present the results from the regression case. The data are generated using the setting described in Example 3 of Section 5.1, under eight different noise levels σ ranging from 0.01 to 4. We set $n = 50$ and $p = 100$. The true model is represented by the vector $\mathcal{A}^* = (1, 1, 1, 0, 0, 0, \ldots, 0)$ with $|\mathcal{A}^*| = 3$, i.e. only the first three variables are nonzero, the remaining 97 are noise variables. Suppose that a given MCP model $\mathcal{A}^0$ is evaluated by using the estimated F-measure $\widehat{F}(\mathcal{A}^0)$ obtained from the BIC-p (the modified BIC) weighting with prior adjustment $\psi = 1$. The sets of candidate models used in estimation of $\widehat{F}(\mathcal{A}^0)$ are generated under the following two settings:

Setting I ($\mathcal{A}^*$ is not included in $\mathbb{S}$.) We use a union of 100 models as the set of candidate models $\mathbb{S} = \{\mathcal{A}^k\}_{k=1}^{100}$. Each $\mathcal{A}^k$ is a contaminated version of the true model $\mathcal{A}^*$ with a pre-specified contamination level $r \in (0, 1)$. Specifically, each $\mathcal{A}^k$ is generated in the following way: we take $\mathcal{A}^*$, randomly select $100r\%$ of its elements and flip their values, i.e. switch to 1 if the original value is 0, and to 0 if the original value is 1. Thus r controls heterogeneity of $\mathbb{S}$: the smaller r becomes, the closer the candidate model gets to the true model.

Setting II ($\mathcal{A}^*$ is included in $\mathbb{S}$.) The set of candidate models $\mathbb{S} = \{\mathcal{A}^k\}_{k=1}^{100}$ is also generated using Setting I, except that one of $\mathcal{A}^k$'s is replaced by $\mathcal{A}^*$.

We compare estimation performances of $\widehat{F}(\mathcal{A}^0)$ under Setting I and II with varying contamination levels $r = \{0.01, 0.03, 0.05, 0.1, 0.2\}$. All simulation cases are repeated for 100 times and the corresponding values are computed and averaged. The results are shown in Figure 2.8: (1) The left panel shows the results under Setting I. We find that less heterogeneity in $ leads to better estimation performance of $\widehat{F}(\mathcal{A}^0)$ when $\mathcal{A}^* \notin$ $. This indicates that, if the true model is not included in the candidate models, it leads to better performance when $ has most of its models being close to the true model; (2) However, from the results under Setting II shown in the right panel, we can see that if the true model is included in $, then heterogeneity of $ becomes not much influential on the estimation performance.

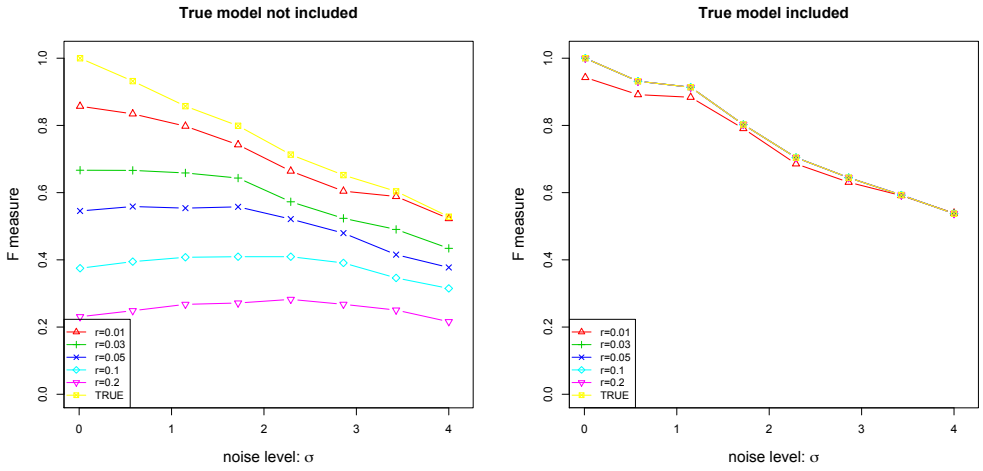

Figure 2.8: Impact of candidate models on estimation performance of F-measures in the regression case, $n = 50$ and $p = 100$, under Setting I where $\mathcal{A}^*$ is not included in $ (left panel), and Setting II where $\mathcal{A}^*$ is included in $ (right panel) with varying contamination levels $r = \{0.01, 0.03, 0.05, 0.1, 0.2\}$.

2.8.8 Additional real data examples

Table 2.11: Estimated F- and G-measures and standard deviations for `Prostate`. L10 has numerically zero $\widehat{F}$ and $\widehat{G}$ values (shown in bold).

	ARM				BIC-p			
	F	*sd.F*	*G*	*sd.G*	*F*	*sd.F*	*G*	*sd.G*
Lasso	0.064	0.004	0.181	0.005	0.064	0.003	0.181	0.004
AdLasso	0.190	0.011	0.323	0.009	0.189	0.008	0.323	0.007
MCP	0.018	0.019	0.027	0.022	0.018	0.012	0.027	0.014
SCAD	0.097	0.006	0.225	0.007	0.096	0.005	0.225	0.005
ImpS	0.333	0.011	0.447	0.008	0.333	0.012	0.447	0.009
S12	0.395	0.037	0.494	0.047	0.400	0.003	0.500	0.007
L10	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**

Table 2.12: Labels of selected genes for `Colon`.

	Labels of selected genes
Lasso	{66, 249, 377, 493, 765, 1325, 1346, 1423, 1582, 1644, 1772, 1870}
AdLasso	{249, 377, 765, 1582, 1772, 1870}
MCP	{249, 377, 1644, 1772, 1870}
SCAD	{377, 617, 765, 1024, 1325, 1346, 1482, 1504, 1582, 1644, 1772, 1870}
ImpS	{249, 1772}
L11	{249, 286, 765, 1058, 1485, 1671, 1771, 1836}
Y10	{14, 161, 249, 377, 492, 493, 576, 792, 822, 1042, 1210, 1346, 1400, 1423, 1549, 1635, 1772, 1843, 1924}
C11	{249, 399, 513, 515, 780, 1042, 1325, 1582, 1771, 1772}
L10	{732, 994, 1473, 1763, 1794, 1843}

Table 2.13: Labels of selected genes for `Leukemia`.

	Labels of selected genes
Lasso	{804, 1239, 1674, 1745, 1779, 1796, 1834, 1882, 1928, 1933, 1941, 2121, 2288, 3847, 4196, 4328, 4847, 4951, 4973, 5002, 5107, 5335, 5766, 6055, 6169, 6539, 6855}
AdLasso	{1779, 1834, 4328, 4847, 4951}
MCP	{804, 1941, 3837, 4714, 4847, 4951, 6539}
SCAD	{804, 1674, 1745, 1779, 1834, 1882, 1928, 1941, 2288, 3847, 4196, 4328, 4847, 4951, 4973, 5002, 5766, 5772, 6169, 6225, 6281, 6539, 6855}
ImpS	{1239, 4847, 4951}
J11[1]	{1376, 1394, 1674, 1882, 2186, 2402, 6200, 6201, 6803}
J11[2]	{1394, 1674, 1882, 2186, 5976, 6200, 6201, 6806}
Y10	{760, 804, 1745, 1829, 1834, 1882, 2354, 3320, 4052, 4211, 4377, 4535, 4847, 5039, 6041, 6218, 6376, 6540}
L10	{220, 1086, 1834, 2020}

Table 2.14: Labels of selected genes for `Prostate`.

	Labels of selected genes
Lasso	{1107, 3617, 4282, 4438, 4525, 4636, 5661, 5838, 5890, 6145, 6185, 6838, 7375, 7428, 7539, 7623, 7915, 8123, 8965, 9034, 9093, 9816, 9850, 10234, 10537, 10956, 11858, 11871, 12153, 12462}
AdLasso	{5661, 5890, 6185, 7539, 7623, 8965, 9034, 9093, 10234, 11858}
MCP	{7623, 7924, 8965, 9034, 9816, 10234, 11858}
SCAD	{1107, 3540, 4636, 5661, 5838, 5890, 6185, 7623, 8603, 8965, 9034, 9093, 9816, 10234, 10956, 11858, 11871, 12153}
ImpS	{8965, 9034, 10234, 11858}
S12	{4377, 6185, 6390, 6915}
L10	{4743, 6096, 8475, 9575, 9927, 12331}

2.8.9 An extented study of the real data examples

In this section, we provide a more objective comparison of the aforementioned models to be evaluated in the three real data examples. Specifically, we include all of the models to be evaluated in the set of candidate models as well. The rationale is that those models identified in Sections 2.6.3 and 2.8.8 as highly inaccurate in terms of variable selection performance might be unfairly treated due to our choice of the candidate models from the solution paths of lasso, MCP and SCAD alone, which might be quite far away from those models to be evaluated. By enlarging the candidate model set to also include those models to be evaluated, we hope to minimize the bias in the choice of the candidate models and to assign certain weights to those models to be evaluated. The results for the extended analyses of the three real data sets, `Colon`, `Leukemia`, and `Prostate`, are summarized in Tables 2.15 – 2.17, from which we can see that those highly inaccurate models still have very low F- and G-measures. One possible explanation is that though those models have now been included in the set of candidate models, they receive extremely low weights (close to zero) during the weight assignment process. This is another evidence that the selected variables in those models are indeed not supported by many other methods.

Table 2.15: Estimated F- and G-measures and standard deviations for `Colon` with extended set of candidate models. L10 has numerically zero $\widehat{F}$ and $\widehat{G}$ values (shown in bold).

	ARM-p				BIC-p			
	F	sd(F)	G	sd(G)	F	sd(F)	G	sd(G)
Lasso	0.033	0.062	0.062	0.116	0.199	0.067	0.326	0.072
AdLasso	0.019	0.054	0.029	0.084	0.114	0.093	0.191	0.148
MCP	0.143	0.290	0.149	0.302	0.737	0.233	0.759	0.222
SCAD	0.032	0.062	0.058	0.113	0.195	0.070	0.314	0.082
ImpS	0.146	0.276	0.154	0.291	0.787	0.198	0.809	0.186
L11	0.031	0.076	0.050	0.122	0.130	0.076	0.228	0.117
Y10	0.022	0.043	0.050	0.095	0.138	0.048	0.267	0.059
C11	0.040	0.077	0.069	0.130	0.239	0.082	0.362	0.083
L10	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**

Table 2.16: Estimated F- and G-measures and standard deviations for `Leukemia` with extended set of candidate models. J11^1 and J11^2 have numerically zero $\widehat{F}$ and $\widehat{G}$ values (shown in bold).

	ARM-p				BIC-p			
	F	sd(F)	G	sd(G)	F	sd(F)	G	sd(G)
Lasso	0.075	0.005	0.197	0.011	0.075	0.002	0.198	0.003
AdLasso	0.137	0.090	0.228	0.150	0.191	0.042	0.317	0.069
MCP	0.324	0.467	0.324	0.467	0.049	0.216	0.049	0.216
SCAD	0.089	0.005	0.216	0.012	0.090	0.003	0.217	0.003
ImpS	0.665	0.037	0.705	0.040	0.667	0.010	0.707	0.009
J11^1	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
J11^2	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
Y10	0.105	0.006	0.235	0.013	0.105	0.003	0.236	0.003
L10	0.269	0.187	0.336	0.234	0.380	0.086	0.475	0.108

Table 2.17: Estimated F- and G-measures and standard deviations for `Prostate` with extended set of candidate models. L10 has numerically zero $\widehat{F}$ and $\widehat{G}$ values (shown in bold).

	ARM-p				BIC-p			
	F	sd(F)	G	sd(G)	F	sd(F)	G	sd(G)
Lasso	0.067	0.009	0.186	0.017	0.072	0.017	0.192	0.021
AdLasso	0.143	0.019	0.276	0.025	0.152	0.033	0.285	0.031
MCP	0.817	0.077	0.838	0.075	0.824	0.090	0.846	0.079
SCAD	0.094	0.011	0.221	0.019	0.097	0.019	0.224	0.022
ImpS	0.666	0.066	0.706	0.066	0.692	0.090	0.729	0.079
S12	0.395	0.036	0.494	0.042	0.388	0.041	0.488	0.040
L10	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**

Chapter 3

Variable Selection Diagnostics for Generalized Linear Models

3.1 Introduction

Variable selection methods are indispensable tools in many contemporary data analyses, for both feasibility and interpretability purposes. Many exciting results have been obtained on variable selection in the last two decades or so, particularly for high-dimensional data, via various regularization techniques that help achieve sparse representations of the rather complex nature of the data. See Ding et al. (2018), Wainwright (2019), Fan et al. (2020) and references therein for a comprehensive review of recent developments. An important aspect of the performance evaluation of a variable selection method is to reliably assess the accuracy of its identified sparsity pattern, which is difficult for real world data as the true sparsity pattern is often not known *apriori*. For example, in the high-dimensional least squares regression, different regularization methods have been proposed to encourage sparsity in the coefficient estimation, including, but not limited to, the lasso (Tibshirani, 1996), the adaptive lasso (AdaLasso, Zou, 2006b), the smoothly clipped absolute deviation (SCAD, Fan and Li, 2001) penalization and the minimax concave penalization (MCP, Zhang,

2010). It is often observed that for the same real world data, these methods could yield drastically different sets of selected variables, most probably due to the discrepancies in the underlying assumptions made by the different methods about the truth. This leaves practitioners a difficult decision to make with regard to which method is the best for the data at hand because they typically do not know exactly what assumptions best describe or approximate the true nature of the data.

To that end, Nan and Yang (2014) propose variable selection deviation (VSD) measures in the context of high-dimensional least squares regression, to evaluate the trustworthiness of a variable selection method, by assembling a pool of candidate models and aggregating them by an exponential weighting scheme based on the information criterion or the cross-assessed negative log-likelihood. This is particularly relevant for real-world data analysis where one does not know the underlying true model, but an importance score can be assigned to each candidate model according to their predictive performance, thus providing a way of uncovering or approximating the truth via proper model aggregation. The VSD measures provide a quantitative tool for describing the accuracy of the identified sparsity pattern from a variable selection method based on the votes from the pool of candidate models. A larger VSD value typically means the corresponding variable selection method is not favored by the majority of the models and may be less trustworthy than a method with a smaller VSD value.

It is noted in Yu et al. (2022) that the VSD measures are counts derived from the false positives and false negatives and may depend on the sizes of the true and selected models. For easier interpretation of the variable selection performance of a method, several other variable selection diagnostic measures such as the F- and G- measures are proposed by Yu et al. (2022). There are metrics that are bounded in $[0, 1]$ with larger values indicating better

variable selection performance. Both the least squares and logistic regression diagnostics are discussed by Yu et al. (2022) and in Chapter 2 of this dissertation in terms of their variable selection performance using these measures.

In this chapter, we extend the work of Nan and Yang (2014) and Yu et al. (2022) to a broader class of models in the exponential dispersion family (Jørgensen, 1987; McCullagh and Nelder, 1989; Jørgensen, 1997) , including, for example, the Poisson, gamma and compound Poisson-gamma models. In particular, we consider the Tweedie family of models that possesses a power mean-variance relationship, for its wide spectrum of applications in fields such as insurance (Smyth and Jørgensen, 2002), ecology (Dunn, 2004), political science (Lauderdale, 2012) and health and biomedical studies (Coates et al., 2018). We note that the variable selection diagnostic measures for the logistic regression, also an important member of the exponential dispersion models, has been considered in Nan (2015) and thoroughly studied in Yu et al. (2022) and Chapter 2 of this dissertation, and hence will not be further discussed in this chapter.

The remainder of this chapter is organized as follows. In Section 3.2, we introduce the Tweedie distribution, which is an important member of the exponential dispersion models, and the associated Tweedie generalized linear models that include the commonly used regression models such as the least squares, Poisson and gamma regressions. In Section 3.3, we give an introduction to the concept of the variable selection diagnostic measures, following the idea of Nan and Yang (2014) and Yu et al. (2022), and briefly discuss their applications to the high-dimensional least squares and logistic regressions. We discuss the variable selection diagnostic measures in the high-dimensional Tweedie models in Section 3.4 and conduct relevant numerical studies in Section 3.5. Concluding remarks are made in Section 3.6.

3.2 The Tweedie Models

The exponential dispersion family (EDF, Jørgensen, 1987, 1997) is the prototype distribution family for constructing the generalized linear models (McCullagh and Nelder, 1989). The family includes response distributions for the most commonly used generalized linear models such as the Gaussian, binomial, Poisson models and so on. We say that a random variable Y comes from the EDF if the probability density or mass function of Y is of the form

$$f(y;\theta,\phi) = c(y,\phi)\exp\left\{\frac{y\theta-\kappa(\theta)}{\phi}\right\}, \tag{3.1}$$

for some suitable cumulant function $\kappa(\cdot)$ and normalizing function $c(\cdot,\cdot)$. The canonical parameter θ resides in an open interval in $\mathbb{R}$ such that $\kappa(\theta) < \infty$ and the dispersion parameter ϕ is positive. By Bartlett identities (Bartlett, 1953; Cordeiro et al., 2014), we have $\mathbb{E}(Y) = \kappa'(\theta)$ and $\text{var}(Y) = \phi\kappa''(\theta)$. It is often assumed that $\kappa(\cdot)$ is a strictly convex function such that $\text{var}(Y) > 0$. Hence, $\kappa'(\cdot)$ is strictly monotone, which implies that $\theta = (\kappa')^{-1}(\mu)$ where $\mu = \mathbb{E}(Y)$. As a result, one has $\text{var}(Y) = \phi\kappa''((\kappa')^{-1}(\mu))$. The function $V(\cdot) = \kappa''((\kappa')^{-1}(\cdot))$ is often referred to as the variance function.

The Tweedie family of distributions obeys the power law mean-variance relationship $V(\mu) = \mu^{\zeta}$ for some power parameter $\zeta \in \mathbb{R}$. The Tweedie family was first considered by Tweedie (1984) and was thoroughly studied later by Jørgensen (1987) and Jørgensen (1997). It can be shown that the family is well defined for all $\zeta \notin (0,1)$ (Jørgensen, 1997) and many of the response distributions commonly used in generalized linear models belong to this family, including the normal ($\zeta = 0$), Poisson ($\zeta = 1$), gamma ($\zeta = 2$), and inverse Gaussian ($\zeta = 3$) distributions. According to Dunn and Smyth (2005), the case $\zeta < 0$ is seldom of

practical interest and the case $\zeta > 2$ is characterized by the stable distribution (Nolan, 2020). Of particular interest is the case $\zeta \in (1, 2)$, which can be represented as a Poisson mixtures of gamma distributions. The mixture distribution has mass at zero and is supported on the nonnegative reals. This interpretation gives rise to its wide applications in fields such as insurance, ecology and health science where, for example, the claim amount, precipitation, and alcohol consumption are frequently modeled but zero records are common.

In the sequel, we will focus particularly on the case $\zeta \in (1, 2)$ due to its common use in practice. Let $X_i \sim \text{Gamma}(\alpha, \gamma), i = 1, 2, \ldots$ be independent and identically distributed gamma random variables with shape parameter $\alpha > 0$ and scale parameter $\gamma > 0$. Assume they are all independent of the Poisson random variable $N \sim \text{Poisson}(\lambda)$, where λ is positive. Then, it can be shown that

$$Y = \sum_{i=1}^{N} X_i = X_1 + \cdots + X_N$$

follows a Tweedie distribution with power parameter $\zeta = (\alpha + 2)/(\alpha + 1) \in (1, 2)$. More specifically, in the form of (3.1), the distribution of Y satisfies

$$\theta = \frac{\mu^{1-\zeta}}{1-\zeta}, \quad \kappa(\theta) = \frac{\mu^{2-\zeta}}{2-\zeta} = \frac{[(1-\zeta)\theta]^{\frac{2-\zeta}{1-\zeta}}}{2-\zeta},$$

and

$$c(y, \phi) = \begin{cases} 1, & y = 0 \\ \frac{1}{y} \sum_{j=1}^{\infty} \frac{y^{j\frac{2-\zeta}{\zeta-1}}}{j!\phi^{\frac{j}{\zeta-1}}(2-\zeta)^j(\zeta-1)^{j\frac{2-\zeta}{\zeta-1}}\Gamma(j\frac{2-\zeta}{\zeta-1})}, & y > 0, \end{cases}$$

where

$$\mu = \lambda\alpha\gamma, \quad \zeta = \frac{\alpha+2}{\alpha+1}, \quad \text{and} \quad \phi = (\alpha+1)(\lambda\alpha)^{-\frac{1}{\alpha+1}}\gamma^{\frac{\alpha}{\alpha+1}}.$$

Note that the Poisson-gamma mixture has positive probability mass $\Pr(Y = 0) = \exp(-\lambda)$ at zero and a continuous density on the positive real line, that is,

$$f(y; \mu, \phi, \zeta) = e^{-\lambda}\delta_0(y) + \sum_{j=1}^{\infty} \frac{\lambda^j e^{-\lambda}}{j!} \frac{y^{j\alpha-1}e^{-y/\gamma}}{\gamma^{j\alpha}\Gamma(j\alpha)},$$

where $\delta_0(\cdot)$ is the Dirac delta function at zero. Moreover, $\mu = \mathbb{E}(Y) = \lambda\alpha\gamma$ is always positive.

In applications of the Tweedie model, one is interested in studying the relationship between a response Y, assumed to follow a Tweedie distribution, and a set of explanatory variables $\mathbf{x} = (X_1, \ldots, X_p)^\intercal$. Following the generalized linear model framework, it is often assumed in the compound Poisson-gamma model that

$$\log(\mu(\mathbf{x})) = \log(\mathbb{E}(Y \mid \mathbf{x})) = \beta_0 + \mathbf{x}^\intercal\boldsymbol{\beta}$$

for some unknown parameters β_0 and $\boldsymbol{\beta}$ when the logarithmic link is employed. The conditional distribution of Y given $\mathbf{x}$ is then

$$\begin{aligned} f(y \mid \mu, \phi, \zeta) &= c(y, \phi) \exp\left\{\frac{1}{\phi}\left(\frac{y\mu^{1-\zeta}(\mathbf{x})}{1-\zeta} - \frac{\mu^{2-\zeta}(\mathbf{x})}{2-\zeta}\right)\right\} \\ &= c(y, \phi) \exp\left\{\frac{1}{\phi}\left(\frac{y\exp[(1-\zeta)(\beta_0 + \mathbf{x}^\intercal\boldsymbol{\beta})]}{1-\zeta} - \frac{\exp[(2-\zeta)(\beta_0 + \mathbf{x}^\intercal\boldsymbol{\beta})]}{2-\zeta}\right)\right\}. \end{aligned} \tag{3.2}$$

Given independent observations $(\mathbf{x}_i, y_i)_{i=1}^n$ from the above model, the likelihood of the parameters can be obtained from the joint distribution function of the observations based on (3.2), which can be then used to obtain the maximum likelihood estimates (MLEs) of the parameters. Given ζ, since the MLEs of $(\beta_0, \boldsymbol{\beta})$ and ϕ are asymptotically independent, $(\beta_0, \boldsymbol{\beta})$ and ϕ are usually estimated separately, where the MLEs of the mean parameters $(\beta_0, \boldsymbol{\beta})$ are obtained from minimizing the negative log-likelihood (up to constant terms not related to β_0 and $\boldsymbol{\beta}$)

$$\ell_n(\beta_0, \boldsymbol{\beta}; \zeta) = \sum_{i=1}^{n} w_i \left\{\frac{y_i \exp[-(\zeta-1)(\beta_0 + \mathbf{x}_i^\intercal\boldsymbol{\beta})]}{\zeta - 1} + \frac{\exp[(2-\zeta)(\beta_0 + \mathbf{x}_i^\intercal\boldsymbol{\beta})]}{2-\zeta}\right\}, \tag{3.3}$$

where $w_i \geq 0$ is the weight for the ith observation for $i = 1, \ldots, n$. The dispersion parameter ϕ can be estimated by MLE (Smyth, 1996) or by moment estimation (Dunn and Smyth, 2005). In practice, the estimation of ζ is usually performed via profile likelihood (Dunn and Smyth, 2005).

When $\mathbf{x}$ is of high dimension, it is often necessary to do variable selection to avoid noise accumulation and to enhance interpretability. In the current literature, a common approach is regularizing the likelihood function via various penalties such as the lasso, SCAD, MCP and so on. More concretely, since we are mainly interested in the mean parameters, we estimate $(\beta_0, \boldsymbol{\beta})$ from minimizing the penalized negative log-likelihood

$$\ell_n(\beta_0, \boldsymbol{\beta}; \zeta) + \sum_{j=1}^{p} p_\lambda(|\beta_j|), \tag{3.4}$$

where $p_\lambda(\cdot)$ denotes the penalty function applied component-wisely to $\boldsymbol{\beta}$ and $\lambda \geq 0$ is the regularization parameter. In particular, the lasso, SCAD ($a > 2$) and MCP ($a > 1$) penalties are respectively

$$p_\lambda^{\text{lasso}}(|u|) = \lambda|u|,$$

$$p_\lambda^{\text{SCAD}}(|u|; a) = \lambda|u| I(|u| \leq \lambda) + \frac{a\lambda|u| - \frac{|u|^2+\lambda^2}{2}}{a-1} I(\lambda < |u| \leq a\lambda) + \frac{(a+1)\lambda^2}{2} I(|u| > a\lambda),$$

and

$$p_\lambda^{\text{MCP}}(|u|; a) = \left(\lambda|u| - \frac{|u|^2}{2a}\right) I(|u| \leq a\lambda) + \frac{1}{2}a\lambda^2 I(|u| > a\lambda).$$

Under different penalties, the sparse solutions obtained from (3.4) typically require different sets of conditions to be variable selection consistent. In practice, it is often hard to check if one set of conditions are satisfied for the real data at hand as most of them involve unknown parameters of the data distribution. It is therefore important to come up with

a metric that can check the selection performance of a given variable selection method without reference to the hard-to-check conditions. This is described in the next section.

3.3 Variable Selection Diagnostic Measures

We briefly describe the variable selection deviation (VSD) measures proposed by Nan and Yang (2014) and review the F- and G-measures by Yu et al. (2022) as alternative variable selection diagnostic measures. All of these measures provide a practical means of evaluating the variable selection accuracy of a given statistical procedure, based on the ensemble of its weighted deviations (in terms of variable selection) from multiple candidate models. Since the weights are adaptive in the sense that they are calculated based on the predictive performance of the candidate models, these diagnostic measures are reasonable metrics to use in practice for evaluating variable selection performance in problems where the truth is unknown.

Let $\mathbb{S} = \{m_k, k = 1, \ldots, K\}$ be a set of candidate models for fitting the given data $\mathbf{D} = \{Z_i, i = 1, \ldots, n\}$, where $K > 1$ is an integer and $Z_i = (\mathbf{x}_i, y_i)$ with $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\intercal$. For simplicity, we assume that each candidate model is represented by a subset of the given predictors in $\mathbf{x}$. Let m^0 be a model whose variable selection accuracy is to be evaluated and m^* the true model whose sparsity level (the true number of active predictors) is denoted by s^*. According to Nan and Yang (2014), given a vector of weights $\underline{w} = (w_1, \ldots, w_K)^\intercal$, the variable selection deviation (VSD) of m^0 with respect to the weighting $\underline{w}$ on the models in $\mathbb{S}$ is given by

$$\widehat{\mathrm{VSD}}(m^0) = \sum_{m_k \in \mathbb{S}} w_k \cdot |m_k \nabla m^0|, \tag{3.5}$$

where ∇ denotes the symmetric difference (of two sets) and $|m|$ denotes the cardinality of the set m. The upper and lower VSD of m^0 are similarly defined, respectively, as

$$\widehat{\text{VSD}}^{+}(m^0) = \sum_{m_k \in \mathbb{S}} w_k \cdot |m_k \backslash m^0| \quad \text{and} \quad \widehat{\text{VSD}}^{-}(m^0) = \sum_{m_k \in \mathbb{S}} w_k \cdot |m^0 \backslash m_k|, \tag{3.6}$$

where $A \backslash B = A \cap B^c$ denotes the set difference of A and B. Yu et al. (2022) note that the VSD measures by Nan and Yang (2014) are related to pure counts of the false positives and false negatives of a variable selection method, which may depend on the size of the true model. This means the absolute magnitude of these measures alone may not be able to directly give a sense of the actual variable selection performance. For example, the performance of a model m^0 with $|m^0 \nabla m^*| = 10$ may be interpreted very differently given $|m^*| = 11$ from, say, given $|m^*| = 100$. To that end, Yu et al. (2022) propose the F- and G-measures as integrative summaries of both the precision and recall of a variable selection method. See also Chapter 2 of this dissertation. Specifically, the F- and G-measures are estimated by

$$\widehat{F}(m^0) = \sum_{m_k \in \mathbb{S}} w_k \cdot \frac{2|m^0 \cap m^k|}{(|m^0| + |m^k|) \vee 1} \tag{3.7}$$

and

$$\widehat{G}(m^0) = \sum_{m_k \in \mathbb{S}} w_k \cdot \frac{|m^0 \cap m^k|}{\sqrt{(|m^0| \cdot |m^k|) \vee 1}}, \tag{3.8}$$

where $a \vee b = \max(a, b)$.

Note that $\widehat{\text{VSD}}(m^0)$, $\widehat{F}(m^0)$ and $\widehat{G}(m^0)$ can be viewed as weighted averages of the various deviations of m^0 from the models in $\mathbb{S}$. In order for the estimates of these variable selection diagnostic measures to be an accurate, ideally we want the set of candidate models $\mathbb{S}$ to be

rich enough to include m^* if possible and to put more weights on models closer to the true model m^*, that is, models where $|m_k \nabla m^*|$ is small. Since m^* is unknown in practice, the weights w_k's are often adaptively computed based on the model performance of the m_k's on the data Z. The set \$, according to Nan and Yang (2014) and Yu et al. (2022), is often constructed by taking the union of the solution paths from regularized regressions using, for example, the lasso, SCAD and MCP penalties. This often dramatically reduces the number of candidate models one needs to consider compared to an exhaustive approach which includes the set of all possible submodels.

Three weighting methods are considered by Nan and Yang (2014) and Yu et al. (2022) to adaptively compute the weights w_k's. Specifically, they are information-criterion-based weighting, and adaptive-regression-by-mixing (ARM, Yang, 2001) based weighting. Based on information criteria such as AIC (Akaike, 1973) and BIC (Schwarz, 1978b), the information-criterion-based weighting approach computes the weights as

$$w_k = \frac{\exp(-\frac{1}{2}I_k - \psi C_k)}{\sum_{i=1}^{K} \exp(-\frac{1}{2}I_i - \psi C_i)} \text{ for some } \psi \geq 0, k = 1, \ldots, K, \tag{3.9}$$

where $I_k = -2\log(L_k) + q_k$ is the information criterion, L_k is the maximized likelihood of model m_k, q_k is a penalty function of the model size $s_k = |m_k|$ of m_k, and $C_k = s_k \log(ep/s_k) + 2\log(s_k+2)$ is the prior weight used for adjusting the model complexity from an information-theoretical point of view. For AIC, $q_k = 2s_k$, and for BIC, $q_k = s_k \log(n)$. The constant ψ controls the relative importance between the information criterion and the prior weight. In particular, $\psi = 0$ corresponds to the uniform prior. The model-mixing-based weighting approach relies on data splitting and will be articulated in detail for each of the response distributions we consider in the sequel.

3.4 Variable Selection Diagnostics in Tweedie Models

Though the idea of variable selection diagnosis can be applied to almost all likelihood-based models, Nan and Yang (2014) mainly focus on the VSD of the least squares regression. The VSD of the logistic regression is considered by Yu et al. (2022) and Chapter 2 of this dissertation. To our knowledge, the VSD of other likelihood models, especially of other generalized linear models, have not yet been fully explored in the literature. In this section, we generalize the variable selection diagnostic tools to handle more response types, and in particular the Tweedie family of distributions. For simplicity, we will mainly discuss the Poisson and compound Poisson-gamma cases due to their wide applications.

3.4.1 Poisson models

The Poisson regression is often used as a benchmark model for count data. Under the generalized linear model framework, it is often assumed that the response variable Y has a Poisson distribution, and a link function connects the expected value of Y to a linear combination of the predictors $\mathbf{x} = (X_1, \ldots, X_p)^\intercal$. For ease of presentation, we focus on the canonical logarithmic link (other link functions can be used as well) and assume $(Y \mid \mathbf{x})$ follows a Poisson distribution with

$$\log(\mathbb{E}(Y \mid \mathbf{x})) = \log(\lambda(\mathbf{x})) = \beta_0^* + \mathbf{x}^\intercal \boldsymbol{\beta}^*, \tag{3.10}$$

where $\beta_0^* \in \mathbb{R}$ and $\boldsymbol{\beta}^* \in \mathbb{R}^p$. Let $\mathcal{A}^* = \text{supp}(\boldsymbol{\beta}^*) \equiv \{j : \beta_j^* \neq 0, 1 \leq j \leq p\}$ be the support of the true coefficient vector $\boldsymbol{\beta}^*$. We further assume true model is sparse so that $|\mathcal{A}^*| < p$.

Given independent observations $(\mathbf{x}_i, y_i)_{i=1}^n$, the coefficients are often estimated by the

maximum likelihood estimation

$$\min_{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p} -\ell(\beta_0,\boldsymbol{\beta}) = \sum_{i=1}^{n}\left[\lambda(\mathbf{x}_i) - y_i\log\lambda(\mathbf{x}_i) + \log(y_i!)\right],$$

where $\lambda(\mathbf{x}_i) = \exp(\beta_0 + \mathbf{x}_i^\intercal\boldsymbol{\beta})$ is the mean function for the ith observation (note that $\lambda(\mathbf{x}_i) = g^{-1}(\beta_0 + \mathbf{x}_i^\intercal\boldsymbol{\beta})$ if a general link function $g(\cdot)$ is used). This generally works in the low-dimensional regime. However, when the dimension is high and even higher than the sample size, we typically need to consider the regularized maximum likelihood estimation

$$\min_{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p} -\ell(\beta_0,\boldsymbol{\beta}) + \sum_{j=1}^{p} p_\lambda(|\beta_j|), \tag{3.11}$$

where $p_\lambda(|\cdot|)$ is the penalty function with tuning parameter $\lambda \geq 0$. In practice, we find the lasso, SCAD and MCP penalties work reasonably well in most scenarios.

Candidate models

Since the exhaustive list of all possible submodels grows exponentially with the number of predictors, we instead consider the list of submodels formed by the solution paths of the regularized Poisson regression (3.11) for feasible purposes. Specifically, we compute the submodels $\mathbb{S} = \{\mathcal{A}(\lambda) : \lambda = \lambda_1, \ldots, \lambda_L\}$ for a decreasing sequence $\lambda_1 > \cdots > \lambda_L > 0$ of the tuning parameter, where $\mathcal{A}(\lambda) = \{j : \hat{\beta}_j(\lambda) \neq 0, 1 \leq j \leq p\}$ and $\widehat{\boldsymbol{\beta}}(\lambda)$ is the regularized estimator at tuning parameter value λ. This can be done for each type of penalty functions, such as lasso, SCAD and MCP, and finally we compile the set of candidate models by taking the union of the solution paths from several types of penalization.

We note that the lasso regression can be fitted using the popular **glmnet** package (Friedman et al., 2010). However, for the nonconvex penalized regressions, there are

no reliable software packages for fitting them to the best of our knowledge. Although the **ncvreg** package (Breheny, 2011) supports the Poisson family using SCAD and MCP penalties, our empirical studies have encountered too many convergence issues with their implementation. Instead, we consider the local linear approximation (LLA, Zou and Li, 2008; Fan et al., 2014) approach to solving the nonconvex regularized Poisson regression problem. It converts the nonconvex problem into a few steps of weighted lasso penalized problems, where each step can be solved quickly using the robust implementation of **glmnet**. The LLA algorithm for fitting the nonconvex penalized Poisson regression is shown as follows:

(1) Initialize $(\beta_0, \boldsymbol{\beta})$ with $(\hat{\beta}_0^{(0)}, \widehat{\boldsymbol{\beta}}^{(0)})$. Compute weights

$$\hat{w}_j^{(0)} = p'_\lambda(|\hat{\beta}_j^{(0)}|),\ j = 1, \ldots, p.$$

(2) For $m = 1, 2, \ldots, M$, repeat the LLA iterations in (2.a) and (2.b).

(2.a) Solve the following weighted lasso penalized Poisson regression

$$(\hat{\beta}_0^{(m)}, \widehat{\boldsymbol{\beta}}^{(m)}) := \arg\min_{\beta_0, \boldsymbol{\beta}} -\ell(\beta_0, \boldsymbol{\beta}) + \sum_{j=1}^{p} \hat{w}_j^{(m-1)} |\beta_j|.$$

(2.b) Calculate the weights $\hat{w}_j^{(m)} = p'_\lambda(|\hat{\beta}_j^{(m)}|),\ j = 1, \ldots, p.$

As is recommended by Fan et al. (2014), we take $(\hat{\beta}_0^{(0)}, \widehat{\boldsymbol{\beta}}^{(0)})$ to be the cross-validated Poisson lasso estimator and set $M = 2$. Note that the optimization problem in Step (2.a) is a weighted lasso problem since the weights $\hat{w}_j^{(m-1)}$'s are all nonnegative. It can be very efficiently solved by the coordinate descent algorithm implemented in the **glmnet** package.

Weighting methods

There are various ways of assigning weights to the candidate models, including the information-criterion-based weighting, Bayesian model averaging, adaptive regression by mixing (ARM), and fiducial-based weighting. These are detailed in Chapter 2. We mainly focus on the information-criterion-based and ARM-based weighting schemes due to their easy implementation and good empirical performance.

Let $\boldsymbol{\beta}_s^{(k)}$ be the subvector representing the nonzero components of $\boldsymbol{\beta}^{(k)}$ of model $\mathcal{A}^k$, and let $\mathbf{x}_s^{(k)} \in \mathbb{R}^{|\mathcal{A}^k|}$ be the corresponding subset of active variables in model $\mathcal{A}^k$ for $k = 1, \ldots, K$. The ARM weighting method for the Poisson regression models is summarized in Algorithm 3. When $\psi = 0$, we denote the algorithm as ARM, while if $\psi > 0$, we denote it as ARM-p to signify the effect of the prior weight on the final weight calculation.

Algorithm 3: ARM weighting procedure for Poisson regression.

Randomly split $\mathbf{D}$ into a training set $\mathbf{D}_1$ and a test set $\mathbf{D}_2$ of (approximately) equal size.
For each $\mathcal{A}^k \in \mathbb{S}$, fit a standard Poisson regression of y on $\mathbf{x}_s^{(k)}$ using the training set $\mathbf{D}_1$ and get the estimated regression coefficients $\hat{\beta}_0^{(k)}$ and $\widehat{\boldsymbol{\beta}}_s^{(k)}$.
For each $\mathcal{A}^k$, compute the predictions $\hat{\beta}_0^{(k)} + \mathbf{x}_s^{(k)\intercal}\widehat{\boldsymbol{\beta}}_s^{(k)}$ on the test set $\mathbf{D}_2$. Hence, we have $\widehat{\lambda}^{(k)}(\mathbf{x}) = \exp(\hat{\beta}_0^{(k)} + \mathbf{x}_s^{(k)\intercal}\widehat{\boldsymbol{\beta}}_s^{(k)})$ [and $\widehat{\lambda}^{(k)}(\mathbf{x}) = g^{-1}(\hat{\beta}_0^{(k)} + \mathbf{x}_s^{(k)\intercal}\widehat{\boldsymbol{\beta}}_s^{(k)})$ if link $g(\cdot)$ is used instead].
Compute the weight w_k for each candidate model $\mathcal{A}^k$:

$$w_k = \frac{e^{-\psi C_k}\prod_{(\mathbf{x}_i,y_i)\in\mathbf{D}_2}\frac{(\hat{\lambda}^{(k)}(\mathbf{x}_i))^{y_i}e^{-\hat{\lambda}^{(k)}(\mathbf{x}_i)}}{y_i!}}{\sum_{\ell=1}^{K} e^{-\psi C_\ell}\prod_{(\mathbf{x}_i,y_i)\in\mathbf{D}_2}\frac{(\hat{\lambda}^{(\ell)}(\mathbf{x}_i))^{y_i}e^{-\hat{\lambda}^{(\ell)}(\mathbf{x}_i)}}{y_i!}}, k = 1, \ldots, K,$$

where the prior weight C_k, for $k = 1, \ldots, K$, is defined in (2.7).
Repeat the steps above (with random data splitting) L times to get $w_k^{(l)}$, for $l = 1, \ldots, L$, and return the final weight $w_k = L^{-1}\sum_{l=1}^{L} w_k^{(l)}$.

Let $\hat{\lambda}^{(k)}(\mathbf{x}) = \exp(\hat{\beta}_0^{(k)} + \mathbf{x}_s^{(k)\intercal}\widehat{\boldsymbol{\beta}}_s^{(k)})$ (again $\widehat{\lambda}^{(k)}(\mathbf{x}) = g^{-1}(\hat{\beta}_0^{(k)} + \mathbf{x}_s^{(k)\intercal}\widehat{\boldsymbol{\beta}}_s^{(k)})$ if link $g(\cdot)$ is used) be the estimation of the mean function at $\mathbf{x}$ from model $\mathcal{A}^k$ and

$$I_k = -2\ell(\hat{\beta}_0^{(k)}, \widehat{\boldsymbol{\beta}}^{(k)}) + |\mathcal{A}^k| \cdot q = 2\sum_{i=1}^{n}\left[\hat{\lambda}^{(k)}(\mathbf{x}_i) - y_i \log\hat{\lambda}^{(k)}(\mathbf{x}_i) + \log(y_i!)\right] + |\mathcal{A}^k| \cdot q$$

be the information criterion of model $\mathcal{A}^k$. For AIC $q = 2$ and for BIC $q = \log(n)$. The weights are then given by

$$w_k = \frac{\exp(-\frac{1}{2}I_k - \psi C_k)}{\sum_{\ell=1}^{K} \exp(-\frac{1}{2}I_\ell - \psi C_\ell)}, k = 1, \dots, K.$$

When $\psi = 0$, we call the weighting method AIC or BIC depending on the value of q being used, and when $\psi > 0$, we call it AIC-p or BIC-p.

3.4.2 Compound Poisson-gamma models

The compound Poisson-gamma model is widely used in practice to model continuous data with a spike at zero. It is commonly used to model the claim amount in insurance pricing, the precipitation in ecological studies, and the alcohol consumption in health studies. Given a vector of predictor variables $\mathbf{x} = (X_1, \dots, X_p)^\intercal$, assume the response variable Y follows a compound Poisson-gamma distribution, or equivalently according to Section 3.2, a Tweedie distribution with power parameter $\zeta \in (1, 2)$. Specifically, we assume $(Y \mid \mathbf{x})$ has the distribution function (3.2) with $\mu(\mathbf{x}) = \mathbb{E}(Y \mid \mathbf{x}) = \exp(\beta_0 + \mathbf{x}^\intercal \boldsymbol{\beta})$. In many Tweedie modeling problems, the main interest is the estimation of the mean function $\mu(\mathbf{x})$, which according to our assumption is equivalent to the estimation of the parameters β_0 and $\boldsymbol{\beta}$. We further assume that we have independent observations $\mathbf{D} = \{Z_i = (\mathbf{x}_i, y_i), i = 1, \dots, n\}$ sampled according to the above model with true parameter values β_0^* and $\boldsymbol{\beta}^*$. Then, when the dimension is low, the parameters β_0 and $\boldsymbol{\beta}$ are often estimated by the maximum likelihood estimation

$$\min_{\beta_0 \in \mathbb{R}, \boldsymbol{\beta} \in \mathbb{R}^p} -\ell(\beta_0, \boldsymbol{\beta}; \zeta) = \sum_{i=1}^{n} \left\{ \frac{y_i \exp[-(\zeta - 1)(\beta_0 + \mathbf{x}_i^\intercal \boldsymbol{\beta})]}{\zeta - 1} + \frac{\exp[(2 - \zeta)(\beta_0 + \mathbf{x}_i^\intercal \boldsymbol{\beta})]}{2 - \zeta} \right\},$$

where ζ is either known *apriori* or estimated from the profile likelihood (Dunn and Smyth, 2005). When the dimension is high, we usually consider the regularized Tweedie regression

$$\min_{\beta_0 \in \mathbb{R}, \boldsymbol{\beta} \in \mathbb{R}^p} -\ell(\beta_0, \boldsymbol{\beta}; \zeta) + \sum_{j=1}^{p} p_\lambda(|\beta_j|),$$

where $p_\lambda(|\cdot|)$ is the penalty function with tuning parameter $\lambda \geq 0$. In this work, we mainly use the lasso, SCAD and MCP penalties.

Candidate models

Like in other generalized linear models, we create the set of candidate models by considering the solution paths of the regularized Tweedie regression models using the lasso, SCAD and MCP penalties. For each type of penalty, we consider the submodels $\mathbb{S} = \{\mathcal{A}(\lambda) : \lambda = \lambda_1, \ldots, \lambda_L\}$ for a decreasing sequence of tuning parameters $\lambda_1 > \cdots > \lambda_L > 0$, where $\mathcal{A}(\lambda) = \{j : \hat{\beta}_j \neq 0, 1 \leq j \leq p\}$ with $\widehat{\boldsymbol{\beta}}(\lambda)$ being the regularized estimator at tuning parameter λ.

The lasso penalized Tweedie regression can be fitted efficiently using the `HDtweedie()` function from the **HDtweedie** package (Qian et al., 2016). The nonconvex penalized Tweedie regression, however, has no out-of-box implementation to the best of our knowledge. We thus consider again the LLA approach by Zou and Li (2008) and Fan et al. (2014) to utilize the efficient coordinate descent algorithm in the **HDtweedie** package. We outline the LLA algorithm for the nonconvex penalized Tweedie regression below:

(1) Initialize $(\beta_0, \boldsymbol{\beta})$ with $(\hat{\beta}_0^{(0)}, \widehat{\boldsymbol{\beta}}^{(0)})$. Compute weights

$$\hat{w}_j^{(0)} = p'_\lambda(|\hat{\beta}_j^{(0)}|), \ j = 1, \ldots, p.$$

(2) For $m = 1, 2, \ldots, M$, repeat the LLA iterations in (2.a) and (2.b).

(2.a) Solve the following weighted lasso penalized Poisson regression

$$(\hat{\beta}_0^{(m)}, \widehat{\boldsymbol{\beta}}^{(m)}) := \arg\min_{\beta_0, \boldsymbol{\beta}} -\ell(\beta_0, \boldsymbol{\beta}; \zeta) + \sum_{j=1}^{p} \hat{w}_j^{(m-1)} |\beta_j|.$$

(2.b) Calculate the weights $\hat{w}_j^{(m)} = p'_\lambda(|\hat{\beta}_j^{(m)}|),\ j = 1, \ldots, p$.

We take $(\hat{\beta}_0^{(0)}, \widehat{\boldsymbol{\beta}}^{(0)})$ to be the cross-validated Tweedie lasso estimator and set $M = 2$.

Weighting methods

Similar to the Poisson case, we summarize the ARM weighting method for the compound Poisson-gamma (Tweedie) regression in Algorithm 4. Here for simplicity, we assume that ζ is given or obtained elsewhere.

Algorithm 4: ARM weighting procedure for Tweedie regression.

Randomly split $\mathbf{D}$ into a training set $\mathbf{D}_1$ and a test set $\mathbf{D}_2$ of (approximately) equal size.
For each $\mathcal{A}^k \in \mathbb{S}$, fit a standard compound Poisson-gamma regression of y on $\mathbf{x}_s^{(k)}$ using the training set $\mathbf{D}_1$ and get the estimated regression coefficients $\hat{\beta}_0^{(k)}$ and $\widehat{\boldsymbol{\beta}}_s^{(k)}$, and the estimated dispersion $\hat{\phi}^{(k)}$.
For each $\mathcal{A}^k$, compute the predictions $\hat{\beta}_0^{(k)} + \mathbf{x}_s^{(k)\intercal} \widehat{\boldsymbol{\beta}}_s^{(k)}$ on the test set $\mathbf{D}_2$. Hence, we have $\widehat{\mu}^{(k)}(\mathbf{x}) = \exp(\hat{\beta}_0^{(k)} + \mathbf{x}_s^{(k)\intercal} \widehat{\boldsymbol{\beta}}_s^{(k)})$.
Compute the weight w_k for each candidate model $\mathcal{A}^k$:

$$w_k = \frac{e^{-\psi C_k} \prod_{(\mathbf{x}_i, y_i) \in \mathbf{D}_2} f(y_i \mid \hat{\mu}^{(k)}(\mathbf{x}_i), \hat{\phi}^{(k)}, \zeta)}{\sum_{\ell=1}^{K} e^{-\psi C_\ell} \prod_{(\mathbf{x}_i, y_i) \in \mathbf{D}_2} f(y_i \mid \hat{\mu}^{(\ell)}(\mathbf{x}_i), \hat{\phi}^{(\ell)}, \zeta)}, k = 1, \ldots, K,$$

where C_k, for $k = 1, \ldots, K$ is defined in (2.7) and $f(y \mid \mu, \phi, \zeta)$ is defined in (3.2).
Repeat the steps above (with random data splitting) L times to get $w_k^{(l)}$, for $l = 1, \ldots, L$, and get $w_k = L^{-1} \sum_{l=1}^{L} w_k^{(l)}$.

In practice, the dispersion parameter ϕ is often estimated using moment estimation (with the deviance) or the maximum likelihood estimation (Dunn and Smyth, 2005).

Let $I_k = -2 \sum_{i=1}^{n} \log f(y_i \mid \hat{\mu}^{(k)}(\mathbf{x}_i), \hat{\phi}^{(k)}, \zeta) + |\mathcal{A}^k| \cdot q$ be the information criterion of the Tweedie submodel $\mathcal{A}^k$, where $q = 2$ for AIC and $q = \log(n)$ for BIC. The information-

criterion-based weights are given by

$$w_k = \frac{\exp(-\frac{1}{2}I_k - \psi C_k)}{\sum_{\ell=1}^{K} \exp(-\frac{1}{2}I_\ell - \psi C_\ell)}, k = 1, \ldots, K.$$

Note that although the preceding discussion assumes a logarithmic link, similar weighting methods can be easily developed for a general link $g(\cdot)$. Our software implementation (discussed in Chapter 4) works for most of the commonly used link functions in the generalized linear models.

3.5 Numerical Studies

To study the performance of the variable selection diagnostic measures of the Poisson and Tweedie models, we conduct simulation studies under various settings, including both $n \geq p$ and $n < p$ cases. We also consider independent and correlated features.

3.5.1 Setting I: Poisson models

We generate Y from a Poisson distribution with mean function $\lambda(\mathbf{x}) = \mathbf{x}^\intercal \boldsymbol{\beta}^*$. We vary the sample size and dimension of the model to study their effects on the estimation. We also set different correlation structures for $\mathbf{x}$.

Example 3.1

$n = 200$, $p = 8$, and $\boldsymbol{\beta}^* = (3, 1.5, 0, 0, -2, \mathbf{0}_3)^\intercal$. The components of $\mathbf{x}$ are independent and identically distributed (i.i.d.) from the uniform distribution on $(0, 1)$. □

Example 3.2

Same as Example 3.1, except $n = 1000$. □

Example 3.3

$n = 200$, $p = 500$, and $\boldsymbol{\beta}^* = (3, 1.5, 0, 0, -2, \mathbf{0}_{495})^\intercal$. The components of $\mathbf{x}$ are i.i.d. from Unif$(0, 1)$. □

Example 3.4

$n = 200$, $p = 30$, and $\boldsymbol{\beta}^* = (3, 1.5, 0, 0, -2, \mathbf{0}_{25})^\intercal$. The components of $\mathbf{x}$ are marginally distributed as Unif$(0, 1)$ but have correlation matrix $\boldsymbol{\rho} = (0.3^{|j-j'|})_{p\times p}$. Note that to sample the random vector $\mathbf{x} = (X_1, \ldots, X_p)^\intercal$, we first sample $\mathbf{x}' = (X_1', \ldots, X_p')^\intercal \sim N(\mathbf{0}, \Sigma)$ where $\Sigma_{jj'} = 2\sin[\frac{\pi}{6}(0.3)^{|j-j'|}]$, and set $X_j = \Phi(X_j')$ for $j = 1, \ldots, p$, where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution. This uses the fact that for two standard normal random variables Z_1 and Z_2, $\text{corr}(\Phi(Z_1), \Phi(Z_2)) = \frac{6}{\pi}\arcsin(\frac{1}{2}\text{corr}(Z_1, Z_2))$ by Spearman's rank correlation. □

Example 3.5

$n = 200$, $p = 200$, and $\boldsymbol{\beta}^* = (3, 1.5, 0, 0, -2, \mathbf{0}_{195})^\intercal$. The components of $\mathbf{x}$ are marginally distributed as Unif$(0, 1)$ but have correlation matrix $\boldsymbol{\rho} = (0.6^{|j-j'|})_{p\times p}$. □

In all of the above settings, we evaluate the variable selection performance of four optimally tuned penalized Poisson regression models using the lasso, adaptive lasso, SCAD and MCP penalties, respectively, where the tuning is done by five-fold cross-validation. The candidate models are obtained from the solution paths of the lasso, adaptive lasso, SCAD and MCP penalized Poisson regression models at sequences of tuning parameters. The nonconvex (i.e., SCAD and MCP) penalized Poisson regression models are fitted using the two-step LLA algorithm described in Section 3.4.1.

Because we know the true model $\mathcal{A}^* = \{j : \beta_j^* \neq 0, 1 \leq j \leq p\}$ in this simulation, we report the true VSD$(\mathcal{A}^0)$, VSD$^+(\mathcal{A}^0)$, VSD$^-(\mathcal{A}^0)$, $F(\mathcal{A}^0)$ and $G(\mathcal{A}^0)$ measures, besides

their estimated counterparts, for each model $\mathcal{A}^0 \in \{\mathcal{A}^{\text{lasso}}, \mathcal{A}^{\text{AdaLasso}}, \mathcal{A}^{\text{MCP}}, \mathcal{A}^{\text{SCAD}}\}$. For comparison, we compute these measures using six different weighting methods: ARM, AIC, BIC, ARM-p, AIC-p, and BIC-p, where $\psi = 1$ is used for those using prior adjustment. The number of observations used in the training part to compute the ARM weights is roughly half of the sample size, i.e., $\lfloor n/2 \rfloor$, and the corresponding repetition time is one hundred.

All simulation cases are repeated one hundred times, and the corresponding values are computed and averaged. We summarize the results in Tables 3.1 – 3.5 for Examples 3.1 – 3.5. The standard errors are reported in the parentheses. As shown in the tables, the estimates $\widehat{\text{VSD}}(\mathcal{A}^0)$, $\widehat{\text{VSD}}^{+}(\mathcal{A}^0)$, $\widehat{\text{VSD}}^{-}(\mathcal{A}^0)$, $\widehat{F}(\mathcal{A}^0)$ and $\widehat{G}(\mathcal{A}^0)$ can reasonably approximate their true values. In particular, the estimates using prior adjustments ($\psi = 1$) are generally better than those without prior adjustment, with BIC-p performing the best in most cases. Comparing results from Examples 3.1 – 3.5, we can see that generally the performance of all four variable selection methods (i.e., lasso, adaptive lasso, MCP and SCAD) becomes better when the sample size increases and becomes worse when the dimension increases or when the correlations among the predictors increase. However, our VSD estimates can always estimate the true deviations very accurately. Since the signals are strong in the simulation settings, we observe that the adaptive penalties (adaptive lasso, MCP and SCAD) generally give better results than the lasso in terms of variable selection performance.

3.5.2 Setting II: Tweedie models

We generate Y from the compound Poisson-gamma distribution with mean function $\mu(\mathbf{x}) = \mathbf{x}^\intercal \boldsymbol{\beta}^*$. Since our main interest is in the mean function or in $\boldsymbol{\beta}$, we fix the power parameter at $\zeta = 1.5$ and dispersion parameter at $\phi = 1$ for simplicity.

Table 3.1: Poisson case (Example 3.1).

	VSD	VSD^+	VSD^-	F	G
			Lasso		
True	3.050 (0.164)	0.000 (0.000)	3.050 (0.164)	0.688 (0.014)	0.729 (0.012)
ARM	2.013 (0.083)	0.233 (0.034)	1.781 (0.109)	0.809 (0.006)	0.829 (0.005)
AIC	1.892 (0.093)	0.146 (0.023)	1.746 (0.110)	0.822 (0.007)	0.840 (0.006)
BIC	2.733 (0.148)	0.011 (0.003)	2.722 (0.150)	0.727 (0.013)	0.761 (0.011)
ARM-p	2.524 (0.131)	0.053 (0.009)	2.471 (0.138)	0.751 (0.011)	0.780 (0.009)
AIC-p	2.486 (0.134)	0.028 (0.006)	2.458 (0.138)	0.755 (0.011)	0.784 (0.010)
BIC-p	2.932 (0.157)	0.003 (0.001)	2.930 (0.158)	0.703 (0.014)	0.741 (0.012)
			AdaLasso		
True	0.200 (0.062)	0.000 (0.000)	0.200 (0.062)	0.975 (0.007)	0.977 (0.007)
ARM	1.417 (0.033)	1.360 (0.040)	0.057 (0.023)	0.837 (0.004)	0.854 (0.003)
AIC	1.308 (0.054)	1.279 (0.057)	0.029 (0.015)	0.846 (0.006)	0.861 (0.005)
BIC	0.358 (0.039)	0.248 (0.018)	0.110 (0.041)	0.954 (0.004)	0.957 (0.004)
ARM-p	0.642 (0.034)	0.537 (0.024)	0.105 (0.036)	0.921 (0.004)	0.928 (0.003)
AIC-p	0.567 (0.040)	0.494 (0.035)	0.073 (0.029)	0.929 (0.004)	0.935 (0.004)
BIC-p	0.232 (0.050)	0.078 (0.006)	0.155 (0.052)	0.970 (0.006)	0.973 (0.005)
			MCP		
True	0.220 (0.070)	0.000 (0.000)	0.220 (0.070)	0.973 (0.008)	0.976 (0.007)
ARM	1.432 (0.038)	1.357 (0.044)	0.075 (0.032)	0.836 (0.004)	0.853 (0.003)
AIC	1.339 (0.059)	1.284 (0.063)	0.054 (0.026)	0.843 (0.006)	0.859 (0.006)
BIC	0.400 (0.051)	0.260 (0.021)	0.141 (0.053)	0.949 (0.005)	0.953 (0.005)
ARM-p	0.672 (0.042)	0.542 (0.027)	0.129 (0.047)	0.918 (0.004)	0.925 (0.004)
AIC-p	0.616 (0.050)	0.508 (0.040)	0.108 (0.043)	0.924 (0.005)	0.930 (0.005)
BIC-p	0.267 (0.061)	0.085 (0.009)	0.182 (0.062)	0.967 (0.006)	0.969 (0.006)
			SCAD		
True	0.230 (0.068)	0.000 (0.000)	0.230 (0.068)	0.972 (0.008)	0.975 (0.007)
ARM	1.420 (0.034)	1.346 (0.044)	0.074 (0.025)	0.837 (0.004)	0.854 (0.003)
AIC	1.299 (0.053)	1.260 (0.058)	0.040 (0.014)	0.847 (0.006)	0.862 (0.005)
BIC	0.411 (0.045)	0.260 (0.021)	0.151 (0.048)	0.947 (0.005)	0.951 (0.005)
ARM-p	0.671 (0.038)	0.537 (0.027)	0.134 (0.042)	0.918 (0.004)	0.925 (0.004)
AIC-p	0.596 (0.040)	0.493 (0.035)	0.103 (0.034)	0.925 (0.005)	0.932 (0.004)
BIC-p	0.288 (0.059)	0.090 (0.010)	0.197 (0.061)	0.964 (0.007)	0.967 (0.006)

Table 3.2: Poisson case (Example 3.2).

	VSD	VSD^+	VSD^-	F	G
			Lasso		
True	2.760 (0.137)	0.000 (0.000)	2.760 (0.137)	0.704 (0.012)	0.740 (0.010)
ARM	1.730 (0.076)	0.182 (0.027)	1.548 (0.094)	0.828 (0.006)	0.845 (0.005)
AIC	1.644 (0.086)	0.115 (0.018)	1.529 (0.098)	0.838 (0.007)	0.853 (0.006)
BIC	2.580 (0.133)	0.003 (0.001)	2.577 (0.133)	0.727 (0.012)	0.759 (0.010)
ARM-p	2.264 (0.115)	0.042 (0.007)	2.222 (0.120)	0.765 (0.010)	0.791 (0.008)
AIC-p	2.212 (0.120)	0.023 (0.004)	2.189 (0.123)	0.771 (0.010)	0.797 (0.009)
BIC-p	2.685 (0.135)	0.001 (0.000)	2.684 (0.135)	0.713 (0.012)	0.748 (0.010)
			AdaLasso		
True	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	1.000 (0.000)	1.000 (0.000)
ARM	1.394 (0.041)	1.394 (0.041)	0.000 (0.000)	0.836 (0.004)	0.853 (0.004)
AIC	1.346 (0.055)	1.346 (0.055)	0.000 (0.000)	0.837 (0.006)	0.853 (0.005)
BIC	0.186 (0.023)	0.186 (0.023)	0.000 (0.000)	0.974 (0.003)	0.976 (0.003)
ARM-p	0.580 (0.027)	0.580 (0.027)	0.000 (0.000)	0.927 (0.003)	0.933 (0.003)
AIC-p	0.594 (0.045)	0.594 (0.045)	0.000 (0.000)	0.923 (0.005)	0.930 (0.005)
BIC-p	0.077 (0.015)	0.077 (0.015)	0.000 (0.000)	0.989 (0.002)	0.990 (0.002)
			MCP		
True	0.270 (0.069)	0.000 (0.000)	0.270 (0.069)	0.966 (0.008)	0.969 (0.008)
ARM	1.282 (0.037)	1.203 (0.046)	0.079 (0.022)	0.850 (0.004)	0.865 (0.004)
AIC	1.167 (0.050)	1.121 (0.055)	0.045 (0.014)	0.860 (0.006)	0.874 (0.005)
BIC	0.297 (0.048)	0.106 (0.010)	0.190 (0.051)	0.962 (0.006)	0.965 (0.005)
ARM-p	0.613 (0.034)	0.461 (0.023)	0.152 (0.041)	0.924 (0.004)	0.930 (0.003)
AIC-p	0.537 (0.034)	0.431 (0.031)	0.106 (0.030)	0.932 (0.004)	0.937 (0.004)
BIC-p	0.262 (0.058)	0.034 (0.004)	0.227 (0.059)	0.967 (0.007)	0.970 (0.006)
			SCAD		
True	0.320 (0.082)	0.000 (0.000)	0.320 (0.082)	0.961 (0.009)	0.965 (0.008)
ARM	1.288 (0.038)	1.181 (0.050)	0.107 (0.032)	0.850 (0.004)	0.865 (0.004)
AIC	1.158 (0.048)	1.092 (0.055)	0.066 (0.021)	0.862 (0.006)	0.875 (0.005)
BIC	0.334 (0.060)	0.100 (0.009)	0.234 (0.063)	0.959 (0.007)	0.962 (0.006)
ARM-p	0.632 (0.041)	0.446 (0.023)	0.186 (0.050)	0.922 (0.004)	0.929 (0.004)
AIC-p	0.550 (0.039)	0.412 (0.029)	0.138 (0.040)	0.931 (0.004)	0.937 (0.004)
BIC-p	0.307 (0.070)	0.032 (0.003)	0.275 (0.072)	0.963 (0.008)	0.966 (0.007)

Table 3.3: Poisson case (Example 3.3).

	VSD	VSD^+	VSD^-	F	G
			Lasso		
True	19.460 (1.115)	0.000 (0.000)	19.460 (1.115)	0.286 (0.013)	0.403 (0.011)
ARM	13.780 (0.531)	5.569 (0.479)	8.210 (0.764)	0.676 (0.008)	0.719 (0.006)
AIC	13.247 (0.721)	5.734 (0.764)	7.513 (0.840)	0.700 (0.013)	0.738 (0.011)
BIC	16.352 (1.015)	0.006 (0.004)	16.346 (1.016)	0.468 (0.014)	0.552 (0.011)
ARM-p	19.356 (1.112)	0.000 (0.000)	19.356 (1.112)	0.293 (0.013)	0.409 (0.011)
AIC-p	19.206 (1.110)	0.000 (0.000)	19.206 (1.110)	0.303 (0.014)	0.418 (0.011)
BIC-p	19.373 (1.112)	0.000 (0.000)	19.373 (1.112)	0.292 (0.013)	0.408 (0.011)
			AdaLasso		
True	0.080 (0.027)	0.000 (0.000)	0.080 (0.027)	0.989 (0.004)	0.989 (0.004)
ARM	16.739 (0.356)	16.739 (0.356)	0.000 (0.000)	0.354 (0.006)	0.458 (0.005)
AIC	17.601 (0.700)	17.601 (0.700)	0.000 (0.000)	0.298 (0.008)	0.415 (0.007)
BIC	3.040 (0.168)	3.040 (0.168)	0.000 (0.000)	0.708 (0.012)	0.745 (0.010)
ARM-p	0.139 (0.020)	0.081 (0.009)	0.057 (0.020)	0.980 (0.003)	0.982 (0.003)
AIC-p	0.228 (0.027)	0.201 (0.026)	0.027 (0.012)	0.969 (0.004)	0.971 (0.003)
BIC-p	0.106 (0.021)	0.056 (0.012)	0.049 (0.019)	0.985 (0.003)	0.986 (0.003)
			MCP		
True	0.250 (0.093)	0.000 (0.000)	0.250 (0.093)	0.973 (0.009)	0.976 (0.008)
ARM	16.580 (0.362)	16.574 (0.362)	0.005 (0.003)	0.364 (0.008)	0.467 (0.006)
AIC	17.432 (0.713)	17.431 (0.713)	0.000 (0.000)	0.310 (0.011)	0.424 (0.009)
BIC	2.974 (0.169)	2.922 (0.173)	0.052 (0.033)	0.716 (0.013)	0.752 (0.010)
ARM-p	0.312 (0.087)	0.083 (0.010)	0.229 (0.089)	0.964 (0.008)	0.968 (0.007)
AIC-p	0.393 (0.083)	0.198 (0.027)	0.195 (0.083)	0.953 (0.008)	0.957 (0.007)
BIC-p	0.293 (0.089)	0.065 (0.015)	0.228 (0.089)	0.967 (0.008)	0.970 (0.007)
			SCAD		
True	0.090 (0.032)	0.000 (0.000)	0.090 (0.032)	0.988 (0.004)	0.988 (0.004)
ARM	16.730 (0.355)	16.730 (0.355)	0.001 (0.000)	0.354 (0.006)	0.459 (0.005)
AIC	17.591 (0.699)	17.591 (0.699)	0.000 (0.000)	0.299 (0.008)	0.415 (0.007)
BIC	3.032 (0.171)	3.031 (0.171)	0.001 (0.001)	0.709 (0.013)	0.746 (0.010)
ARM-p	0.151 (0.025)	0.082 (0.009)	0.068 (0.026)	0.979 (0.003)	0.981 (0.003)
AIC-p	0.243 (0.030)	0.203 (0.026)	0.040 (0.018)	0.967 (0.004)	0.969 (0.004)
BIC-p	0.121 (0.027)	0.059 (0.012)	0.062 (0.026)	0.983 (0.004)	0.984 (0.003)

Table 3.4: Poisson case (Example 3.4).

	VSD	VSD^+	VSD^-	F	G
			Lasso		
True	6.830 (0.395)	0.000 (0.000)	6.830 (0.395)	0.514 (0.016)	0.589 (0.013)
ARM	4.383 (0.221)	0.756 (0.070)	3.627 (0.270)	0.741 (0.007)	0.773 (0.006)
AIC	3.779 (0.223)	0.569 (0.077)	3.209 (0.262)	0.784 (0.008)	0.808 (0.007)
BIC	5.975 (0.360)	0.004 (0.002)	5.971 (0.361)	0.598 (0.014)	0.656 (0.011)
ARM-p	6.483 (0.382)	0.001 (0.001)	6.482 (0.383)	0.550 (0.015)	0.617 (0.012)
AIC-p	6.336 (0.373)	0.001 (0.000)	6.335 (0.373)	0.563 (0.015)	0.628 (0.012)
BIC-p	6.725 (0.391)	0.000 (0.000)	6.725 (0.391)	0.525 (0.016)	0.598 (0.012)
			AdaLasso		
True	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	1.000 (0.000)	1.000 (0.000)
ARM	3.959 (0.109)	3.959 (0.109)	0.000 (0.000)	0.668 (0.007)	0.715 (0.006)
AIC	4.190 (0.149)	4.190 (0.149)	0.000 (0.000)	0.633 (0.009)	0.684 (0.007)
BIC	0.863 (0.060)	0.863 (0.060)	0.000 (0.000)	0.892 (0.007)	0.902 (0.006)
ARM-p	0.350 (0.022)	0.350 (0.022)	0.000 (0.000)	0.954 (0.003)	0.958 (0.002)
AIC-p	0.496 (0.045)	0.496 (0.045)	0.000 (0.000)	0.936 (0.005)	0.942 (0.005)
BIC-p	0.105 (0.013)	0.105 (0.013)	0.000 (0.000)	0.985 (0.002)	0.986 (0.002)
			MCP		
True	0.370 (0.138)	0.000 (0.000)	0.370 (0.138)	0.966 (0.011)	0.970 (0.010)
ARM	3.770 (0.101)	3.679 (0.114)	0.091 (0.042)	0.688 (0.007)	0.731 (0.006)
AIC	3.934 (0.150)	3.877 (0.159)	0.057 (0.031)	0.659 (0.010)	0.706 (0.008)
BIC	0.930 (0.091)	0.712 (0.054)	0.219 (0.091)	0.893 (0.007)	0.903 (0.006)
ARM-p	0.609 (0.117)	0.295 (0.021)	0.315 (0.122)	0.934 (0.009)	0.940 (0.008)
AIC-p	0.653 (0.103)	0.389 (0.037)	0.263 (0.105)	0.928 (0.008)	0.934 (0.007)
BIC-p	0.425 (0.130)	0.080 (0.011)	0.345 (0.132)	0.958 (0.010)	0.962 (0.009)
			SCAD		
True	0.220 (0.085)	0.000 (0.000)	0.220 (0.085)	0.976 (0.009)	0.979 (0.007)
ARM	3.830 (0.108)	3.784 (0.114)	0.046 (0.019)	0.682 (0.007)	0.726 (0.006)
AIC	4.004 (0.152)	3.987 (0.154)	0.017 (0.008)	0.652 (0.010)	0.700 (0.008)
BIC	0.880 (0.064)	0.762 (0.058)	0.119 (0.048)	0.893 (0.007)	0.903 (0.006)
ARM-p	0.480 (0.066)	0.305 (0.021)	0.175 (0.069)	0.942 (0.006)	0.947 (0.005)
AIC-p	0.571 (0.063)	0.424 (0.042)	0.148 (0.058)	0.930 (0.006)	0.936 (0.006)
BIC-p	0.283 (0.077)	0.084 (0.011)	0.199 (0.078)	0.967 (0.007)	0.970 (0.007)

Table 3.5: Poisson case (Example 3.5).

	VSD	VSD$^+$	VSD$^-$	F	G
			Lasso		
True	13.070 (0.849)	0.000 (0.000)	13.070 (0.849)	0.377 (0.016)	0.479 (0.013)
ARM	8.805 (0.381)	3.006 (0.290)	5.800 (0.517)	0.710 (0.006)	0.746 (0.005)
AIC	8.687 (0.540)	3.348 (0.549)	5.339 (0.554)	0.729 (0.011)	0.761 (0.009)
BIC	10.926 (0.789)	0.010 (0.007)	10.916 (0.790)	0.537 (0.015)	0.607 (0.012)
ARM-p	12.920 (0.848)	0.000 (0.000)	12.920 (0.848)	0.390 (0.016)	0.489 (0.013)
AIC-p	12.749 (0.844)	0.000 (0.000)	12.749 (0.844)	0.404 (0.016)	0.500 (0.013)
BIC-p	12.971 (0.849)	0.000 (0.000)	12.971 (0.849)	0.386 (0.016)	0.485 (0.013)
			AdaLasso		
True	0.050 (0.022)	0.000 (0.000)	0.050 (0.022)	0.993 (0.003)	0.993 (0.003)
ARM	10.227 (0.282)	10.227 (0.282)	0.001 (0.000)	0.464 (0.007)	0.549 (0.006)
AIC	11.028 (0.517)	11.028 (0.517)	0.000 (0.000)	0.408 (0.010)	0.504 (0.008)
BIC	2.114 (0.103)	2.114 (0.103)	0.000 (0.000)	0.771 (0.009)	0.798 (0.008)
ARM-p	0.171 (0.019)	0.136 (0.012)	0.035 (0.015)	0.976 (0.003)	0.978 (0.002)
AIC-p	0.300 (0.031)	0.286 (0.030)	0.015 (0.008)	0.960 (0.004)	0.962 (0.004)
BIC-p	0.109 (0.020)	0.079 (0.014)	0.030 (0.015)	0.985 (0.003)	0.986 (0.003)
			MCP		
True	0.150 (0.048)	0.000 (0.000)	0.150 (0.048)	0.980 (0.006)	0.982 (0.006)
ARM	10.137 (0.291)	10.131 (0.291)	0.005 (0.002)	0.473 (0.009)	0.556 (0.007)
AIC	10.929 (0.529)	10.929 (0.529)	0.000 (0.000)	0.418 (0.012)	0.512 (0.010)
BIC	2.065 (0.108)	2.039 (0.111)	0.025 (0.012)	0.778 (0.010)	0.803 (0.009)
ARM-p	0.253 (0.040)	0.127 (0.012)	0.126 (0.041)	0.966 (0.005)	0.969 (0.005)
AIC-p	0.368 (0.043)	0.270 (0.031)	0.099 (0.036)	0.952 (0.005)	0.955 (0.005)
BIC-p	0.205 (0.044)	0.077 (0.016)	0.128 (0.043)	0.973 (0.006)	0.975 (0.005)
			SCAD		
True	0.180 (0.066)	0.000 (0.000)	0.180 (0.066)	0.978 (0.007)	0.980 (0.006)
ARM	10.117 (0.286)	10.107 (0.288)	0.011 (0.008)	0.474 (0.009)	0.557 (0.007)
AIC	10.904 (0.524)	10.901 (0.524)	0.003 (0.003)	0.419 (0.012)	0.513 (0.010)
BIC	2.054 (0.108)	2.019 (0.111)	0.035 (0.026)	0.780 (0.010)	0.805 (0.009)
ARM-p	0.267 (0.055)	0.119 (0.011)	0.148 (0.057)	0.966 (0.006)	0.969 (0.005)
AIC-p	0.354 (0.050)	0.248 (0.028)	0.107 (0.047)	0.955 (0.005)	0.958 (0.005)
BIC-p	0.213 (0.058)	0.066 (0.013)	0.147 (0.058)	0.974 (0.006)	0.976 (0.006)

Example 3.6

$n = 200$, $p = 8$, and $\boldsymbol{\beta}^* = (3, 1.5, 0, 0, -2, \mathbf{0}_3)^\intercal$. The components of $\mathbf{x}$ are independent and identically distributed (i.i.d.) from the uniform distribution on $(0, 1)$. □

Example 3.7

Same as Example 3.6, except $n = 1000$. □

Example 3.8

$n = 200$, $p = 500$, and $\boldsymbol{\beta}^* = (3, 1.5, 0, 0, -2, \mathbf{0}_{495})^\intercal$. The components of $\mathbf{x}$ are i.i.d. from Unif$(0, 1)$. □

Example 3.9

$n = 200$, $p = 30$, and $\boldsymbol{\beta}^* = (3, 1.5, 0, 0, -2, \mathbf{0}_{25})^\intercal$. The components of $\mathbf{x}$ are marginally distributed as Unif$(0, 1)$ but has correlation matrix $\boldsymbol{\rho} = (0.3^{|j-j'|})_{p\times p}$. □

Example 3.10

$n = 200$, $p = 200$, and $\boldsymbol{\beta}^* = (3, 1.5, 0, 0, -2, \mathbf{0}_{195})^\intercal$. The components of $\mathbf{x}$ are marginally distributed as Unif$(0, 1)$ but have correlation matrix $\boldsymbol{\rho} = (0.6^{|j-j'|})_{p\times p}$. □

In all of the above settings, we evaluate the variable selection performance of four optimally tuned penalized Tweedie regression models using the lasso, adaptive lasso, SCAD and MCP penalties, where the tuning is done by five-fold cross-validation. The candidate models are obtained from the solution paths of the lasso, adaptive lasso, SCAD and MCP penalized Tweedie models at sequences of tuning parameters. The nonconvex (i.e., SCAD and MCP) penalized Tweedie regression models are fitted using the two-step LLA algorithm described in Section 3.4.2.

As in the Poisson cases, we report the estimates and true values of $\text{VSD}(\mathcal{A}^0)$, $\text{VSD}^+(\mathcal{A}^0)$,

$\text{VSD}^{-}(\mathcal{A}^0)$, $F(\mathcal{A}^0)$ and $G(\mathcal{A}^0)$ for each model $\mathcal{A}^0 \in \{\mathcal{A}^{\text{lasso}}, \mathcal{A}^{\text{AdaLasso}}, \mathcal{A}^{\text{MCP}}, \mathcal{A}^{\text{SCAD}}\}$. Six different weighting methods are used: ARM, AIC, BIC, ARM-p, AIC-p, and BIC-p, where $\psi = 1$ is used for those using prior adjustment. The number of observations used in the training part to compute the ARM weights is roughly half of the sample size, i.e., $\lfloor n/2 \rfloor$, and the corresponding repetition time is one hundred.

All simulation cases are repeated one hundred times, and the corresponding values are computed and averaged. We summarize the results in Tables 3.6 – 3.10 for Examples 3.6 – 3.10. The standard errors are reported in the parentheses. Similar conclusions can be drawn from these tables. Specifically, our weighting methods typically give accurate estimates of the true VSD, F- and G-measures. The estimates with prior adjustments again are more accurate than those without prior adjustments, with BIC-p being the best. In terms of the variable selection performance, we can see that the adaptive lasso, SCAD and MCP are generally very good in terms of variable selection accuracy and they are better than the lasso, probably due to the fact that the lasso tends to over-select the variables under our settings. Overall, the performance of the models becomes better when the sample size increases and becomes worse when the dimension increases or when the correlations among the features increase.

Table 3.6: Tweedie case (Example 3.6).

	VSD	VSD^+	VSD^-	F	G
			Lasso		
True	1.650 (0.105)	0.000 (0.000)	1.650 (0.105)	0.799 (0.011)	0.819 (0.010)
ARM	0.904 (0.055)	0.057 (0.014)	0.846 (0.058)	0.895 (0.005)	0.904 (0.005)
AIC	0.806 (0.052)	0.052 (0.013)	0.754 (0.054)	0.907 (0.005)	0.914 (0.005)
BIC	1.363 (0.089)	0.007 (0.003)	1.357 (0.090)	0.837 (0.009)	0.852 (0.008)
ARM-p	1.266 (0.080)	0.019 (0.005)	1.247 (0.082)	0.849 (0.008)	0.863 (0.007)
AIC-p	1.203 (0.078)	0.014 (0.004)	1.189 (0.079)	0.857 (0.008)	0.870 (0.007)
BIC-p	1.536 (0.098)	0.002 (0.001)	1.534 (0.098)	0.815 (0.010)	0.832 (0.009)
			AdaLasso		
True	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	1.000 (0.000)	1.000 (0.000)
ARM	0.863 (0.049)	0.862 (0.049)	0.001 (0.001)	0.889 (0.006)	0.899 (0.005)
AIC	0.948 (0.060)	0.948 (0.060)	0.000 (0.000)	0.878 (0.007)	0.889 (0.006)
BIC	0.300 (0.027)	0.300 (0.027)	0.000 (0.000)	0.959 (0.004)	0.962 (0.003)
ARM-p	0.429 (0.027)	0.426 (0.027)	0.003 (0.001)	0.943 (0.003)	0.947 (0.003)
AIC-p	0.475 (0.041)	0.475 (0.041)	0.000 (0.000)	0.937 (0.005)	0.942 (0.005)
BIC-p	0.118 (0.015)	0.118 (0.015)	0.000 (0.000)	0.984 (0.002)	0.985 (0.002)
			MCP		
True	0.230 (0.074)	0.000 (0.000)	0.230 (0.074)	0.973 (0.008)	0.976 (0.007)
ARM	0.825 (0.048)	0.728 (0.050)	0.097 (0.033)	0.895 (0.005)	0.905 (0.005)
AIC	0.853 (0.054)	0.785 (0.058)	0.067 (0.025)	0.891 (0.006)	0.901 (0.006)
BIC	0.374 (0.049)	0.222 (0.020)	0.152 (0.052)	0.952 (0.005)	0.956 (0.005)
ARM-p	0.499 (0.046)	0.346 (0.025)	0.154 (0.050)	0.936 (0.005)	0.942 (0.004)
AIC-p	0.483 (0.043)	0.364 (0.032)	0.119 (0.041)	0.938 (0.005)	0.943 (0.004)
BIC-p	0.267 (0.060)	0.077 (0.008)	0.189 (0.062)	0.967 (0.007)	0.970 (0.006)
			SCAD		
True	0.250 (0.081)	0.000 (0.000)	0.250 (0.081)	0.971 (0.009)	0.974 (0.008)
ARM	0.839 (0.050)	0.725 (0.050)	0.114 (0.040)	0.894 (0.006)	0.904 (0.005)
AIC	0.862 (0.055)	0.780 (0.057)	0.082 (0.034)	0.891 (0.006)	0.900 (0.006)
BIC	0.396 (0.059)	0.223 (0.021)	0.173 (0.062)	0.950 (0.006)	0.954 (0.005)
ARM-p	0.514 (0.053)	0.343 (0.025)	0.171 (0.058)	0.935 (0.005)	0.941 (0.005)
AIC-p	0.502 (0.052)	0.364 (0.033)	0.138 (0.051)	0.937 (0.005)	0.942 (0.005)
BIC-p	0.290 (0.069)	0.079 (0.009)	0.211 (0.071)	0.965 (0.007)	0.968 (0.006)

Table 3.7: Tweedie case (Example 3.7).

	VSD	VSD^+	VSD^-	F	G
			Lasso		
True	0.200 (0.047)	0.000 (0.000)	0.200 (0.047)	0.972 (0.006)	0.974 (0.006)
ARM	0.104 (0.024)	0.040 (0.019)	0.064 (0.015)	0.986 (0.003)	0.987 (0.003)
AIC	0.078 (0.026)	0.051 (0.025)	0.027 (0.007)	0.990 (0.003)	0.990 (0.003)
BIC	0.136 (0.033)	0.013 (0.006)	0.123 (0.032)	0.982 (0.004)	0.983 (0.004)
ARM-p	0.131 (0.028)	0.024 (0.011)	0.107 (0.026)	0.982 (0.004)	0.983 (0.003)
AIC-p	0.100 (0.024)	0.037 (0.018)	0.063 (0.016)	0.987 (0.003)	0.987 (0.003)
BIC-p	0.168 (0.040)	0.004 (0.002)	0.164 (0.040)	0.977 (0.005)	0.979 (0.005)
			AdaLasso		
True	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	1.000 (0.000)	1.000 (0.000)
ARM	0.176 (0.037)	0.176 (0.037)	0.000 (0.000)	0.976 (0.005)	0.977 (0.005)
AIC	0.224 (0.047)	0.224 (0.047)	0.000 (0.000)	0.969 (0.006)	0.972 (0.006)
BIC	0.090 (0.019)	0.090 (0.019)	0.000 (0.000)	0.987 (0.003)	0.988 (0.003)
ARM-p	0.117 (0.025)	0.117 (0.025)	0.000 (0.000)	0.984 (0.003)	0.985 (0.003)
AIC-p	0.174 (0.037)	0.174 (0.037)	0.000 (0.000)	0.976 (0.005)	0.978 (0.005)
BIC-p	0.040 (0.010)	0.040 (0.010)	0.000 (0.000)	0.994 (0.001)	0.995 (0.001)
			MCP		
True	0.010 (0.010)	0.000 (0.000)	0.010 (0.010)	0.999 (0.001)	0.999 (0.001)
ARM	0.171 (0.036)	0.168 (0.036)	0.003 (0.003)	0.976 (0.005)	0.978 (0.004)
AIC	0.215 (0.047)	0.214 (0.047)	0.000 (0.000)	0.971 (0.006)	0.973 (0.006)
BIC	0.086 (0.018)	0.083 (0.018)	0.003 (0.003)	0.988 (0.003)	0.989 (0.002)
ARM-p	0.115 (0.024)	0.111 (0.024)	0.004 (0.004)	0.984 (0.003)	0.985 (0.003)
AIC-p	0.166 (0.037)	0.165 (0.037)	0.001 (0.001)	0.977 (0.005)	0.979 (0.004)
BIC-p	0.042 (0.011)	0.036 (0.009)	0.006 (0.006)	0.994 (0.002)	0.994 (0.001)
			SCAD		
True	0.090 (0.040)	0.000 (0.000)	0.090 (0.040)	0.988 (0.005)	0.989 (0.004)
ARM	0.139 (0.028)	0.113 (0.027)	0.027 (0.012)	0.981 (0.004)	0.982 (0.004)
AIC	0.149 (0.033)	0.141 (0.033)	0.007 (0.004)	0.979 (0.005)	0.981 (0.004)
BIC	0.116 (0.032)	0.058 (0.015)	0.058 (0.030)	0.985 (0.004)	0.986 (0.004)
ARM-p	0.121 (0.027)	0.074 (0.018)	0.047 (0.022)	0.984 (0.003)	0.985 (0.003)
AIC-p	0.127 (0.027)	0.106 (0.025)	0.022 (0.011)	0.982 (0.004)	0.984 (0.003)
BIC-p	0.102 (0.036)	0.026 (0.008)	0.076 (0.036)	0.987 (0.004)	0.988 (0.004)

Table 3.8: Tweedie case (Example 3.8).

	VSD	VSD$^+$	VSD$^-$	F	G
			Lasso		
True	14.280 (0.999)	0.000 (0.000)	14.280 (0.999)	0.367 (0.017)	0.470 (0.014)
ARM	21.742 (0.307)	17.293 (0.549)	4.449 (0.498)	0.543 (0.011)	0.589 (0.008)
AIC	46.901 (1.078)	44.490 (1.229)	2.412 (0.292)	0.374 (0.015)	0.459 (0.012)
BIC	11.312 (0.631)	3.652 (0.298)	7.660 (0.746)	0.643 (0.009)	0.671 (0.008)
ARM-p	14.199 (0.995)	0.003 (0.001)	14.197 (0.995)	0.373 (0.017)	0.475 (0.014)
AIC-p	13.817 (0.977)	0.017 (0.006)	13.800 (0.979)	0.402 (0.017)	0.498 (0.013)
BIC-p	14.158 (0.994)	0.002 (0.001)	14.156 (0.994)	0.376 (0.017)	0.477 (0.014)
			AdaLasso		
True	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	1.000 (0.000)	1.000 (0.000)
ARM	27.124 (0.286)	27.124 (0.286)	0.000 (0.000)	0.215 (0.002)	0.341 (0.002)
AIC	56.358 (0.975)	56.358 (0.975)	0.000 (0.000)	0.099 (0.002)	0.227 (0.002)
BIC	10.272 (0.403)	10.272 (0.403)	0.000 (0.000)	0.410 (0.010)	0.506 (0.008)
ARM-p	0.114 (0.009)	0.100 (0.009)	0.014 (0.003)	0.983 (0.001)	0.984 (0.001)
AIC-p	0.497 (0.055)	0.497 (0.055)	0.000 (0.000)	0.936 (0.006)	0.942 (0.006)
BIC-p	0.126 (0.018)	0.126 (0.018)	0.000 (0.000)	0.982 (0.002)	0.984 (0.002)
			MCP		
True	0.080 (0.031)	0.000 (0.000)	0.080 (0.031)	0.989 (0.004)	0.990 (0.004)
ARM	27.044 (0.283)	27.044 (0.283)	0.000 (0.000)	0.219 (0.003)	0.345 (0.002)
AIC	56.278 (0.976)	56.278 (0.976)	0.000 (0.000)	0.101 (0.002)	0.230 (0.002)
BIC	10.192 (0.403)	10.192 (0.403)	0.000 (0.000)	0.417 (0.011)	0.512 (0.009)
ARM-p	0.162 (0.025)	0.084 (0.007)	0.078 (0.026)	0.977 (0.003)	0.978 (0.003)
AIC-p	0.467 (0.053)	0.442 (0.053)	0.025 (0.012)	0.941 (0.006)	0.946 (0.005)
BIC-p	0.145 (0.024)	0.095 (0.013)	0.049 (0.022)	0.980 (0.003)	0.981 (0.003)
			SCAD		
True	0.180 (0.094)	0.000 (0.000)	0.180 (0.094)	0.982 (0.008)	0.985 (0.007)
ARM	26.946 (0.290)	26.945 (0.290)	0.001 (0.001)	0.224 (0.005)	0.349 (0.004)
AIC	56.178 (0.986)	56.178 (0.986)	0.000 (0.000)	0.104 (0.003)	0.233 (0.003)
BIC	10.105 (0.402)	10.099 (0.403)	0.006 (0.005)	0.423 (0.012)	0.517 (0.009)
ARM-p	0.260 (0.089)	0.083 (0.007)	0.177 (0.090)	0.970 (0.007)	0.973 (0.006)
AIC-p	0.549 (0.088)	0.433 (0.053)	0.116 (0.077)	0.936 (0.008)	0.942 (0.007)
BIC-p	0.239 (0.088)	0.092 (0.013)	0.146 (0.089)	0.974 (0.007)	0.977 (0.006)

Table 3.9: Tweedie case (Example 3.9).

	VSD	VSD^+	VSD^-	F	G
			Lasso		
True	5.240 (0.274)	0.000 (0.000)	5.240 (0.274)	0.567 (0.015)	0.631 (0.012)
ARM	3.067 (0.130)	0.814 (0.087)	2.253 (0.162)	0.791 (0.005)	0.812 (0.005)
AIC	2.589 (0.126)	1.118 (0.125)	1.471 (0.125)	0.836 (0.007)	0.848 (0.006)
BIC	4.156 (0.238)	0.049 (0.010)	4.108 (0.244)	0.683 (0.012)	0.724 (0.010)
ARM-p	4.883 (0.260)	0.014 (0.004)	4.869 (0.262)	0.607 (0.013)	0.663 (0.011)
AIC-p	4.555 (0.249)	0.020 (0.005)	4.535 (0.251)	0.641 (0.013)	0.690 (0.011)
BIC-p	5.065 (0.266)	0.002 (0.001)	5.064 (0.266)	0.587 (0.014)	0.647 (0.011)
			AdaLasso		
True	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	1.000 (0.000)	1.000 (0.000)
ARM	3.803 (0.103)	3.802 (0.103)	0.001 (0.000)	0.656 (0.006)	0.704 (0.005)
AIC	4.886 (0.167)	4.886 (0.167)	0.000 (0.000)	0.582 (0.009)	0.642 (0.007)
BIC	1.181 (0.062)	1.181 (0.062)	0.000 (0.000)	0.856 (0.007)	0.869 (0.006)
ARM-p	0.404 (0.021)	0.395 (0.021)	0.009 (0.001)	0.947 (0.003)	0.951 (0.002)
AIC-p	0.725 (0.056)	0.725 (0.056)	0.000 (0.000)	0.909 (0.006)	0.917 (0.006)
BIC-p	0.178 (0.022)	0.178 (0.022)	0.000 (0.000)	0.975 (0.003)	0.977 (0.003)
			MCP		
True	0.240 (0.075)	0.000 (0.000)	0.240 (0.075)	0.971 (0.008)	0.974 (0.007)
ARM	3.629 (0.108)	3.595 (0.113)	0.034 (0.014)	0.676 (0.008)	0.720 (0.006)
AIC	4.653 (0.176)	4.650 (0.176)	0.003 (0.002)	0.607 (0.011)	0.663 (0.009)
BIC	1.085 (0.058)	1.013 (0.058)	0.072 (0.035)	0.870 (0.006)	0.882 (0.005)
ARM-p	0.498 (0.051)	0.322 (0.018)	0.177 (0.056)	0.937 (0.005)	0.943 (0.004)
AIC-p	0.686 (0.055)	0.585 (0.047)	0.101 (0.042)	0.916 (0.006)	0.924 (0.005)
BIC-p	0.293 (0.057)	0.115 (0.014)	0.177 (0.059)	0.964 (0.006)	0.967 (0.005)
			SCAD		
True	0.400 (0.137)	0.000 (0.000)	0.400 (0.137)	0.960 (0.011)	0.965 (0.010)
ARM	3.584 (0.105)	3.492 (0.118)	0.092 (0.048)	0.681 (0.008)	0.724 (0.006)
AIC	4.526 (0.173)	4.506 (0.177)	0.020 (0.013)	0.616 (0.012)	0.671 (0.009)
BIC	1.140 (0.091)	0.960 (0.054)	0.179 (0.093)	0.869 (0.007)	0.881 (0.006)
ARM-p	0.627 (0.115)	0.306 (0.017)	0.321 (0.121)	0.930 (0.008)	0.936 (0.007)
AIC-p	0.759 (0.098)	0.542 (0.040)	0.217 (0.101)	0.914 (0.007)	0.922 (0.006)
BIC-p	0.436 (0.125)	0.107 (0.013)	0.329 (0.127)	0.955 (0.009)	0.959 (0.008)

Table 3.10: Tweedie case (Example 3.10).

	VSD	VSD^+	VSD^-	F	G
			Lasso		
True	11.230 (0.687)	0.000 (0.000)	11.230 (0.687)	0.407 (0.017)	0.503 (0.013)
ARM	9.982 (0.263)	6.152 (0.303)	3.830 (0.373)	0.675 (0.007)	0.704 (0.006)
AIC	15.203 (0.543)	12.840 (0.636)	2.363 (0.282)	0.599 (0.015)	0.638 (0.012)
BIC	8.117 (0.543)	0.447 (0.074)	7.670 (0.574)	0.648 (0.013)	0.691 (0.011)
ARM-p	11.148 (0.681)	0.002 (0.001)	11.146 (0.681)	0.412 (0.016)	0.507 (0.013)
AIC-p	10.714 (0.656)	0.007 (0.003)	10.707 (0.657)	0.449 (0.016)	0.537 (0.013)
BIC-p	11.103 (0.679)	0.001 (0.001)	11.102 (0.679)	0.417 (0.016)	0.511 (0.013)
			AdaLasso		
True	0.020 (0.014)	0.020 (0.014)	0.000 (0.000)	0.996 (0.003)	0.996 (0.003)
ARM	13.573 (0.221)	13.572 (0.221)	0.000 (0.000)	0.355 (0.004)	0.462 (0.003)
AIC	21.727 (0.484)	21.727 (0.484)	0.000 (0.000)	0.226 (0.004)	0.355 (0.004)
BIC	4.027 (0.176)	4.027 (0.176)	0.000 (0.000)	0.634 (0.010)	0.685 (0.008)
ARM-p	0.195 (0.014)	0.151 (0.014)	0.044 (0.004)	0.970 (0.002)	0.972 (0.002)
AIC-p	0.551 (0.067)	0.551 (0.067)	0.000 (0.000)	0.929 (0.007)	0.936 (0.006)
BIC-p	0.149 (0.026)	0.149 (0.026)	0.000 (0.000)	0.978 (0.004)	0.980 (0.003)
			MCP		
True	0.320 (0.106)	0.000 (0.000)	0.320 (0.106)	0.964 (0.009)	0.968 (0.008)
ARM	13.252 (0.234)	13.242 (0.236)	0.010 (0.007)	0.380 (0.007)	0.482 (0.006)
AIC	21.387 (0.504)	21.387 (0.504)	0.000 (0.000)	0.247 (0.007)	0.373 (0.006)
BIC	3.736 (0.164)	3.711 (0.166)	0.025 (0.021)	0.667 (0.011)	0.711 (0.009)
ARM-p	0.437 (0.097)	0.102 (0.007)	0.335 (0.099)	0.945 (0.008)	0.950 (0.007)
AIC-p	0.594 (0.081)	0.403 (0.047)	0.192 (0.075)	0.930 (0.006)	0.936 (0.005)
BIC-p	0.344 (0.091)	0.077 (0.009)	0.268 (0.092)	0.960 (0.008)	0.964 (0.007)
			SCAD		
True	0.130 (0.049)	0.000 (0.000)	0.130 (0.049)	0.983 (0.006)	0.985 (0.005)
ARM	13.427 (0.221)	13.425 (0.221)	0.003 (0.001)	0.367 (0.005)	0.472 (0.004)
AIC	21.577 (0.489)	21.577 (0.489)	0.000 (0.000)	0.235 (0.005)	0.363 (0.004)
BIC	3.882 (0.169)	3.879 (0.169)	0.003 (0.001)	0.651 (0.010)	0.699 (0.008)
ARM-p	0.275 (0.039)	0.116 (0.007)	0.159 (0.041)	0.960 (0.005)	0.963 (0.004)
AIC-p	0.503 (0.049)	0.452 (0.049)	0.051 (0.019)	0.937 (0.005)	0.942 (0.004)
BIC-p	0.182 (0.033)	0.091 (0.010)	0.092 (0.033)	0.976 (0.004)	0.978 (0.004)

3.6 Conclusion

We have generalized the idea of the variable selection deviations (Nan and Yang, 2014) and F- and G-measures (Yu et al., 2022) to more members in the generalized linear model family. Specifically, we have studied the selection performance of a given variable selection technique for the Poisson and Tweedie (compound Poisson-gamma) models. We have proposed both information-criterion-based and ARM-based weighting methods to signify the relative important of multiple candidate models. Intuitively, the weighting methods make sure that heavier weights are put on models that are closer to the true model and if the set of candidate models is rich enough to include models that are close to the true model, then the weighted diagnostic measures between the candidate models and the model to be evaluated can accurately estimate their counterparts between the true model and the model to be evaluated. Our extensive numerical studies have shown that the proposed weighting methods for calculating the deviation measures are very effective in recovering the true deviations of commonly used variable selection methods with a sensible and feasible choice of the set of candidate models. Though the current chapter focuses on the Poisson and Tweedie models, the method works for essentially any member of the generalized linear model family. In particular, similar procedures can be easily adapted to the gamma model. We have implemented the general procedure in an R package call **PAVI**, detailed in Chapter 4 of this dissertation, that works for any member of the generalized linear model family out of the box.

Chapter 4

PAVI: An R Package for General Variable Selection Diagnostics

The R package **PAVI**, an abbreviation for performance assessment of variable identification, has one major fitting function `glmvsd()` for computing the variable selection diagnostic measures mentioned in previous chapters such as VSD, VSD^+, VSD^-, precision, recall, F- and G- measures, and various helper functions for carrying out tasks such as candidate model fitting and weight assignment for the candidate models, using either adaptive regression by mixing (ARM, Yang, 2001) or an information criterion such as AIC or BIC. Table 4.1 lists all the main functions provided by **PAVI**.

Table 4.1: An overview of functions in **PAVI**.

Function	Description
`glmvsd()`	Main function of the package. Calculates the variable selection deviations to measure the uncertainty of various model selection methods. Returns an object of S3 class "`glmvsd`", a list which consists of the variable selection deviation values (VSD, VSD^+, VSD^-, precision, recall, F- and G-measures), the weights for the candidate models, and the actual candidate models used.
`IC()`	Returns the weights for a given set of candidate models based on their information criteria (such as AIC and BIC) computed using the training data.
`ARM()`	Returns the weights for a given set of candidate models based on their predictive performance computed using the adaptive regression by mixing (ARM, Yang, 2001) procedure which relies on data splitting.
`glm.ridge()`	An auxiliary function to run the generalized linear model with ridge penalty. Mainly handles scenarios where the Fisher scoring algorithm of the `glm()` function fails to converge. Has the same interface as the `glm()` function.
`tweedie()`	The family function for the Tweedie models. Defines the variance, link and deviance functions required for running the generalized linear model with the `glm()` function.

4.1 The main fitting function

The `glmvsd()` function calculates the variable selection deviation measures to quantify the variable selection uncertainty of many feature selection models based on the methods discussed in Chapters 2 and 3. It has the following arguments

```
glmvsd(
  x,
  y,
  family = c("gaussian", "binomial", "poisson", "tweedie"),
  model_check,
  method = c("union", "customize"),
  candidate_models,
  weight_type = c("BIC", "AIC", "ARM"),
  prior = TRUE,
  psi = 1,
  arm_ntrain = ceiling(n/2),
  arm_nrep = 100L,
  arm_maxsize = arm_ntrain - 2L,
  arm_mccores = 2L,
  ic_maxsize = n - 2L,
  reduce_bias = FALSE,
  dispersion = NULL,
  lambda = 1e-04,
  control = list(...),
  ...
)
```

This function takes four required arguments: an $n \times p$ design matrix x (without intercept column), an $n \times 1$ response vector y, a vector (resp. matrix) that represents the variable

selection model (resp. models) to be assessed `model_check`, and a matrix representing the candidate models `candidate_models`. The model(s) to be assessed and the candidate models should both be able to be converted into matrices, each of whose rows consists of zeros or ones that indicate whether the variables are excluded or included in the model. The design matrix is expected to contain only numeric features with the assumption that all categorical variables have already been converted into numeric values (e.g., dummy variables) in advance.

The `family` argument specifies the error distribution and link function to be used in the model. This can be a character string giving the name of the family function, a family function itself, or the result of a call to a family function. Though any family function available in the **stats** package can be used, we have highlighted four main GLM families of interest: "`gaussian`", "`binomial`", "`poisson`" and "`tweedie`", where "`tweedie`" is an additional family defined in the package to handle the Tweedie model.

The `method` argument specifies which method to use for compiling the matrix of candidate models. If `method` is set to `"union"`, the function automatically obtains the candidate models as a union of the solution paths of the GLM under the Lasso, SCAD, and MCP regularization, which currently works only for the Gaussian and binomial regressions. In this case, the argument `candidate_models` is ignored. The user can also set `method` to `"customize"` in which case a customized set of candidate models can be specified in the argument `candidate_models`.

The `weight_type` argument determines the method for computing the weights of the candidate models which are finally used for calculating the VSD measures. Three options "`ARM`", "`AIC`" and "`BIC`" are available with "`BIC`" being the default. The arguments `arm_ntrain`, `arm_nrep`, `arm_maxsize` and `arm_mccores` are all related to the

ARM weighting method, and determine the split size, the number of splits, the maximum model size and the number of cores to use, respectively. Since data partitioning is needed in the ARM method, the training size is often smaller than the sample size which restricts the size of the unpenalized model to be less than or equal to the training size. The `ic_maxsize` argument determines the maximum model size for the AIC or BIC weighting method.

The `prior` argument is often used together with the `psi` argument. The details of the prior weights are given in Section 3.3. By default, the uniform prior (`prior = FALSE`) is used. Our extensive empirical experiments show that usually a nonuniform prior (with $\psi = 1$) works very well and is better than the uniform prior. This can be achieved by setting `prior = TRUE` and `psi = 1`.

The `dispersion` argument is used in the Gaussian and Tweedie cases when the dispersion parameter is known *apriori*. By leaving this argument unspecified (`dispersion = NULL`), the dispersion parameter can be estimated via moment method using the deviance or via the maximum likelihood estimation.

The `control` argument can be used to pass a list of hyper-parameters from `glmvsd`'s general interface to each individual candidate model fitting procedure for controlling their fitting processes. For example, if the `glm()` function is used for fitting the candidate models, then the list of parameters in the `control` argument will be passed to `glm.control()` to control the iteratively reweighted least squares (IRLS) fitting process. Similarly, those parameters will be passed to `glmnet.control` (resp. `brglm.control`) if the `glmnet` (resp. `brglm`) function is used to fit the candidate models. Anything that belongs to the dot-dot-dot (...) argument will be used to form the `control` argument if it is not directly supplied.

Finally, the `reduce_bias` and `lambda` arguments are there for dealing with scenarios

when the algorithm of the `glm()` function fails to converge for fitting some of the candidate models. Recall that in the ARM weighting method, data partitioning is repeated many times to the data for stabilizing the weights. We observe in many of our empirical studies convergence issues with the logistic and Poisson regressions. For the logistic regression, it is well known that complete separation or quasi-complete separation of the classes results in nonexistence of the maximum likelihood estimates (Albert and Anderson, 1984; Santner and Duffy, 1986). For the Poisson regression, it is pointed out by Silva and Tenreyro (2010) that the maximum likelihood estimate does not exist either when perfect collinearity happens for the subsample with positive response values. Near-collinearity may cause convergence issues as well due to numerical errors in practice (Silva and Tenreyro, 2011). Two main strategies are implemented here for handling these non-convergent cases. First, for the logistic regression, we use the `brglm()` function from the **brglm** package to fit the bias-reduced logistic regression, which implements the bias-reduction method developed in Firth (1993) to remove the leading $\mathcal{O}(n^{-1})$ term from the asymptotic expansion of the bias of the maximum likelihood estimate. The limitation of this function is that it mainly works for the logistic regression and may not be able to handle all pathological cases. This is what motivated us to develop the second strategy for solving the convergence issues in the Poisson regression and beyond. Specifically, we have implemented a function called `glm.ridge()` that fits the ridge penalized generalized linear models. Adding the ridge penalty to the likelihood function can not only solve the collinearity problem seen in a lot of partitioned data, but also make sure the optimization algorithm converge from a numerical optimization point of view. Our implementation relies on the `glmnet()` function from the **glmnet** package under the hood, but provides a consistent user interface in much the same way as the `glm()` function. For numerical stability of the VSD procedure, it

is recommended that this `glm.ridge()` function is called with small `lambda` value (e.g., `lambda = 1e-4`) instead of the `glm()` function when fitting the candidate models with moderate to large dimensions.

4.2 Function for information-criterion-based weighting

The `IC()` function is called by the main function `glmvsd()` for calculating the adaptive weights of the candidate models based on an information criterion such as AIC or BIC. The exact formula is given in (2.9). The function can be used on its own if one is interested to compute the weights for a given set of candidate models. The arguments of the `IC()` function are shown below

```
IC(x,
   y,
   candidate_models,
   psi = 1.0,
   type = c("BIC", "AIC"),
   prior = TRUE,
   family = gaussian(),
   reduce_bias = FALSE,
   dispersion = NULL,
   control = list(...),
   lambda = 1e-6,
   ...)
```

Most of the arguments are similar to those of the `glmvsd()` function. Here we give some details of the `reduce_bias` argument. By default, `reduce_bias` is set to `FALSE` so that a regular generalized linear model is fit to only the active covariates in each candidate model. When bias reduction is deemed necessary (`reduce_bias = TRUE`), we separately handle the binomial (logistic) family and the other GLM families. For the logistic regression,

the following implementation is given using the `brglm()` function from the **brglm** package

```
getIC <- function(k) {
  df <- data.frame(y = y, x = x[, candidate_models[k, ] == 1])
  glmfit <- if (reduce_bias) {
              brglm(y ~ ., data = df, family = family)
            } else {
              glm(y ~ ., data = df, family = family)
            }
  icc <- if (type == "BIC") log(n) else 2
  ic <- extractAIC(glmfit, k = icc)[2L]
  return(ic)
}
```

For the other GLM families, we solve the convergence issue with the ridge regression for generalized linear models using our own implementation `glm.ridge()`

```
getIC <- function(k) {
  df <- data.frame(y = y, x = x[, candidate_models[k, ] == 1])
  glmfit <- if (reduce_bias) {
              glm.ridge(y ~ ., data = df, family = family, lambda = lambda)
            }
            else {
              glm(y ~ ., data = df, family = family)
            }
  icc <- if (type == "BIC") log(n) else 2
  ic <- extractAIC(glmfit, k = icc)[2L]
  return(ic)
}
```

Note that we have created a method function `extractAIC.glm.ridge()` for the model class `glm.ridge` to calculate the information criterion from a ridge GLM model.

4.3 Function for ARM-based weighting

The adaptive regression by mixing (ARM) methodology often requires many repetitions of data splitting for stable calculation of the weights. To accelerate the computation, a parallel scheme is implemented in the **PAVI** package to fully utilize the resources of current computers. This is realized using the `mclapply()` function in the **parallel** package to run the model fitting using multiple cores. The ARM procedure also supports bias reduced estimates using either `brglm()` or `glm.ridge()`. For example, the following implementation works for the Poisson family

```
calc_wts <- function(index) {
  rows <- sample(n, ntrain, replace = FALSE)
  loglik <- function(k) {
    df <- data.frame(y = y, x = x[, candidate_models[k, ] == 1])
    if (reduce_bias) {
      glmfit <- glm.ridge(y ~ ., data = df, family = family,
                          lambda = lambda, subset = rows)
      lamk <- predict.glm.ridge(glmfit,
                                newdata = df[-rows, , drop = FALSE],
                                type = "response")
    } else {
      glmfit <- glm(y ~ ., data = df, family = family, subset = rows)
      lamk <- predict.glm(glmfit,
                          newdata = df[-rows, , drop = FALSE],
                          type = "response")
    }
```

```
    ## sum(log(lamk) * (y[-rows]) - lamk - lfactorial(y[-rows]))
    sum(dpois(y[-rows], lambda = lamk, log = TRUE))
  }
  sapply(seq(nmodels), loglik)
}

## RUN ARM REPITITIONS IN PARALLEL
logwts <- matrix(unlist(mclapply(seq(nrep), calc_wts, mc.cores = mc_cores)),
                 nrow = nrep, ncol = nmodels, byrow = TRUE)
```

The `calc_wts()` function receives a single argument `index`, which is the repetition index, and returns the unnormalized log weights of the candidate models. This function is then used inside the `mclapply()` function to do parallel computation for the repetitive data splitting. Inside the `calc_wts()` function, we define a `loglik()` function to calculate the log-likelihood of the candidate models using the ARM method. Note that we have also defined a prediction method `predict.glm.ridge()` for the model class `glm.ridge`, which is in charge of making prediction from a fitted ridge GLM.

4.4 Function for fitting ridge generalized linear models

The ridge GLM function `glm.ridge()` calls the `glmnet()` function from the **glmnet** package in the background, but has a very similar interface to the commonly used `glm()` function which `glmnet()` is lacking.

```
glm.ridge(formula, family = gaussian, data, weights, subset, na.action,
          start = NULL, etastart, mustart, offset, control = list(...),
          model = TRUE, lambda = 1e-6, x = FALSE, y = TRUE, singular.ok = TRUE,
          contrasts = NULL, ...)
```

In particular, it allows the formula specification of the response and predictor variables in the same form as many R base function such as `lm()` and `glm()`, which is handy when categorical predictors are present in the data. Also, similar to `glm()` and `glmnet()`, it accepts any family function defined in the **stats** package as well as any user-defined family function. A few method functions are associated with the `glm.ridge()` function. They are `extractAIC.glm.ridge()`, `predict.glm.ridge()`, and `AICtweedie.glm.ridge()`.

4.5 The Tweedie family function

The Tweedie family function `tweedie()` is an improvement over its counterpart from the **statmod** package (Giner and Smyth, 2016) by defining a better power link function and semantic checks. As required by the `glm()` function, it defines the link function, inverse of the link function, the variance function, the function for calculating the deviance residuals, and the derivative of the mean function with respect to the linear predictor. The family is broad enough to also include those special cases such as the normal, Poisson, gamma, and inverse Gaussian family, and provides ways to handle more type of link functions beyond the commonly used logarithmic link.

4.6 Examples

We start with a simple simulation study to demonstrate the usefulness of our function in estimating the variable selection performance of a method. We consider an eight-dimensional covariate vector $\mathbf{x} = (X_1, \ldots, X_8)^\intercal$ from the multivariate normal distribution $N(\mathbf{0}, \Sigma)$ with an autoregressive covariance structure $\Sigma = (0.5^{|i-j|})$. The response Y is

simulated from the linear model

$$Y = \mathbf{x}^\intercal \boldsymbol{\beta}^* + \epsilon,$$

where $\epsilon \sim N(0, 1)$ and is independent of $\mathbf{x}$, and $\boldsymbol{\beta}^* = (3, 1.5, 0, 0, 2, 0, 0, 0)^\intercal$.

We generate a random sample of $n = 50$ observations from the above model.

```
## Generate data
set.seed(2022)
n <- 50
p <- 8
beta <- c(3, 1.5, 0, 0, 2, 0, 0, 0)
sigma <- (0.5) ^ abs(outer(seq(p), seq(p), "-"))
x <- matrix(rnorm(n * p), n, p) %*% chol(sigma)
e <- rnorm(n)
y <- as.vector(x %*% beta) + e
```

Given a method that results in the selection of variables X_2, X_3, X_4 and X_8, suppose we want to evaluate its variable selection performance using the ARM weighting with prior adjustment $\psi = 1$. To make things simple, we let the function itself pick the set of candidate models using the union of solution paths returned from the lasso, MCP and SCAD penalized least squares.

```
library(glmvsd)
## Model to be checked
model_check <- c(0, 1, 1, 1, 0, 0, 0, 1)
## Compute VSD using ARM with prior
vsd_arm <- glmvsd(x, y, model_check = model_check, psi = 1, family = "gaussian",
                  method = "union", weight_type = "ARM", prior = TRUE)
vsd_arm

## $VSD
```

```
## [1] 5.002283
##
## $VSD_minus
## [1] 2.792144
##
## $VSD_plus
## [1] 2.210139
##
## $precision
## [1] 0.3019641
##
## $recall
## [1] 0.343114
##
## $F_measure
## [1] 0.2227004
##
## $G_measure
## [1] 0.3199104
##
## $sd.F
## [1] 0.1401151
##
## $sd.G
## [1] 0.1061944
##
## $weight
##  [1] 2.793789e-10 7.387803e-06 9.604812e-05 2.633081e-02 6.507289e-01
##  [6] 1.335805e-01 1.175402e-01 3.493754e-02 1.349825e-02 1.188448e-02
```

```
## [11] 5.778429e-03 3.907721e-03 1.709814e-03
##
## $model_check
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]    0    1    1    1    0    0    0    1
##
## $candidates_models
##       V1 V2 V3 V4 V5 V6 V7 V8
##  [1,]  0  0  0  0  0  0  0  0
##  [2,]  1  0  0  0  0  0  0  0
##  [3,]  1  1  0  0  0  0  0  0
##  [4,]  1  0  0  0  1  0  0  0
##  [5,]  1  1  0  0  1  0  0  0
##  [6,]  1  1  1  0  1  0  0  0
##  [7,]  1  1  0  0  1  0  1  0
##  [8,]  1  1  1  0  1  0  1  0
##  [9,]  1  1  1  0  1  1  1  0
## [10,]  1  1  1  0  1  0  1  1
## [11,]  1  1  1  0  1  1  1  1
## [12,]  1  1  1  1  1  0  1  1
## [13,]  1  1  1  1  1  1  1  1
##
## attr(,"class")
## [1] "glmvsd"
```

Note that the true model lies on the solution paths of the regularized regression models and corresponds to the fifth candidate model. Our weighting method assigns the majority of the weight to it ($w_5 = 0.651$). Since the checked model (arbitrarily picked) is quite far away from the true model, our calculated VSD measures have large values and F- and

G-measures have small values which clearly reflect the fact that the checked model may not be properly supported.

We also consider a simple binary classification example, where $(Y \mid \mathbf{x}) \sim \text{Bernoulli}(p(\mathbf{x}))$ with $p(\mathbf{x}) = [1 + \exp(-\mathbf{x}^\intercal \boldsymbol{\beta}^*)]^{-1}$. We assume $\mathbf{x} \sim N(\mathbf{0}, \Sigma)$ with $\Sigma = (0.5 + 0.5I(j = j'))_{p\times p}$ and $\boldsymbol{\beta}^* = (1, 0.5, 1.5, -\frac{3}{2}\sqrt{2}, \mathbf{0}_{p-4})^\intercal$. Now take $n = 100$ and $p = 200$ and simulate the data.

```
## Generate data
set.seed(2002)
n <- 100
p <- 200
b <- c(1,  1,  1,  -3 * sqrt(2) / 2,  rep(0,  p - 4))
sigma <- matrix(0.5,  p,  p) + diag(0.5,  p)
x <- matrix(rnorm(n * p),  n,  p) %*% chol(sigma)
eta.x <- x %*% b
p.x <- 1 / (1 + exp(-eta.x))
y <- rbinom(n,  1,  p.x)
```

Suppose we want to evaluate the model that selects X_1, X_3 and X_4 using the BIC weighting with prior adjustment ($\psi = 1$).

```
model_check <- c(1, 0, 1, 1, rep(0, p - 4))
vsd_bic <- glmvsd(x, y, family = "binomial", model_check = model_check, psi = 1,
    method = "union", weight_type = "BIC", prior = TRUE, reduce_bias = TRUE)
str(vsd_bic)

## List of 13
##  $ VSD              : num 0.557
##  $ VSD_minus        : num 0.312
##  $ VSD_plus         : num 0.245
##  $ precision        : num 0.896
##  $ recall           : num 0.942
```

```
##  $ F_measure        : num 0.889
##  $ G_measure        : num 0.903
##  $ sd.F             : num 0.198
##  $ sd.G             : num 0.175
##  $ weight           : num [1:66] 8.19e-03 1.43e-01 6.08e-04 1.12e-06 7.12e-01 ...
##  $ model_check      : num [1, 1:200] 1 0 1 1 0 0 0 0 0 0 ...
##  $ candidates_models: int [1:66, 1:200] 0 0 1 1 1 1 1 1 1 1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:200] "V1" "V2" "V3" "V4" ...
##  $ ylevels          : chr [1:2] "0" "1"
##  - attr(*, "class")= chr "glmvsd"
```

We can see that this checked model is reasonably good to have low VSD, VSD^+ and VSD^- values and high F- and G-measures since it is close to the true model. The true model receives the highest weight $w_5 = 0.712$. Note that we have used the `reduce_bias` option to avoid convergence issues with the logistic regressions due to nearly-perfect separation in many submodels.

4.7 Summary

In this chapter, we present the R package **PAVI** for computing various variable selection deviation measures, including VSD, VSD^+, VSD^-, F- and G-measures, to quantify the selection performance of a variable selection method, as proposed in Nan and Yang (2014), Yu et al. (2022) and Chapters 2 and 3 of this dissertation. The package provides a framework for calculating the deviation measures for all members of the generalized linear model family. It goes beyond the available family functions in the **stats** R package to also include

the highly useful Tweedie family. Tremendous efforts have been spent stabilizing the weight calculations which give rise to bias-reduced and ridge procedures for fitting some of the pathological submodels. A parallel mechanism is implemented in the package to accelerate the ARM weight computation. We hope that this package will be a valuable addition to the existing family of high-dimensional variable selection methods and be especially useful for measuring variable selection performance of various methods on real problems. Future work will aim at further improving the speed of computing the weights for large sets of candidate models by fitting the candidate models in a parallel fashion. We also plan to include functions for more specialized use case, such as separation checking and collinearity checking of the design matrix in candidate models after data splitting.

Chapter 5

Conclusion

5.1 Discussion

In this dissertation, we considered approaches to measuring variable selection uncertainties of high-dimensional regularized regression models, including proposing effective ways of estimating the VSD, F-measure and G-measure in many members of the generalized linear model family. We also provided an R package which implements the proposed methods for calculating the deviation measures for various high-dimensional generalized linear models.

In Chapter 2, we studied the F- and G-measures based on a combination of multiple candidate models under proper weighting schemes to better access the variable identification in high-dimensional models. We demonstrated its application to both regression and classification, and to both synthetic and real data. Theoretically, we proved that our estimates are uniformly consistent in estimating the true F- and G-measures for any set of models to be checked. Two weighting schemes were proposed to supplement the theories, including the ARM-based and information-criterion-based weighting. Extensive simulations were conducted which showed the very good finite-sample performance of our approach. We also applied our methods to several microarray gene expression data

sets and made intriguing findings.

In Chapter 3, we extended the work of Nan and Yang (2014) and Yu et al. (2022) to more members of the generalized linear model family, with an emphasis on the Poisson and Tweedie models for their wide spectrum of applications in many applied fields. Our numerical studies demonstrated that the proposed weighting schemes work well for the high-dimensional Poisson and Tweedie models.

In Chapter 4, we gave a detailed account of the usage of the main function and various helper functions in our R package **PAVI**, which supplements the previous chapters by providing the numerical software for computing the variable selection diagnostic measures such as VSD, VSD^+, VSD^-, precision, recall, F- and G-measures. The various helper functions were shown to help fulfill tasks such as candidate model fitting and weight assignment for the candidate models using parallel scheme that greatly improves the computational efficiency. The simulation examples provided a good demonstration of the functional usage.

5.2 Future Work

This dissertation provides an aspiring start for the methodological and numerical research on performance assessment of high-dimensional variable selection methods. On the one hand, the methods can be extended to other high-dimensional (quasi) likelihood-based models such as the Cox model (Cox, 1972) and generalized estimating equations (GEEs, Liang and Zeger, 1986), and to loss-driven models such as the quantile regression (Koenker and Bassett, 1978). For example, censored data are commonly encountered in observational studies and clinical trials in a variety of fields such as medicine, biology, epidemiology, sociology and economics. The Cox model is the most widely used survival model for

censored data. When such data include many features, variable selection is often needed. Our methodologies have the potential to help measure the variable selection uncertainty from a high-dimensional Cox model. GEEs are often used to model correlated responses in longitudinal data. Pan (2001) provides an AIC-like information criterion for GEEs using the quasi-likelihood, which can be potentially employed to develop variable selection deviation measures for regularized GEEs (Wang et al., 2012).

On the other hand, it would be interesting to have proper variable selection deviation measures when categorical features are present. Ideally, the multiple dummy variables corresponding to the same categorical feature should be counted as one group. The current framework treats these dummy variables as separate. It would be useful and of practical interest to extend our current work to incorporate the deviation measures from a group perspective.

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory (Tsahkadsor, 1971)*. Budapest: Akadémiai Kiadó, pp. 267–281.

Albert, A. and Anderson, J. A. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika* **71**, 1–10.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D. and Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences* **96**, 6745–6750.

Baggerly, K. A., Morris, J. S. and Coombes, K. R. (2004). Reproducibility of SELDI-TOF protein patterns in serum: comparing datasets from different experiments. *Bioinformatics* **20**, 777–785.

Bartlett, M. (1953). Approximate confidence intervals. *Biometrika* **40**, 12–19.

Billsus, D. and Pazzani, M. J. (1998). Learning collaborative information filters. In *International Conference on Machine Learning*, vol. 98. Morgan Kaufmann Publishers.

Breheny, P. (2011). Regularization paths for SCAD and MCP penalized regression models.

Breheny, P. and Huang, J. (2011). Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics* **5**, 232–253.

Breiman, L. (1996a). Bagging predictors. *Machine Learning* **24**, 123–140.

Breiman, L. (1996b). Heuristics of instability and stabilization in model selection. *The Annals of Statistics* **24**, 2350–2383.

Buckland, S., Burnham, K. and Augustin, N. (1997). Model selection: an integral part of inference. *Biometrics* **53**, 603–618.

Chandra, B. and Gupta, M. (2011). An efficient statistical feature selection approach for classification of gene expression data. *Journal of Biomedical Informatics* **44**, 529–535.

Chatfield, C. (1995). Model uncertainty, data mining and statistical inference. *Journal of the Royal Statistical Society. Series A (Statistics in Society)* **158**, 419–466.

Chen, L., Giannakouros, P. and Yang, Y. (2007). Model combining in factorial data analysis. *Journal of Statistical Planning and Inference* **137**, 2920–2934.

Coates, J. M., Gullo, M. J., Feeney, G. F. X., Young, R. M. and Connor, J. P. (2018). A randomized trial of personalized cognitive-behavior therapy for alcohol use disorder in a public health clinic. *Frontiers in Psychiatry* **9**, 297.

Cordeiro, G. M., Cribari-Neto, F. et al. (2014). *An introduction to Bartlett correction and bias reduction.* Springer.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)* **34**, 187–202.

Ding, J., Tarokh, V. and Yang, Y. (2018). Model selection techniques: An overview. *IEEE Signal Processing Magazine* **35**, 16–34.

Draper, D. (1995). Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **57**, 45–97.

Dunn, P. K. (2004). Occurrence and quantity of precipitation can be modelled simultaneously. *International Journal of Climatology* **24**, 1231–1239.

Dunn, P. K. and Smyth, G. K. (2005). Series evaluation of tweedie exponential dispersion model densities. *Statistics and Computing* **15**, 267–280.

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

Fan, J., Li, R., Zhang, C.-H. and Zou, H. (2020). *Statistical foundations of data science*. Chapman and Hall/CRC.

Fan, J. and Lv, J. (2010). A selective overview of variable selection in high dimensional feature space. *Statistica Sinica* **20**, 101–148.

Fan, J., Xue, L. and Zou, H. (2014). Strong oracle optimality of folded concave penalized estimation. *The Annals of Statistics* **42**, 819–849.

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika* **80**, 27–38.

Friedman, J., Hastie, T. and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33**, 1.

Giner, G. and Smyth, G. K. (2016). statmod: probability calculations for the inverse gaussian distribution. *arXiv preprint arXiv:1603.06687* .

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R. and Caligiuri, M. A. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286**, 531–537.

Henry, N. L. and Hayes, D. F. (2012). Cancer biomarkers. *Molecular Oncology* **6**, 140–146.

Hoeting, J. A., Madigan, D., Raftery, A. E. and Volinsky, C. T. (1999). Bayesian model averaging: A tutorial. *Statistical Science* **14**, 382–401.

Ji, G., Yang, Z. and You, W. (2011). PLS-based gene selection and identification of tumor-specific genes. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **41**, 830–841.

Jørgensen, B. (1987). Exponential dispersion models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* , 127–162.

Jørgensen, B. (1997). *The theory of dispersion models*, vol. 76. CRC Press.

Karatzoglou, A., Smola, A., Hornik, K. and Zeileis, A. (2004). kernlab - An S4 package for kernel methods in R. *Journal of Statistical Software* **11**, 1–20.

Koenker, R. and Bassett, G. (1978). Regression quantiles. *Econometrica: Journal of the Econometric Society* , 33–50.

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

Lauderdale, B. E. (2012). Compound poisson-gamma regression models for dollar outcomes that are sometimes zero. *Political Analysis* **20**, 387–399.

Lee, C.-P. and Leu, Y. (2011). A novel hybrid feature selection method for microarray data analysis. *Applied Soft Computing* **11**, 208–213.

Leung, G. and Barron, A. R. (2006). Information theory and mixing least-squares regressions. *IEEE Transactions on Information Theory* **52**, 3396–3410.

Leung, Y. and Hung, Y. (2010). A multiple-filter-multiple-wrapper approach to gene selection and microarray data classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **7**, 108–117.

Liang, K.-Y. and Zeger, S. L. (1986). Longitudinal data analysis using generalized linear models. *Biometrika* **73**, 13–22.

Lim, C. (2011). *Modeling High Dimensional Data: Prediction, Sparsity, and Robustness*. Ph.D. thesis, University of California, Berkeley.

Lim, C. and Yu, B. (2016). Estimation stability with cross-validation (ESCV). *Journal of Computational and Graphical Statistics* **25**, 464–492.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*, vol. 37. CRC Press.

McNutt, M. (2014). Raising the bar. *Science* **345**, 9.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics* **34**, 1436–1462.

Nan, Y. (2015). Variable selection deviation measures. *Retrieved from the University of Minnesota Digital Conservancy* .

Nan, Y. and Yang, Y. (2014). Variable selection diagnostics measures for high-dimensional regression. *Journal of Computational and Graphical Statistics* **23**, 636–656.

Nolan, J. (2020). *Univariate Stable Distributions: Models for Heavy Tailed Data*. Springer Series in Operations Research and Financial Engineering. Springer International Publishing.

Pan, W. (2001). Akaike's information criterion in generalized estimating equations. *Biometrics* **57**, 120–125.

Qian, W., Yang, Y. and Zou, H. (2016). Tweedie's compound poisson model with grouped elastic net. *Journal of Computational and Graphical Statistics* **25**, 606–625.

Santner, T. J. and Duffy, D. E. (1986). A note on A. Albert and JA Anderson's conditions for the existence of maximum likelihood estimates in logistic regression models. *Biometrika* **73**, 755–758.

Schwarz, G. (1978a). Estimating the dimension of a model. *The Annals of Statistics* **6**, 461–464.

Schwarz, G. (1978b). Estimating the dimension of a model. *The Annals of Statistics* , 461–464.

Sharma, A., Imoto, S. and Miyano, S. (2012). A top-r feature selection algorithm for microarray gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **9**, 754–764.

Silva, J. S. and Tenreyro, S. (2010). On the existence of the maximum likelihood estimates in poisson regression. *Economics Letters* **107**, 310–312.

Silva, J. S. and Tenreyro, S. (2011). Poisson: Some convergence issues. *The Stata Journal* **11**, 207–212.

Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A. A., D'Amico, A. V. and Richie, J. P. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell* **1**, 203–209.

Smyth, G. K. (1996). Regression analysis of quantity data with exact zeros. In *Proceedings of the Second Australia–Japan Workshop on Stochastic Models in Engineering, Technology and Management*. Citeseer.

Smyth, G. K. and Jørgensen, B. (2002). Fitting Tweedie's compound poisson model to insurance claims data: dispersion modelling. *ASTIN Bulletin: The Journal of the IAA* **32**, 143–157.

Stodden, V. (2015). Reproducing statistical results. *Annual Review of Statistics and Its Application* **2**, 1–19.

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

Tweedie, M. (1984). An index which distinguishes between some important exponential families. In *Statistics: Applications and New Directions: Proc. Indian Statistical Institute Golden Jubilee International Conference*.

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, vol. 48. Cambridge University Press.

Wang, L., Zhou, J. and Qu, A. (2012). Penalized generalized estimating equations for high-dimensional longitudinal data analysis. *Biometrics* **68**, 353–360.

Wang, Z., Paterlini, S., Gao, F. and Yang, Y. (2014). Adaptive minimax regression estimation over sparse lq-hulls. *The Journal of Machine Learning Research* **15**, 1675–1711.

West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson, J. A., Marks, J. R. and Nevins, J. R. (2001). Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of the National Academy of Sciences* **98**, 11462–11467.

Yang, A.-J. and Song, X.-Y. (2010). Bayesian variable selection for disease classification using gene expression data. *Bioinformatics* **26**, 215–222.

Yang, Y. (1999). Model selection for nonparametric regression. *Statistica Sinica* **9**, 475–499.

Yang, Y. (2000). Adaptive estimation in pattern recognition by combining different procedures. *Statistica Sinica* **10**, 1069–1090.

Yang, Y. (2001). Adaptive regression by mixing. *Journal of the American Statistical Association* **96**, 574–588.

Yang, Y. (2007). Consistency of cross validation for comparing regression procedures. *The Annals of Statistics* **35**, 2450–2473.

Yang, Y. and Barron, A. R. (1998). An asymptotic property of model selection criteria. *IEEE Transactions on Information Theory* **44**, 95–116.

Ye, C. and Yang, Y. (2019). High-dimensional adaptive minimax sparse estimation with interactions. *IEEE Transactions on Information Theory* **65**, 5367–5379.

Ye, C., Yang, Y. and Yang, Y. (2018). Sparsity oriented importance learning for high-

dimensional linear regression. *Journal of the American Statistical Association* **113**, 1797–1812.

Yu, Y., Yang, Y. and Yang, Y. (2022). Performance assessment of high-dimensional variable identification. *Statistica Sinica* **32**, 695–718.

Yuan, Z. and Yang, Y. (2005). Combining linear regression models: when and how? *Journal of the American Statistical Association* **100**, 1202–1214.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics* **38**, 894–942.

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

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

Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *The Annals of Statistics* **36**, 1509–1533.